Application of Fuzzy-Rough Sets in Modular Neural Networks

Manish Sarkar and B. Yegnanarayana
Department of Computer Science & Engineering
Indian Institute of Technology, Madras - 600 036, INDIA
{manish@bronto, yegna}.iitm.ernet.in

Abstract

In a modular neural network, the conflicting information supplied by the information sources, i.e., the outputs of the subnetworks, can be combined by applying the concept of fuzzy integral. To compute the fuzzy integral, it is essential to know the importance of each subset of the information sources in a quantified form. In practice, it is very difficult to determine the worth of the information sources. However, in the fuzzy integral approach the importance of a particular information source is considered to be independent of the other information sources. Therefore, determination of the importance of each information source should be based on the incomplete knowledge supplied by the source itself. This paper proposes a fuzzy-rough set theoretic approach to find the importance of each subset of the information sources from this incomplete knowledge.

I. Introduction

In many complex pattern classification tasks, where the number of classes is large and the similarity amongst the classes is high, it is difficult to train a monolithic feedforward neural network for the whole classification task. In these cases one viable approach is ***divide and conquer***, which permits one to solve a complex classification task by dividing it into simpler subtasks, and then by combining the solutions of the subtasks. The philosophy of modular neural network is based on this principle. In the modular approach to classification, the classes are grouped into smaller subgroups, and a separate neural network is trained for each subgroup [1]. The outputs of the modules are mediated by an integrating unit, which is not permitted to feed the information back to the modules. One way to decompose a network is to create modules that serve very different functions, not different versions of the same function. The top-down structure of large software projects is an example, where each procedure has its own function. This is called ***functional*** modularization [2]. Another way is to decompose the networks such that the modules perform different versions of the same job. It is called *categorical* modularization. This can be thought of as a set of experts giving their individual opinions on the same subject.

This paper proposes a technique to fuse the information supplied by the subnetworks of a modular network with functional modularization. The proposed method interprets each subnetwork as a nonlinear filter tailored to the subgroup. The set of outputs of all the filters is viewed as a feature vector representing the input. Each module classifies the input pattern from different angles. In other words, each feature, i.e., the set of outputs of each module, can be considered as an evidence in classifying the input. Each of these evidences may support or contradict one another. Hence, each of these evidences would have a different degree of importance in classifying the input. The classification capability of an evidence for a particular class is known as ***partial evaluation***. The fuzzy integral combines the partial evaluations of all the evidences with the importance of the subsets of the evidences to yield the final classification result.

The behavior of fuzzy integral in an application depends critically on the importance of the subsets of the evidences, which further depends on the importance of the individual evidences. Therefore, determination of the worth of each evidence is very important. In some applications of the fuzzy integral, these importances are supplied subjectively by an expert or they are estimated directly from the data [3] [4]. These methods require some kinds of prior knowledge about the information sources. In many applications, it may be very difficult to obtain this type of prior knowledge. However, it is interesting to note that in the fuzzy integral approach while considering a particular evidence, influence of the other evidences is not considered. Hence, determination of the importance of a particular evidence is based on the partial information supplied by the evidence itself. The notion of rough sets [5] can be effectively exploited to determine the importance of each evidence from this incomplete knowledge. Moreover, the information supplied by each evidence in terms of the outputs of the subnetworks is inherently fuzzy. Therefore, in this paper, an attempt is made to determine the importance of each evidence from this incomplete knowledge by using a fuzzy-rough set [6] theoretic technique. The performance of the proposed scheme is studied for a Contract Bridge Opening Bid problem.

II. Background

A. Fuzzy Measure

Let Ξ be a finite set of elements. A set function $g : 2^{\Xi} \rightarrow [0, 1]$ with the following properties is called a fuzzy measure [7]:

P1: $g(\phi) = 0$

P2: $g(\Xi) = 1$

P3: If $\mathbf{U} \subseteq \mathbf{V}$, then $g(\mathbf{U}) \subseteq g(\mathbf{V})$, where $\mathbf{U}, \mathbf{V} \subseteq \Xi$

The fuzzy measure generalizes the classical measure which plays a crucial role in the probability and integration theory. A probability measure P is characterized by the property of additivity: For all sets $\mathbf{U}$ and $\mathbf{V}$, if

0-7803-4859-1/98 $10.00©1998 IEEE

$\mathbf{U} \cap \mathbf{V} = \phi$, then $P(\mathbf{U} \cup \mathbf{V}) = P(\mathbf{U}) + P(\mathbf{V})$. In the fuzzy measure this property of additivity is weakened by the more general property of monotonicity (property *P3*). Sugeno's g_λ measure is a special type of fuzzy measure [7] which satisfies all the properties of the fuzzy measure, in addition to the following:

$$g(\mathbf{U} \cup \mathbf{V}) = g(\mathbf{U}) + g(\mathbf{V}) + \lambda g(\mathbf{U})g(\mathbf{V}) \qquad (1)$$

where $\lambda > -1$, $\mathbf{U}, \mathbf{V} \subseteq \Xi$ and $\mathbf{U} \cap \mathbf{V} = \phi$. By varying the values of λ, one can obtain different types of fuzzy measure. For example, $\lambda = 0$ gives the probability measure.

B. Fuzzy Integral

Let $\Xi = \{\xi_1, \xi_2, \ldots, \xi_S\}$ be a finite set of elements, $h : \Xi \rightarrow [0, 1]$ be a mapping and g be a fuzzy measure on Ξ. Then the fuzzy integral (over Ξ) of the function h with respect onto the fuzzy measure g is defined as

$$\begin{aligned} e &= h(\Xi) \circ g() & (2) \\ &= \max_{\Omega \subseteq \Xi} \left[\min \left(\min_{\xi_s \in \Omega} (h(\xi_s)),\ g(\Omega) \right) \right] & (3) \end{aligned}$$

where $1 \leq s \leq S$. Since both h and g map onto [0, 1], e also lies in [0, 1]. Intuitively the interpretation of the above relation is as follows: Let us suppose, an object is evaluated from the point of view of a set of information sources Ξ. Let $h(\xi_s) \in [0,\ 1]$ denote the decision for the object when a single information source $\xi_s \in \Xi$ is considered. Moreover, suppose $g(\{\xi_s\})$, known as ***fuzzy density***, denotes the importance of the source ξ_s. Instead of a single information source, if a set of sources, namely $\Omega \subseteq \Xi$, is taken to evaluate the object, then it is reasonable to consider $\min_{\xi_s \in \Omega} h(\xi_s)$ as the largest security decision. Evidently, $g(\Omega)$ expresses the degree of importance or the expected worth of the set Ω. Therefore, $\min \left(\min_{\xi_s \in \Omega} (h(\xi_s)), g(\Omega) \right)$ denotes the grade of agreement between the real possibility h and the expectation g. Thus, the fuzzy integral can be interpreted as a search for the maximal grade of agreement between the objective evidence and the expectation. However, the definition can further be simplified if $h(\xi_s)$, $s = 1, 2, \ldots, S$ are ordered in a decreasing manner. Let $h(\xi_1) \geq h(\xi_2) \geq \ldots \geq h(\xi_S)$ (if not, Ξ is rearranged so that this relation holds). Then the relation (3) is simplified to

$$e = h(\Xi) \circ g() = \max_s \left[\min (h(\xi_s),\ g(\Omega_s)) \right] \qquad (4)$$

where $\Omega_s = \{\xi_1, \xi_2, \cdots, \xi_s\}$.

In order to evaluate the fuzzy integral, i.e., e, we should have some way to determine $g(\Omega_s)$ from $g(\{\xi_s\})$. For that, we need to use the concept of the fuzzy measure. In the next section we will show how to determine the individual fuzzy densities $g(\{\xi_s\})$, $s = 1, 2, \ldots, S$ for each information source from the given data. For the time being, let us suppose that we know the fuzzy densities of the individual sources. But, $g(\Omega_s)$ is not necessarily equal to $g(\{\xi_1\}) + g(\{\xi_2\}) + \ldots + g(\{\xi_s\})$. The simple additive property may not hold because there may be some interaction among ξ_s. If the interactions are cooperative then $g(\Omega_s) \geq g(\{\xi_1\}) + g(\{\xi_2\}) + \ldots + g(\{\xi_s\})$. On the contrary, if the interactions are noncooperative, then $g(\Omega_s) \leq g(\{\xi_1\}) + g(\{\xi_2\}) + \ldots + g(\{\xi_s\})$ [8]. From this discussion, note that the probability theory cannot be used to determine the value of $g(\Omega_s)$. However, the concept of Sugeno's g_λ fuzzy measure can be exploited here to find the value of $g(\Omega_s)$. The procedure is as follows:

$$\begin{aligned} g(\Omega_1) &= g(\{\xi_1\}) \\ g(\Omega_2) &= g(\{\xi_2\}) + g(\{\xi_1\}) + \lambda g(\{\xi_2\})g(\{\xi_1\}) \\ &= g(\{\xi_2\}) + g(\Omega_1) + \lambda g(\{\xi_2\})g(\Omega_1) \\ \cdots & \quad \cdots \quad \cdots \quad \cdots \\ g(\Omega_s) &= g(\{\xi_s\}) + g(\Omega_{s-1}) + \lambda g(\{\xi_s\})g(\Omega_{s-1}) \\ & \qquad \text{for } 1 < s \leq S \qquad (5) \end{aligned}$$

One problem remains still unresolved; that is, how to determine λ, which is the key term to decide the amount of interaction among the information sources. In order to find λ, we use the equation (5), and we express $g(\Xi\ )$ in terms of the individual fuzzy densities as follows:

$$\begin{aligned} g(\Xi) &= g(\{\xi_S\}) + g(\{\xi_1, \xi_2, \ldots, \xi_{S-1}\}) \\ & \quad + g(\{\xi_S\})\lambda g(\{\xi_1, \xi_2, \ldots, \xi_{S-1}\}) & (6) \\ &= \sum_{s=1}^{S} g(\{\xi_s\}) \\ & \quad + \lambda \sum_{s=1}^{S-1} \sum_{k=s+1}^{S} g(\{\xi_s\})g(\{\xi_k\}) + \cdots \\ & \quad + \lambda^{S-1} g(\{\xi_1\})g(\{\xi_2\}) \cdots g(\{\xi_S\}) & (7) \\ &= \left[\prod_{s=1}^{S} (1 + \lambda g(\{\xi_s\}) - 1 \right] \Big/ \lambda \\ & \qquad \text{where } \lambda \neq 0 & (8) \end{aligned}$$

From (*P2*), we know that the value of g over the whole set Ξ must be one as no uncertainty is involved. Hence, using $g(\Xi) = 1$ and the equation (8), we get

$$\prod_{s=1}^{S} (1 + \lambda g(\{\xi_s\}) = \lambda + 1 \qquad (9)$$

It is possible to find the value of λ after solving the above $(S - 1\ )$th degree equation. In [3], it has been shown that λ has a unique value in $(-1, 0) \cup (0, +\infty)$ when $0 < g(\{\xi_s\}) < 1, \quad \forall s = 1, 2, \ldots, S$.

C. Rough Sets

In any classification task the aim is to form various classes where each class contains objects that are not significantly different. These *indiscernible* or non-distinguishable objects can be viewed as basic building blocks (concepts) used to build up a knowledge base about the real world. For example, if the objects are classified according to color (red, black) and shape (triangle, square and circle), then the classes are: red triangles, black squares, red circles, etc. Thus, these two attributes make a *partition* in the set of objects and the universe becomes coarse. Now, if two red triangles with different areas belong to different classes, it is impossible for anyone to correctly classify these two red triangles based on the given two attributes. This kind of uncertainty is referred to as ***rough uncertainty*** [5]. The rough uncertainty is formulated in terms of ***rough sets*** [9]. Obviously, the rough uncertainty can be completely avoided if we can successfully extract the essential features so that distinct feature vectors are used to represent different objects. But, it may not be possible to guarantee as our knowledge about the system generating the data is limited . Therefore, rough sets are essential to deal with a classification system, where we do not have complete knowledge about the system.

Next, we briefly describe the formulation of rough sets. Let $\mathbf{R}$ be an equivalence relation on a universal set $\mathbf{\Psi}$. Moreover, let $\mathbf{\Psi}/\mathbf{R}$ denote the family of all equivalence classes induced on $\mathbf{\Psi}$ by $\mathbf{R}$. One such equivalence class in $\mathbf{\Psi}/\mathbf{R}$, that contains $\boldsymbol{\psi} \in \mathbf{\Psi}$, is designated by $[\boldsymbol{\psi}]_{\mathbf{R}}$. For any crisp subset $\mathbf{A} \subseteq \mathbf{\Psi}$, we can define the lower $\overline{\mathbf{R}}(\mathbf{A})$ and upper $\underline{\mathbf{R}}(\mathbf{A})$ approximations, which approach $\mathbf{A}$ as closely as possibly from inside and outside, respectively [10]. Here,

$$\underline{\mathbf{R}}(\mathbf{A}) = \cup\{[\boldsymbol{\psi}]_{\mathbf{R}} \mid [\boldsymbol{\psi}]_{\mathbf{R}} \subseteq \mathbf{A},\ \boldsymbol{\psi} \in \mathbf{\Psi}\} \tag{10}$$

is the union of all the equivalence classes in $\mathbf{\Psi}/\mathbf{R}$ that are contained in $\mathbf{A}$, and

$$\overline{\mathbf{R}}(\mathbf{A}) = \cup\{[\boldsymbol{\psi}]_{\mathbf{R}} \mid [\boldsymbol{\psi}]_{\mathbf{R}} \cap \mathbf{A} \neq \phi,\ \boldsymbol{\psi} \in \mathbf{\Psi}\} \tag{11}$$

is the union of all the equivalence classes in $\mathbf{\Psi}/\mathbf{R}$ that overlap with $\mathbf{A}$. A rough set $\mathbf{R}(\mathbf{A}) = \langle\overline{\mathbf{R}}(\mathbf{A}), \underline{\mathbf{R}}(\mathbf{A})\rangle$ is a representation of the given set $\mathbf{A}$ by $\underline{\mathbf{R}}(\mathbf{A})$ and $\overline{\mathbf{R}}(\mathbf{A})$. The set difference $\overline{\mathbf{R}}(\mathbf{A}) - \underline{\mathbf{R}}(\mathbf{A})$ is a rough description of the boundary of $\mathbf{A}$ by the equivalence classes of $\mathbf{\Psi}/\mathbf{R}$. The approximation is rough uncertainty free if $\overline{\mathbf{R}}(\mathbf{A}) = \underline{\mathbf{R}}(\mathbf{A})$. Thus, when all the patterns from an equivalence class do not carry the same output class labels, rough ambiguity is generated as a manifestation of the one-to-many relationship between that equivalence class and the output class labels.

III. Modular Neural Networks

A. Architecture

The given pattern classification task is initially subdivided into several subtasks, and one subnetwork is assigned for each subtask. Let, the original problem has M output classes $\{\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_M\}$, and these classes are divided into S subnetworks (Fig. 1). The sth subnetwork is assigned to classify a group of classes, represented by $\{\mathbf{C}_{c_{s-1}+1}, \cdots, \mathbf{C}_{c_s}\}$ with $\mathrm{c}_0 = 0$ and $\mathrm{c}_S = M$. The outputs of the sth subnetwork are $\{y_{c_{s-1}+1}, \cdots, y_{c_s}\}$, which are expressed in a vector notation as $\boldsymbol{\xi}_s = [y_{c_{s-1}+1}, \cdots, y_{c_s}]^T$. The proposed method interprets each subnetwork as a nonlinear filter tailored to the subgroup. Thus, the outputs of all the filters corresponding to an input $\mathbf{x}$ is viewed as an S dimensional feature vector $\boldsymbol{\psi} = [\boldsymbol{\xi}_1^T, \boldsymbol{\xi}_2^T, \ldots, \boldsymbol{\xi}_S^T]^T$. This feature vector is presented as an input to a fuzzy integrator, which computes the fuzzy integral value with the help of fuzzy densities. The class label of the input $\mathbf{x}$ is the class index that yields the maximum fuzzy integral value corresponding to $\boldsymbol{\psi}$.

B. Training

When a modular network is used for classification, a given training pattern is input to all the subnetworks and the outputs of the subnetworks are processed to determine the class. We can decide the class label of the input based on the winner-take-all policy. It means that the class label of the input pattern is assigned as j, $1 \leq j \leq M$, if $y_j = \max\limits_{k=1,2,\ldots,M}\{y_k\}$. However, this kind of assignment is not proper as all the subnetworks are independently trained on different sets of data. Hence, a better approach is to declare the jth class winner, if the jth class correspondences to $\max\limits_{k=1,2,\ldots,M}\{g_k y_k\}$, where g_k is the importance associated with the class $\mathbf{C}_k$. One possible choice for g_k is the ***a posteriori*** probability of the class $\mathbf{C}_k$. However, the constraint $\sum\limits_{k=1,2,\ldots,M} g_k = 1$ used in the probability theory cannot distinguish between lack of evidence and ignorance. Therefore, the concept of the fuzzy integral is appealing here. In the fuzzy integral approach, the outputs of the modules are processed further so that the interactions among the outputs are also exploited for the final classification result. Hence, the term g_k is replaced by a more specific term $g_k(\{\boldsymbol{\xi}_s\})$, where $g_k(\{\boldsymbol{\xi}_s\})$ denotes the importance of $\boldsymbol{\xi}_s$ in characterizing the class $\mathbf{C}_k$. With the help of $g_k(\{\boldsymbol{\xi}_s\})$, $s = 1, 2, \ldots, S$, the fuzzy integral e_k for the class $\mathbf{C}_k$ combines the outputs of all the modules, i.e., $\boldsymbol{\xi}_s$, $s = 1, 2, \ldots, S$, in a nonlinear fashion. The final class label corresponding to the input is j, if $e_j = \max\limits_{k=1,2,\ldots,M}\{e_k\}$. Specifically, the training of the whole modular network is comprised of the following two stages:

Training of each subnetwork: For this stage, separate data sets are prepared to train the subnetworks independently. The training data set for a subnetwork generally consists of patterns belonging to the classes in its subgroup only. Then each subnetwork is trained by the conventional backpropagation algorithm to form the decision surfaces for the classes in its subgroup. It may be necessary to train each subnetwork with a few patterns belonging to the output classes of the other subnetworks. These patterns may be considered as negative examples.

Training of the fuzzy integral: This phase of training is essential to know the values of: (a) the partial evaluation $h_k(\boldsymbol{\xi}_s)$, i.e., how good the feature $\boldsymbol{\xi}_s$ alone is to classify the patterns from the class $\mathbf{C}_k$, and (b) the individual fuzzy density $g(\{\boldsymbol{\xi}_s\})$, i.e., the importance of the outputs of the sth subnetwork. The value of $h_k(\boldsymbol{\xi}_s)$ is an indication of how certain we are in the classification of input $\mathbf{x}$ into the class $\mathbf{C}_k$ using the feature $\boldsymbol{\xi}_s$. Here, a one indicates with absolute certainty that the input $\mathbf{x}$ is from the class $\mathbf{C}_k$, and a zero means that the input certainly does not belong to the class $\mathbf{C}_k$. To obtain the value of $h_k(\boldsymbol{\xi}_s)$, a set of features $\{\boldsymbol{\xi}_s\}$ is collected from a set of feature vectors $\{\boldsymbol{\psi}\}$. This set of feature vectors is collected by passing a set of training inputs $\{\mathbf{x}\}$ through all the subnetworks. The set $\{\mathbf{x}\}$ contains training inputs from all the classes. The set of features $\{\boldsymbol{\xi}_s\}$ is fed to a fuzzy K-means clustering algorithm [11]. The cluster centers $\mathbf{m}_k^s$, $k = 1, 2, \ldots, M$, generated by the fuzzy K-means algorithm, are recorded. In the testing phase, these cluster centers will help us to compute the partial evaluation $h_k(\boldsymbol{\xi}_s)$ in the form of fuzzy membership values.

The individual fuzzy densities are calculated based on how well the outputs generated by the subnetworks separate all the classes for the training data. We propose a fuzzy-rough set theoretic approach to determine the individual fuzzy densities. This approach is described below:

While using the fuzzy K-means clustering on the set $\{\boldsymbol{\xi}_s\}$, we can observe the following two points:

1. Some $\boldsymbol{\xi}_s$ belong to more than one cluster partially as the clusters are overlapping.
2. All $\boldsymbol{\xi}_s$ from the same cluster may not belong to the same class.

The first type of uncertainty, known as ***fuzzy uncertainty***, is generated because the outputs of the subnetworks are not from $\{0, 1\}$. It may be due to the fuzziness, if the outputs of the subnetworks are considered to be fuzzy, or it may be due to the lack of confidence, if the outputs of the subnetworks are viewed as ***a posteriori*** probabilities. The second type of uncertainty is known as ***rough uncertainty***, which is generated as the feature $\boldsymbol{\xi}_s$ is not sufficient to classify all the input patterns $\{\mathbf{x}\}$, and hence, two different $\boldsymbol{\xi}_s$ belonging to the same cluster may represent two different classes. Thus, the relationship between the sth feature and the class labels may be a one-to-many mapping. In other words, the classes are ***indescernible*** or not distinguishable with respect to the sth feature. Thus, the sth feature $\boldsymbol{\xi}_s$ is a very important feature if

1. The clusters are compact and wide apart, i.e., the less is the fuzzy uncertainty, the more important the feature is.
2. All the elements from a particular cluster belong to the same class, i.e., the less is the rough uncertainty, the more important the feature is.

In other words, a feature is very important if each cluster generated by the features is compact and isolated, and if all the patterns from each cluster represent the same class. Thus, the more fuzzy and rough the uncertainty is, the less is the importance. We seek to quantify the amount of the fuzzy and rough uncertainty involved by using fuzzy-rough sets. Later we use the quantified value to determine the importance of the sth feature (i.e., importance of the sth module) for the kth class. The lack of discriminatory power of the feature $\boldsymbol{\xi}_s$ is due to the fact that we are not considering the other features $\boldsymbol{\xi}_j$, $j \neq s$, $j = 1, 2 \ldots, S$ into account. Here we do not have complete information to classify a particular pattern in the class $\mathbf{C}_k$ based on the information supplied by $\boldsymbol{\xi}_s$. To determine the importance of the sth feature $\boldsymbol{\xi}_s$ for the class $\mathbf{C}_k$ with such incomplete knowledge, the concept of rough sets is used. In the terminology of rough set, two patterns $\boldsymbol{\psi}_p$ and $\boldsymbol{\psi}_q$ are called ***indiscernible*** with respect to the sth feature when the sth component of these two patterns have the same value. Mathematically, it can be stated as

$$\boldsymbol{\psi}_p \mathbf{R}^s \boldsymbol{\psi}_q \quad \text{iff} \quad \boldsymbol{\xi}_{sp} = \boldsymbol{\xi}_{sq} \tag{12}$$

where $\mathbf{R}^s$ is a binary relation over $\boldsymbol{\Xi} \times \boldsymbol{\Xi}$. Obviously, $\mathbf{R}^s$ is an equivalent relation. Therefore, $\mathbf{R}^s$ partitions $\{\boldsymbol{\psi}\}$ into a set of equivalent classes, namely $\{\mathbf{F}_1^s, \mathbf{F}_2^s, \ldots, \mathbf{F}_K^s\}$ where K is greater than one but less than the cardinality of $\{\boldsymbol{\psi}\}$. However, for continuous features, it is better to consider that $\boldsymbol{\psi}_p \in \boldsymbol{\Psi}$ and $\boldsymbol{\psi}_q \in \boldsymbol{\Psi}$ are related if the sth component of the two features are similar (not necessarily strictly equal as in (12)). Then the resultant equivalence classes are fuzzy clusters. Two patterns from the same cluster are similar as they have spatial similarity. This scenario brings the concept of a fuzzy-rough set. Here one obvious problem is to decide the number of clusters needed for the task. Since for error free classification each class should be represented by one cluster in $\boldsymbol{\Psi}$ space, we can assume that the number of clusters is equal to the number of classes, i.e., $K = M$. It can be shown [6] that the fuzzy clusters $\mathbf{F}_1^s, \mathbf{F}_2^s, \ldots, \mathbf{F}_M^s$ will be present if and only if there exists this kind of similarity relation. Moreover, it can be shown that [6] the generated clusters will follow ***weak fuzzy partitioning*** [6], where the term weak fuzzy partition means that each $\mathbf{F}_j^s$ is a normal fuzzy set, i.e., $\max\limits_{\boldsymbol{\psi}} \mu_{\mathbf{F}_j^s}(\boldsymbol{\psi}) = 1$ and $\inf_{\boldsymbol{\psi}} \max\limits_{j=1,2,\ldots,M} \mu_{\mathbf{F}_j^s}(\boldsymbol{\psi}) > 0$ while $\sup_{\boldsymbol{\psi}} \min\{\mu_{\mathbf{F}_j^s}(\boldsymbol{\psi}), \mu_{\mathbf{F}_j^s}(\boldsymbol{\psi})\} < 1 \quad \forall\, i, j \;\; i \neq j$.

Given a weak fuzzy partition $\{\mathbf{F}_1^s, \mathbf{F}_2^s, \ldots, \mathbf{F}_M^s\}$ on $\{\boldsymbol{\psi}\}$, the aim is to find how important the feature $\boldsymbol{\xi}_s$ is for the class C_k. In other words, it means how good $\boldsymbol{\xi}_s$ is to approximate the crisp class $\mathbf{C}_k$ by using $\{\mathbf{F}_1^s, \mathbf{F}_2^s, \ldots, \mathbf{F}_M^s\}$. It virtually depends on how good each cluster is to approximate the output class $\mathbf{C}_k$. Let us first take the jth cluster into consideration. The extent to which the cluster $\mathbf{F}_j$ is successful in approximating $\mathbf{C}_k$ can be expressed in terms of a lower and an upper approximation $\underline{\mathbf{C}_{k,j}^s}$ and $\overline{\mathbf{C}_{k,j}^s}$ as follows:

$$\begin{aligned}
\mu_{\underline{\mathbf{C}_{k,j}}}(\mathbf{F}_j^s) &= \inf_{\boldsymbol{\psi}} \max\{1 - \mu_{\mathbf{F}_j^s}(\boldsymbol{\psi}), \mu_{\mathbf{C}_k}(\mathbf{x})\} \\
&= \inf_{\boldsymbol{\psi} \notin C_k} \{1 - \mu_{\mathbf{F}_j^s}(\boldsymbol{\psi})\} & \text{(13-a)} \\
\mu_{\overline{\mathbf{C}_{k,j}}}(\mathbf{F}_j^s) &= \sup_{\boldsymbol{\psi}} \min\{\mu_{\mathbf{F}_j^s}(\boldsymbol{\psi}), \mu_{\mathbf{C}_k}(\mathbf{x})\} \\
&= \sup_{\boldsymbol{\psi} \in \mathbf{C}_k} \{\mu_{\mathbf{F}_j^s}(\boldsymbol{\psi})\} & \text{(13-b)}
\end{aligned}$$

$\left\langle \underline{\mathbf{C}_{k,j}}, \overline{\mathbf{C}_{k,j}} \right\rangle$ is called a fuzzy-rough set. Here, $\mu_{\mathbf{C}_k}(\mathbf{x}) \in \{0, 1\}$ is the crisp membership of the input $\mathbf{x}$ to the class $\mathbf{C}_k$.

Any input $\boldsymbol{\psi}$ with $\mu_{\mathbf{F}_j^s}(\boldsymbol{\psi}) \geq \mu_{\underline{\mathbf{C}_{k,j}^s}}(\mathbf{F}_j^s)$ certainly belongs to $\mathbf{C}_k$. Hence, we term the interval $\left[\mu_{\underline{\mathbf{C}_{k,j}}}(\ \boldsymbol{\psi}),\ 1\right]$ as ***certainty interval***. Similarly, we term $\left[\mu_{\overline{\mathbf{C}_{k,j}^s}}(\boldsymbol{\xi}_s),\ 1\right]$ as ***possibility interval***. Because, any training input $\boldsymbol{\psi}$ with $\mu_{\mathbf{F}_j^s}(\boldsymbol{\psi}) \geq \mu_{\overline{\mathbf{C}_{k,j}^s}}(\mathbf{F}_j^s)$ may belong to $\mathbf{C}_k$. Some patterns having membership values in the ***boundary interval*** $\left[\mu_{\overline{\mathbf{C}_{k,j}^s}}(\mathbf{F}_j^s),\ \mu_{\underline{\mathbf{C}_{k,j}^s}}(\mathbf{F}_j^s)\right]$ will be classified to $\mathbf{C}_k$ and some are not, resulting in a one-to-many relationship between the cluster $\mathbf{F}^s_j$ and the class $\mathbf{C}_k$. This interval is responsible for creating fuzzy-roughness in the cluster $\mathbf{F}^s_j$. To quantify the fuzzy-roughness generated due to the boundary interval, let us define the following two fuzzy sets $\underline{C^s_{k,j}}$ and $\overline{C^s_{k,j}} : \Psi \rightarrow [0, 1]$ as follows [12]:

$$\mu_{\underline{C^s_{k,j}}}(\psi) = \mu_{\mathbf{F}^s_j}(\psi) \text{ if } \mu_{\mathbf{F}^s_j}(\psi) \geq 1 - \mu_{\underline{\mathbf{C}_{k,j}}}(\mathbf{F}^s_j) \quad \forall\, \psi \tag{14-a}$$

$$\mu_{\overline{C^s_{k,j}}}(\psi) = \mu_{\underline{\mathbf{F}^s_j}}(\psi) \text{ if } \mu_{\mathbf{F}^s_j}(\psi) \geq \mu_{\overline{\mathbf{C}_{k,j}}}(\mathbf{F}^s_j) \quad \forall\, \psi \tag{14-b}$$

For any $\boldsymbol{\xi}_s \in \Xi$, $\mu_{\underline{C^s_{j,k}}}(\boldsymbol{\xi}_s)$ and $\mu_{\overline{C^s_{k,j}}}(\boldsymbol{\xi}_s)$ can be considered as the degree to which $\boldsymbol{\xi}_s$ certainly or possibly belongs to $\mathbf{C}_k$. We note that $\underline{C^s_{k,j}} \subseteq \overline{C^s_{k,j}}$. Based on the fuzzy-roughness generated in $\mathbf{F}_j$, the importance of $\boldsymbol{\xi}_s$ in $\mathbf{F}_j$ for the class $\mathbf{C}_k$ can be quantified by using a term $\left\|\underline{C^s_{k,j}}\right\| \big/ \left\|\overline{C^s_{k,j}}\right\|$, where $\|\overline{C^s_{k,j}}\|$ means the cardinality of the fuzzy set $\overline{C^s_{k,j}}$. One possible way to determine the cardinality is to use the following definition: $\left\|\overline{C^s_{k,j}}\right\| = \sum_p \mu_{\overline{C^s_{k,j}}}(\psi_p)$. When the certainty interval in $\mathbf{F}_j$ for the output class $\mathbf{C}_k$ is equal to the possibility interval, i.e., $\underline{C^s_{k,j}} = \overline{C^s_{k,j}}$, then the uncertainty to classify a pattern from the class $\mathbf{C}_k$ is the least. It implies that for this case the importance of $\boldsymbol{\xi}_s$ should be maximum, i.e., $\left\|\underline{C^s_{k,j}}\right\| \big/ \left\|\overline{C^s_{k,j}}\right\| = 1$. On the other hand, if the cluster does not classify any pattern from $\mathbf{C}_k$ with certainity, i.e., $\underline{C^s_{k,j}} = \phi$, then $\left\|\underline{C^s_{k,j}}\right\| \big/ \left\|\overline{C^s_{k,j}}\right\| = 0$. It indicates that the importance of the feature $\boldsymbol{\xi}_s$for the class $\mathbf{C}_k$ is zero. Since the importance of $\boldsymbol{\xi}_s$ depends on all the clusters, one possible measure of the importance of $\boldsymbol{\xi}_s$ for the class $\mathbf{C}_k$ is

$$g_k(\{\boldsymbol{\xi}_s\}) = \rho^s_{\mathbf{C}_k} = \frac{\sum_{j=1}^{M} \left\|\underline{C^s_{k,j}}\right\|}{\sum_{j=1}^{M} \left\|\overline{C^s_{k,j}}\right\|} \tag{15}$$

It can be observe that $g_k(\{\boldsymbol{\xi}_s\})$ lies in between 0 and 1. $g_k(\{\boldsymbol{\xi}_s\})$ becomes 0, when no patterns from $\mathbf{C}_k$ is classified to any cluster with some certainty. On the contrary, $g_k(\{\boldsymbol{\xi}_s\})$ becomes 1, when the possibility interval and the certainty interval for each cluster become the same. This situation signifies the formation of isolated and compact clusters, where all patterns from each cluster represent the same class.

The complete training procedure, consisting of subnetworks training and fuzzy integral training, can be expressed in form of an algorithm as follows:

```
Use different training sets $T_s$, $s = 1,2,\ldots,S$ to
train all the subnetworks using the
backpropagation algorithm. The training set
$T_s$ contains the training input-output pairs
only for the $s$th subnetwork.

Prepare another training set $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$,
that contains the training input-output pairs
for all the subnetworks. Pass this training
set through all the subnetworks to
collect the feature vectors $\psi_p$, $p = 1,2,\ldots,N$
as the outputs.

Do for each $s$
    Apply the fuzzy $K$-means clustering
    algorithm with $M$ clusters on
    $\{\boldsymbol{\xi}_{sp} \mid p = 1,2,\ldots,N\}$.
    Record the means $\mathbf{m}^s_j$, $j = 1,2,\ldots,M$ of
    the clusters
    Do for each class $\mathbf{C}_k$, $k = 1,2,\ldots,M$
        Use (15) to compute the fuzzy
        density $g_k(\{\boldsymbol{\xi}_s\})$ from
        $\{\boldsymbol{\xi}_{sp} \mid p = 1,2,\ldots,N\}$.
    End do
End do
```

C. Testing

In this stage, a separate set of test patterns is given as inputs to all the subnetworks. The outputs of all the subnetworks corresponding to the input test pattern $\mathbf{x}$ form the feature vector $\psi = [\boldsymbol{\xi}_1^T, \boldsymbol{\xi}_2^T, \ldots, \boldsymbol{\xi}_S]^T$. To determine the partial evaluation $h_k(\boldsymbol{\xi}_s)$ from the already recorded means, we use the following fuzzy membership assignment relation [11]:

$$h_k(\boldsymbol{\xi}_s) = 1 \Big/ \sum_{h=1}^{M} \left(d_k / d_h\right)^{2/(q-1)} \tag{16}$$

Here, $d_k = \|\boldsymbol{\xi}_s - \mathbf{m}^s_k\|^2 = (\boldsymbol{\xi}_s - \mathbf{m}^s_k)^T \boldsymbol{\Sigma} (\boldsymbol{\xi}_s - \mathbf{m}^s_k)$, where $\boldsymbol{\Sigma}$ is a positive definite matrix and $q \in (1, \infty)$ is an index. Moreover, from the training part the fuzzy densities $g_k(\{\boldsymbol{\xi}_s\})$, $\forall s, k$, are known. Hence, using the equation (4), the fuzzy integral value of $\mathbf{x}$ corresponding to each output class can be computed. The class label corresponding to the test input is the class index which yields the maximum fuzzy integral value. The testing procedure is embodied in form of an algorithm as follows:

```
For the test input pattern x, find the outputs
$\boldsymbol{\xi}_s$, $s = 1,2,\ldots,S$ for all the subnetworks.
Do for each output class $\mathbf{C}_k$, $k = 1,2,\ldots,M$
    Do for each $\boldsymbol{\xi}_s$, $s = 1,2,\ldots,S$
        Compute $h_k(\boldsymbol{\xi}_s)$ from (16).
    End do
    Calculate $\lambda$ from (9).
    Calculate $e_k$ from (4).
End do
The class label of x is $j$ if $e_j = \max_{k=1}^{M}\{e_k\}$.
```

IV. Results and Discussion

The proposed algorithm is tested on a Contract Bridge Bidding example. In Contract Bridge, a player makes a bid to convey information about the pattern of the thirteen cards in his hand. If he is the first to make a bid, it is called "opening bid", which he makes based only on the pattern of the cards he is holding. He has no *a priori* knowledge of the rest of the cards in other players' hands at this stage. The objective of this experiment is to explore the possibility of capturing the reasoning process of a player in making an opening bid based on the pattern of the cards presented to him. This can be interpreted as a task of building a classifier that can classify an input hand pattern into output classes, where each output class corresponds to one bid. This classification process becomes complicated because for a given hand, the same player may make a different bid at a different time.

It has been observed [13] that, compared to a single monolithic network, a modular neural network is more appropriate for the bidding task. The work described in this paper is an attempt along this line. Here, we study the performance of the proposed fuzzy-rough based method and one existing method on a two-module network. We have taken four classes corresponding to three Club (3C), three Diamond (3D), three Heart (3H) and three Spade (3S) card hands. One module is supposed to classify 3C and 3D and the other one 3H and 3S. For our work, Standard American System [13] is used as a bidding convention as it is less artificial than the other systems. Input of the network is represented as a series of 52 one/zero, where the presence or absence of a card is denoted by 1 or 0 [13]. Only one hidden layer with 50 hidden nodes is used in each module. Number of output nodes for both the subnetworks are 2. Both the modules are trained with backpropagation learning law on a training set of 232 card hands. Some negative examples are used in both the modules to fine tune the decision boundaries. The outputs of these two modules are fused by the fuzzy integral approach. To demonstrate the relative performance of the proposed fuzzy-rough method, the densities of the fuzzy integrator are calculated by the frequency-based method (used in [3] [14]) and the proposed method. The fuzzy densities are calculated in [3] [14] by using $g_k(\{\xi_s\}) = p_{s,k} \Big/ \sum_{j=1}^{S} p_{j,k}.d_s$ where $p_{s,k}$ is the classification performance of ξ_s for the class C_k on the validation data and d_s is the desired sum of the fuzzy densities. The comparative classification results on a test set of 60 card hands are given in Table 1. In Table-1, we can observe that the proposed method is performing better than the frequency-based method.

TABLE I
Results of Card Data Classification Based on the Frequency Method and the Proposed Method

Class	Frequency-based Method	Proposed Method
'3C'	80.39%	82.47%
'3D'	74.41%	79.34%
'3H'	82.01%	81.29%
'3S'	80.32%	85.69%
Overall	79.28%	82.65%

Acknowledgment

This work was carried out while Manish Sarkar held a Dr. K. S. Krishnan Research Fellowship from the Department of Atomic Energy, Government of India.

References

[1] C. C. Sekhar, *Neural network models for recognition of stop consonant-vowel (SCV) segments in continuous speech*. PhD thesis, Department of Computer Science and Engineering, Indian Institute of Technology, Madras, December 1996.

[2] P. J. Darwen and X. Yao, "Speciation as automatic categorical modularization," *IEEE Transactions on Evolutionary Computation*, vol. 1, pp. 101–108, July 1997.

[3] H. Tahani and J. K. Keller, "Information fusion in computer vision using fuzzy integral," *IEEE Transactions on System, Man and Cybernetics*, vol. 20, no. 3, pp. 733–741, 1990.

[4] C. Chandra Sekhar and B. Yegnanarayana, "Modular Neural Networks and Constraint Satisfaction Model for Recognition of Stop-Consonant Vowel (SCV) Utterances," *IJCNN-1998.*

[5] Z. Pawlak, "Rough sets," *International Journal of Computer and Information Science*, vol. 11, pp. 341–356, 1982.

[6] D. Dubois and H. Prade, "Putting rough sets and fuzzy sets together," in *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Set Theory* (R. Slowinski, ed.), Dordrecht: Kluwer Academic Publishers, 1992.

[7] M. Sugeno, *Theory of fuzzy integrals and its applications.* PhD thesis, Tokyo Institute of Technology, 1974.

[8] T. Murofushi and M. Sugeno, "An interpretation of fuzzy measure and the Choquet integral as an integral with respect to a fuzzy measure," *Fuzzy Sets and Systems*, vol. 29, pp. 201–227, June 1989.

[9] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning about Data.* Dordrecht: Kluwer, 1991.

[10] G. S. Klir and B. Yuan, *Fuzzy sets and Fuzzy Logic – Theory and Applications.* Englewood Cliffs, NJ: Prentice-Hall, 1995.

[11] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms.* New York: Plenum Press, 1981.

[12] M. Banerjee and S. K. Pal, "Roughness of a fuzzy set," *Information Sciences*, no. 93, pp. 235–246, 1996.

[13] B. Yegnanarayana, D. Khemani, and M. Sarkar, "Neural networks for contract bridge bidding," *Sadhana*, vol. 21, pp. 395–413, June 1996.

[14] S. B. Cho and J. H. Kim, "Multiple network fusion using fuzzy logic," *IEEE Transactions on Neural Networks*, vol. 6, pp. 497–501, March 1995.

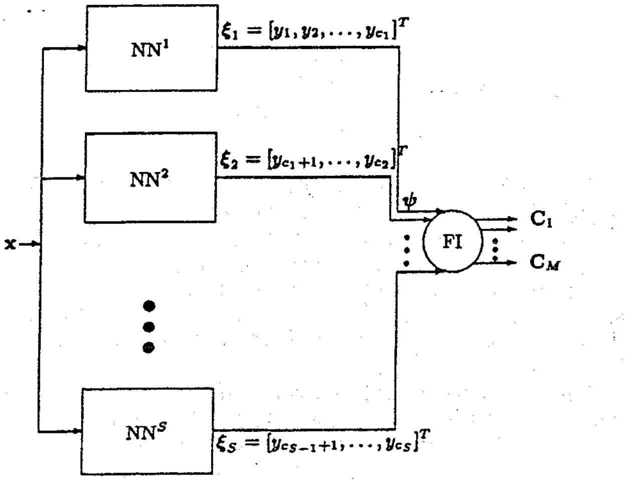
$\xi_1 = [y_1, y_2, \ldots, y_{c_1}]^T$
NN^1
$\xi_2 = [y_{c_1+1}, \ldots, y_{c_2}]^T$
NN^2
ψ
x
FI
C$_1$
C$_M$
NN^S
$\xi_S = [y_{c_{S-1}+1}, \ldots, y_{c_S}]^T$

Fig. 1: A modular network with *s* feedforward subnetworks. The outputs of the subnetworks are fused by a fuzzy integrator (FI) to obtain the final classification result.