International Journal of Computational Intelligence Systems, Vol. 4, No. 5 (September, 2011), 1042-1051

A Novel Fuzzy Rough Granular Neural Network for Classification

Avatharam Ganivada, Sankar K. Pal

Center for Soft Computing Research, Indian Statistical Institute, Kolkata, 700108, India

E-mails: avatharg@yahoo.co.in, sankarpal@yahoo.com

Abstract

A novel fuzzy rough granular neural network (NFRGNN) based on the multilayer perceptron using back-propagation algorithm is described for fuzzy classification of patterns. We provide a development strategy of knowledge extraction from data using fuzzy rough set theoretic techniques. Extracted knowledge is then encoded into the network in the form of initial weights. The granular input vector is presented to the network while the target vector is provided in terms of membership values and zeros. The superiority of NFRGNN is demonstrated on several real life data sets.

Keywords: Granular computing, fuzzy reflexive relation, fuzzy rough sets, rule based layered network, fuzzy pattern classification

1. Introduction

Granular Computing (GC) is a new information-processing paradigm being developed in the past few years. It recognizes that precision is an example in modeling and controlling complex systems. An underlying idea of granular computing is granulation of universe. A granule normally consists of patterns that are grouped together by suitable similarity measure (Ref. 1). When a problem involves incomplete, uncertain, and vague information, it may be difficult to differentiate distinct elements and one may find it convenient to consider granules for its handling.

In general human reasoning and perception are somewhat fuzzy in nature in the sense that their boundaries are not precise and the attributes they can take are granules. Hence, to enable a system to tackle real-life ambiguous situations (Ref. 2) in a manner more akin to humans, one may incorporate the concept of granules into the neural networks, a biologically inspired computing paradigm. Zhang et al (Ref. 21) described granular neural networks using fuzzy sets as their formalism and an evolutionary training algorithm. One of the neuro-fuzzy systems for classification, named fuzzy neural network, was developed by Pal and Mitra (Ref. 3). As part of determining the initial weights, Banerjee et al. (Ref. 5) described a knowledge based network, where knowledge is extracted from data in the form of decision rules using rough set theoretic techniques. Recently, fuzzy sets have been integrated with neural networks to simplify the knowledge representation in a neural network (Ref. 6). Several studies have been made combining fuzzy sets (Ref. 7) and rough sets (Ref. 8). Many relationships have been established to extend and integrate the underlying concepts of these two methodologies to deal with additional aspects of data imperfection, especially in the context of granular computing. The main purpose of fuzzy and rough hybridization is to provide high degree of flexibility (Ref. 9), robust solutions (Ref. 10), and handling uncertainty (Ref. 11).

We use the theory of granulation structures in fuzzy rough sets (Ref. 12, 13) based on fuzzy reflexive relation, defining the dependency factors of all the attributes w.r.t. each decision class in a decision table. Fuzzy-rough sets based on fuzzy reflexive relation provide a means by which discrete or real-valued noisy data can be effectively reduced without any threshold values for data analysis. The granulation structure produced by an fuzzy T-equivalence class provides a partition of the universe. Its intension is to approximate an imprecise concept in the domain of universe by a pair of approximation concepts, called lower and upper approximations. These approximations are used to define the value of positive degree of each object and all the positive degrees are used to determine dependency degree of each conditional attribute. The syntax of the dependency degrees is to define the initial weights of the network.

In this article, we have made an attempt to integrate fuzzy rough sets with a fuzzy neural network for designing a new application of GC, namely, the development of a three layered novel fuzzy rough granular neural network (NFRGNN). It may be noted that the network knowledge encoding procedure, unlike the most other methods (Ref. 5, 6), involving appropriate number of hidden nodes is determined by the number of decision classes. The dependency factor of each conditional attribute, and average value of dependency factors of all the conditional attributes w.r.t. each decision class are encoded between the nodes of input and hidden layers, hidden and output layers, respectively, in the network as initial weights. The components of the input vector consist of membership values to the overlapping partitions of linguistic properties low, medium or high corresponding to each input feature. This provides a scope for incorporating granular information in both the training and testing phases of the network. It increases robustness of the network in tackling uncertainty. Performance of the network is measured in terms of percentage accuracy and Macro averaged measure. The characteristics of NFRGNN have demonstrated with eight real life data sets and compared with that of fuzzy MLP (Ref. 3), robust fuzzy granular neural network (Ref. 4), and rough fuzzy MLP (Ref. 5).

2. NFRGNN Architecture

In this section, we describe a NFRGNN architecture based on multilayer perceptron using back propagation algorithm. A three layered NFRGNN is considered with the nodes of input layer consist of the 3n-dimensional attribute values and the output layer is represented by c-decision classes. The hidden layer nodes are determined based on decision classes. Each neuron of the hidden layer is fully connected to neurons in the next layer and in the previous layer. The granular input vector is supplied to the network by placing it at the neurons in the input layer, the network outputs consist of the activation values of the neurons in the hidden and output layers. The input layer is composed of non-computational units. Each such unit receives a single input and distribute it to all the neurons in the next layer via synaptic weights.

2.1. NFRGNN Back Propagation Algorithm

Input

D, a data set consisting of the training tuples in the granular form and their associated target vector in terms of membership value and zeros.

η, the learning rate

α, the momentum term

$b = b_j$, bias b is a constant at each node j.

network, a granular feed-forward network.

Output

A trained neural network

Method

1. Initial weights are determined among the nodes (units) of all layers in the network by fuzzy rough sets based on fuzzy reflexive relation;
2. While terminating condition is not satisfied{

3. for each training tuple
   Propagate the inputs forward:
4. for each unit j of input layer {
5. $O_j = I_j$; here, the output of an input unit is its actual input value.
6. for each unit j of hidden or output layer, compute the net input of each unit j with respect to the previous layer, i {
7. $I_j = \sum_i w_{ji}.O_i + b$; }
8. Apply logistic activation function to compute the output of each unit j.
9. $\phi(O_j) = \frac{1}{1+e^{-I_j}}$
   Back Propagation:
10. for each unit j in the output layer, compute the error {
11. $Error_j = (\phi(O_j).(1-\phi(O_j))).(T_j - \phi(O_j))$;
12. for each unit j in the hidden layer, compute the error with respect to the next higher layer (output layer).
13. $\gamma_j = (\phi(O_j).(1\text{-}\phi(O_j))).\left(\sum_k Error_k.w_{jk}\right)$; }
14. for each weight w_{ij} in network {
15. $\Delta w_{ij} = (\eta.x_i).\gamma_j$;
    where η is network wise learning rate parameter.
16. $\Delta w_{ij}(k) = ((\eta.x_i(k)).\gamma_j(k)) + (\alpha.\Delta w_{ij}(k-1))$; }
17. for each constant bias b in network {
    $\Delta b = \eta.\gamma_j$;
    b+ =Δb;} }

where α is a momentum parameter used to escape local minima in weight space and k is a number to denote an epoch (i.e., k-1 denotes the previous epoch).

3. Input Pattern Representation in Granular Form

In general, human minds can perform a wide variety of physical and mental tasks without any measurement or computation. Familiar examples of such tasks are parking a car, driving in heavy traffic, and understanding speech. Based on such tasks perceptions of size, distance, weight, speed, time, direction, smell, color, shape force etc occur. But a fundamental difference between such measurements on one hand and perception on the other, is that, the measurements are crisp numbers whereas perceptions are fuzzy numbers or more generally, fuzzy granules (Ref. 14).

Fuzzy granule is formally defined as a group of objects defined by the generalized constraint form "X isr R" where 'R' is constrained relation, 'r' is a random set constraint, which is a combination of probabilistic and posibilistic constraints, 'X' is a fuzzy set valued random variable which takes the values low, medium or high. Using fuzzy-set theoretic techniques (Ref. 15) a pattern point x, belonging to the universe U, may be assigned a grade of membership with the membership function $\mu_A(x)$ to a fuzzy set A. This is defined as

$$A = \{(\mu_A(x), x)\}, \quad x \in U, \quad \mu_A(x) \in [0,1]. \quad (1)$$

The π membership function, with range [0, 1] and $x \in \mathbb{R}^n$, is defined as

$$\pi(x,c,\lambda) = \begin{cases} 2(1-\frac{\|x-c\|}{\lambda})^2, & for \quad \frac{\lambda}{2} \leqslant \| x-c \| \leqslant \lambda, \\ 1-2(\frac{\|x-c\|}{\lambda})^2, & for \quad 0 \leqslant \| x-c \| \leqslant \frac{\lambda}{2}, \\ 0, & otherwise, \end{cases} \quad (2)$$

where $\lambda > 0$ is the radius of the π function with c as the central point, and $\|\ \ \|$ denotes Euclidian norm. Each input feature F_j can be expressed in terms of membership values to each of the three linguistic properties low, medium and high as granules. Therefore, an n-dimensional pattern can be represented as a 3n-dimensional vector

$$\vec{F}_i = [\mu_{low(F_{i1})}(\vec{F}_i), \mu_{medium(F_{i1})}(\vec{F}_i), \mu_{high(F_{i1})}(\vec{F}_i).. \ldots, \mu_{high(F_{in})}(\vec{F}_i)]. \quad (3)$$

3.1. Choice of Parameters of π-functions for Numerical Features

When the input feature is numerical, we use π fuzzy set of Eq. (2) with appropriate parameter values for center c and radius λ. These values are chosen as

explained in the article of Pal and Mitra (Ref. 3). These are used to express features of each input pattern in terms of membership values to each of three fuzzy granules low, medium, or high.

3.2. Class Memberships as Output Vectors

Consider a c-class problem domain such that we have c-nodes in the output layer. Let the n-dimensional vectors O_{kj} and V_{kj} denote the mean and standard deviation, respectively, of the training data for the k^{th} class. The weighted distance of the training pattern $\overrightarrow{F_i}$ from the k^{th} class is defined as

$$Z_{ik} = \sqrt{\sum_{j=1}^{n} \left[\frac{F_{ij}-O_{kj}}{V_{kj}}\right]^2}, \quad for \quad k = 1,2,\ldots,c, \tag{4}$$

where F_{ij} is the value of the j^{th} component of the i^{th} pattern point. The membership of the i^{th} pattern to class c_k, lying in the range [0, 1], is defined by

$$\mu_k(\overrightarrow{F}_i) = \frac{1}{1+(\frac{Z_{ik}}{f_d})^{f_e}}, \tag{5}$$

where Z_{ik} is the weighted distance from Eq. (4), and f_d, f_e are the denominational and exponential fuzzy generators controlling the amount of fuzziness in the class membership. In the fuzziest case, we may use fuzzy modifier contrast internification (INT) from (Ref. 15) to enhance contrast within the class membership to decrease the ambiguity in taking a decision.

3.3. Applying the Membership Concept to Target Vector

The target vector at the output layer is defined by membership values and zeros as shown in Eq. (6). For the patterns belonging to a particular class, the desired vectors of those patterns at the corresponding class node are assigned membership values, and the rest of the class nodes are assigned zeros. For i^{th} input pattern, we define the desired output of the j^{th} output node as

$$d_j = \begin{cases} \mu_{INT(j)}(\overrightarrow{F_i}), & for \ i^{th} pattern \ at \ the \ j^{th} output \ node, \\ 0, & otherwise. \end{cases} \tag{6}$$

4. Preliminaries in Granulations and Approximations

The granulation structure used in rough set theory is typically a partition of the universe. For preliminaries of rough set theory, one may refer to (Ref. 8).

In the context of fuzzy rough set theory, fuzzy set theory (Ref. 7) allows that objects belong to a set, and couple of objects belong to a relation, to a given degree. Recall that Eq. (1) defines a fuzzy set in U. A fuzzy relation R in U is a mapping $U \times U \rightarrow [0,1]$, where this mapping is expressed by a membership function $R(x,y)$ of a relation R. i.e., $R = \{((x,y), R(x,y)) \mid (R(x,y)) \in [0,1], x \in U, y \in U\}$. For each y in U, the R-foreset of y is a fuzzy set R_y defined by $R_y(x) = R(x,y)$, for all x in U.

In fuzzy rough set theory, the similarity of objects in U is modeled by a fuzzy reflexive relation R, which is defined by

$$R(x,x) = 1, \quad \text{(reflexive)},$$

for all x, y, z in U and a given T-norm, R is then called a fuzzy T- equivalence relation. It may be noted that fuzzy T- equivalence relations do not necessarily display fuzzy T-symmetric relation and T-transitive relation. In general, for a T- equivalence relation R, we call R_y as the fuzzy T-equivalence class (fuzzy equivalence granule) of y. The fuzzy logical connectives and fuzzy T- equivalence relations play an important role in the generalization of lower and upper approximations of a set in fuzzy rough set theory. Further details about fuzzy logical connectives can be found (Ref. 13). In this article, we use the Lukasiewicz implicator (I_L) from (Ref. 13) to calculate the lower approximation of each object of the concept. A mapping $I : [0,1] \times [0,1] \rightarrow [0,1]$ is defined by $I(0,0) = 1$, $I(1,x) = x$, $\forall x \in [0,1]$, where I is an implicator. For all $x, y \in [0,1]$, an implicator I_L is defined as

$$I_L(x,y) = \min(1, 1-x+y).$$

4.1. Fuzzy Rough Sets

Hybridization of fuzzy sets and rough sets has been addressed by several researchers (Ref. 9, 10, 11, 12, 13, 16). The following two principles were used to fuzzyfication of lower and upper approximations of a concept in rough sets.

- The set A may be generalized to a fuzzy set in U, allowing that objects can belong to a given concept (i.e. subset of the universe) to varying membership degrees in [0, 1].
- Usually, 'objects indistinguishability'is described by means of an equivalence relation R in U in Pawlak rough approximation. Instead of assessing 'objects indistinguishability'in Pawak rough approximation, we may measure their approximation equality represented by a fuzzy reflexive relation R in generalized approximation space. As a result, objects are categorized into classes or granules with "soft" boundaries based on their fuzzy reflexive relation to one another.

In fuzzy rough set analysis, an information system is a couple $(U, \mathscr{A})$, where $U = \{x_1, x_2, ..., x_m\}$ and $\mathscr{A} = \{a_1, a_2, ..., a_n\}$ are finite non empty sets of objects and conditional attributes, respectively. In this article, the values of conditional attributes can be quantitative (real valued). A decision system $(U, \mathscr{A} \cup \{d\})$ is a special kind of information system in which d $(d \notin \mathscr{A})$ is called a decision attribute and it can be qualitative (discrete - valued). Based on these values, the set U is partitioned into non-overlapping fuzzy sets (called concepts) corresponding to decision concepts $R_d(X_k)$, $k = 1, 2, 3, \ldots, c$, where each decision concept represents a decision class. Each object x_i in U is classified by decision classes.

Let $'d'$ be a quantitative attribute in $\mathscr{A}$, we express the fuzzy reflexive relation R_a between any two objects x, y in U w.r.t. an attribute 'a' as

$$R_a(x,y) = \begin{cases} max\left(min\left(\frac{a(y)-a(x)+\sigma_{a_{k_1}}}{\sigma_{a_{k_1}}}, \frac{a(x)-a(y)+\sigma_{a_{k_1}}}{\sigma_{a_{k_1}}}\right), 0\right), \\ \qquad if \quad a(x), a(y) \in R_d(X_{k_1}), \\ max\left(min\left(\frac{a(y)-a(x)+\sigma_{a_{k_2}}}{\sigma_{a_{k_2}}}, \frac{a(x)-a(y)+\sigma_{a_{k_2}}}{\sigma_{a_{k_2}}}\right), 0\right), \\ \qquad if \quad a(x) \in R_d(X_{k_1}), a(y) \in R_d(X_{k_2}), \\ \qquad\quad and \quad k_1 \neq k_2, \end{cases} \tag{7}$$

where k_1, k_2=1, 2, ..., c. For a qualitative attribute $'a'$ in {d}, two methods are described as follows.

(i) *Method I (Crisp case): Crisp way of defining decision classes:*

$$R_a(x,y) = \begin{cases} 1, & if \quad a(x) = a(y), \\ 0, & otherwise, \end{cases} \tag{8}$$

for all x, y in U. The crisp valued decision class implies that objects in the universe U corresponding to the decision equivalence granule will take value only from the set $\{0, 1\}$. A fuzzy set $A \subseteq U$ can be approximated only the information contained within an attribute $\mathscr{A}$ by constructing lower and upper approximations of A w.r.t. crisp decision classes $[\widetilde{x}]_{R_d}$.

Generally, in real life problems, the data are ill-defined with overlapping class boundaries. Each pattern used in a fuzzy set $A \subseteq U$ may possess nonzero belongingness to more than one class. To model such data we extend the concept of crisp decision granule into fuzzy decision granule by inclusion of fuzzy concepts to crisp decision granule. The fuzzy membership values, lying in the range [0, 1], of objects in the universe U w.r.t. each decision class are defined as follows.

(ii) *Method II (Fuzzy case): Fuzzy way of defining decision classes:*

As we have discussed in Section 3.2, A decision system contains c-decision classes of a decision attribute. The mean and standard deviation of an n-dimensional vectors are O_{kj} and V_{kj}, respectively, of the data for the k^{th} class in the given decision system. The weighted distance of the pattern $\overrightarrow{F_i}$ from the k^{th} class is defined in Eq. (4). The weight $\frac{1}{V_{kj}}$ from Eq. (4) is used to take care of the variance of the classes such that a feature with less variance has high weight (significance) in characterizing a class. Moreover, when the value of all the features of a class is same, the standard deviation will be zero. In that case, we consider V_{kj} = 0.000001 such that the weighting coefficient becomes high. When the weighting coefficient is high, pattern membership value is decreased, so does the initial weights. This decrease does mean that the weights will converge to a local minima within less number of epochs.

The membership of the i^{th} pattern to decision class c_k is defined in Eq. (5). Note that, while defining a fuzzy decision attribute the value of fuzzification parameters f_d, f_e is kept 1 throughout implementation for all the data sets. It gives the lower membership value of patterns in its decision class.

When the i^{th} pattern $\overrightarrow{F}_i$ has its own membership degree in its decision class, the decision attribute becomes quantitative. Calculation of a degree of dependency of each conditional attribute w.r.t. quantitative decision will be resulted in high computational complexity. For this reason, the quantitative decision attribute could be made as qualitative decision attribute (like crisp case) by calculating average membership degree of all the patterns within a decision class and assigning it to all the patterns in a decision class. So the average membership value of i^{th} pattern to the k^{th} class c_k is defined as

$$\mu(D_{rk}) = \frac{\sum_{i=1}^{m_k} \mu(\overrightarrow{F}_{ik})}{|m_k|}, \quad for \quad r,k = 1,2,\ldots,c, \tag{9}$$

where $|m_k|$ indicates the number of patterns within the k^th class. For a qualitative attribute 'a' $\in$ {d}, the fuzzy decision relation is defined as

$$R_a(x,y) = \begin{cases} \mu'(D_{rk}), & if \quad a(x) = a(y), \\ \mu(D_{rk}), & otherwise, \end{cases} \tag{10}$$

for all x and y in U. Where $\mu'(D_{rk})$ is used to characterize patterns belong to the same class $(r = k)$ with an average membership degree, and $\mu(D_{rk})$ is used to characterize the patterns from other class $(r \neq k)$ with an average membership degree. A decision class is fuzzy valued implies that objects in the universe U corresponding decision class would take values only from the set [0, 1].

For the lower and upper approximation of a fuzzy set $A \subseteq U$ by means of a fuzzy reflexive relation R, given fuzzy logic connectives: a t-norm T, and an implicator I, we use the definitions given in (Ref. 17),

$$(R \downarrow A)(y) = \inf_{x \in U} I(R(x,y), A(x)), \tag{11}$$

$$(R \uparrow A)(y) = \sup_{x \in U} T(R(x,y), A(x)), \tag{12}$$

for all y in U, where a fuzzy reflexive relation R is used to measure the approximate equality between any two objects in U. The fuzzy positive region can be defined based on fuzzy B-indiscernibility relation as, for $y \in U$,

$$POS_B(y) = \left(\bigcup_{x \in U} R_B \downarrow R_d x \right)(y), \tag{13}$$

for all y in U. The positive region of a fuzzy set is the maximum membership degree with which a unique class can be classified by fuzzy set. The above Eq. (14) can be simplified as follows. For y in a fuzzy set A which is subset of U, $R_d(x)$ is either crisp set or fuzzy set, the fuzzy positive region can be defined as

$$POS_B(y) = (R_B \downarrow R_d x)(y). \tag{14}$$

The degree of dependency of d on the set of attributes $B \subseteq \mathscr{A}$ is defined by

$$\gamma_B = \frac{\sum_{x \in U} POS_B}{|U|}, \tag{15}$$

where $|\quad|$ denotes the cardinality of the set U, and the value of γ is $0 \leqslant \gamma \leqslant 1$. We say a fuzzy set $A \subseteq U$ completely depends on B if γ=1. It may be noted that initial weights in fuzzy case than crisp case of the proposed neural network play significant role to resolving the overlapping class fuzzy boundaries.

5. Network Configuration Using Fuzzy Rough Sets

In this section, we first show how the decision table can be used to explain the concept of granulation by partitioning the universe and approximations of that partition based on fuzzy rough rflexive relation. Based on this principle the initial weights of the network are determined. During training, this network searches for the set of connection weights that corresponds to some local minima. It is to be noted that there may be a large number of such minimum values corresponding to various good solutions. If we initially set weights of the network so as to be near one such solution, the searching space may be reduced and learning thereby becomes faster. The knowledge encoding procedure is defined below in brief.

5.1. Method

Let $S = (U, \mathscr{A} \cup \{d\})$ be a decision table, with $\mathscr{A}$ = $\{a_1, a_2, \ldots..a_n\}$ its set of conditional attributes, and decision attributes {d}, where $U = \{x_1, x_2, \ldots, x_m\}$ its set of objects form c-classes and objects having

labeled values corresponding to each n-dimensional conditional attribute.

Knowledge encoding procedure:

Let us consider the case of feature F_j for a decision table S. Inputs for i^{th} representative pattern F_i are mapped into corresponding a 3-dimensional feature space of $\mu_{low(F_{i1})}(\overrightarrow{F}_i), \mu_{medium(F_{i1})}(\overrightarrow{F}_i), \mu_{high(F_{i1})}(\overrightarrow{F}_i)$ by (3). In this manner an n-dimensional attribute valued decision table can be formed into a $3n$-dimensional attribute valued decision table. The following Steps are applied to the decision table S.

Step 1: Obtain additional granulation structures using the fuzzy reflexive relation, defined in Eq. (7), on each conditional attribute by generating fuzzy relational matrix.

Step 2: Use Step 1 to compute lower approximations, defined in Eq. (11), of each concept for each conditional attribute w.r.t. decision classes, defined in Eqs. (8) or (10), using fuzzy logic connective (Lukasiewicz implicator).

Step 3: Calculate the fuzzy positive region, defined in Eq. (14), of each object for each conditional attribute.

Step 4: Calculate the degree of dependency, defined in Eq. (15), of each conditional attribute corresponding to objects within the concept w.r.t. each decision class (decision concept). Then the resulting values are determined as initial weights between nodes of the input layer and the hidden layer.

Step 5: Calculate the average value of all the degree of dependencies of conditional attributes corresponding to objects within the concept w.r.t. each decision class (decision concept).Then the resulting average values are encoded in the form of initial connection weights between nodes of the hidden layer and the output layer.

We proceed to description of the initial weight encoding procedure. Let the degree of dependency of conditional attribute of a decision table S, for instance, be γ_i, $i \in A=\{a_1, a_2, \ldots a_n\}$. Given c- decision classes for a decision table S, for instance, be $\{d_1, d_2, d_3 \ldots\ldots d_c\}$. The weight w_{ki} between a input node i and hidden node k is defined as follows.

$$\gamma_i^k = \frac{\sum_{x \in U_{d_k}} POS_i(x)}{|U_{d_k}|}, \quad k = 1, 2, \ldots . c. \tag{16}$$

Let β_l denote the average dependency degree of each concept w.r.t. decision class k, l= $1, 2, \ldots c$.

$$\beta_l = \frac{\sum_{i=1}^{n} \gamma_i^k}{|n|}, \quad k = 1, 2, \ldots . c. \tag{17}$$

The weight w_{kl} between hidden node k and output node l is defined as $\frac{\beta_l}{|k|}$.

6. Experimental Results

The proposed NFRGNN algorithm has been implemented in C. The knowledge extraction procedure and performance of NFRGNN are demonstrated on several real life data sets in this section.

6.1. Data Sets

We describe different characteristics of several real life data sets in Table 1.

Table 1. Data set characteristics.

Dataset	# Patterns	# Features	# Classes	Origin
Telugu vowel	871	3	6	(Ref. 18)
Sonar	208	60	2	UCI
GLASS	214	9	6	UCI
Image Segmentation	2310	19	7	UCI
SPECTF Heart	267	44	2	UCI
Letter Recognition	20000	15	26	UCI
Spam Base	4601	57	2	UCI
Web data	145	2556	5	(Ref. 19)

The speech data "vowel" deals with 871 Indian Telugu vowel sounds (Ref. 18). All the other data sets like sonar, glass are taken from the UCI Machine Learning Repository[a], and the Web data set is from (Ref. 19), where the task was to classify web pages based on their content into one of several predefined categories

[a] http://archive.ics.uci.edu/ml//datasets.

6.2. Performance Evaluation Measure

In order to evaluate the performance of NFRGNN, we have used the following performance measure.

6.2.1. Macro averaged f_1 measure

Macro averaged f_1 is derived from precision and recall (Ref. 20). The precision (p_k),recall (r_k) of a class k are defined below.

$$p_k = \frac{\#\text{ patterns correctly classified into class k}}{\#\text{ patterns classified into class k}}, \tag{18}$$

$$r_k = \frac{\#\text{ patterns correctly classified into class k}}{\#\text{ patterns that are truely present in class k}}. \tag{19}$$

Then $(f_1)_k$, the harmonic mean between precision and recall, of class k is defined as

$$(f_1)_k = \frac{2 \times p_k \times r_k}{p_k + r_k}, \tag{20}$$

where $(f_1)_k$ gives equal importance to both precision and recall. It is computed by first computing the $\mathbf{f}_1$ scores for each class(category) and then averaging all these scores to compute the global means. The value of Macro f_1 lies between 0 and 1, and more close the value of Macro f_1 to 1 is the better classification of the data set.

6.2.2. Results

As mentioned earlier, in order to demonstrate our proposed model, during learning, for the data sets except Web, as shown in Table 1, we have selected randomly 10% data from each representative class for training set. In each case the remaining percentage (90%) of data is used as the test set. In case of Web data, we have selected randomly 30% data from each representative class for training set and the remaining percentage (70%) of data is used as the test set since the number of attributes for the Web data set is high. In our experiments, the parameters f_d, f_e in Eq. (4) were chosen as f_d=6, f_e=1 for Telugu vowel data set. Similarly for other data sets, appropriate integer values were chosen. However, the momentum parameter α, learning rate η, and bias b traverses a range of values between 0 and 1 and finally we put α= 0.08, η=0.04 and b=0.9 in crisp case, and α= 0.09, η=0.05, b=0.25 in fuzzy case for good results. It is observed that NFRGNN converges to local minima at 1000^{th} epoch, 800^{th} epoch, in fuzzy case, crisp case, respectively, for Telugu vowel data and SPECTF heart data. Similarly, in both fuzzy and crisp case, it converges to local minima, at 2000^{th} epoch, at 3000^{th} epoch, at 1000^{th} epoch for glass data, spam base data, Web data, respectively.

Knowledge extraction for Telugu vowel data:

A decision table S=$(U, \mathscr{A} \cup \{d\})$ is used to represent train data. Data in the decision table is transformed into a 3-dimensional granular space using Eq. (3). We apply the knowledge encoding procedure (described in Section 5) to the decision table S. The resulting knowledge is encoded into NFRGNN in the form of the initial connection weights. Then the network learns in the presence of the training data set. We present the initial connection weights of NFRGNN in *fuzzy case* and *crisp case* for Telugu vowel data in Tables 2 and 3.

Table 2. NFRGNN with initial connection weights in fuzzy case for Telugu vowel.

Input to Hidden Layer(w_{ki})								
0.11	0.08	0.12	0.12	0.01	0.12	0.11	0.07	0.11
0.11	0.06	0.11	0.10	0.07	0.10	0.09	0.06	0.09
0.03	0.02	0.03	0.07	0.07	0.07	0.03	0.04	0.03
0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
0.16	0.15	0.16	0.21	0.16	0.21	0.17	0.16	0.17
0.02	0.02	0.02	0.05	0.01	0.05	0.06	0.02	0.06

Table 3. NFRGNN with initial connection weights in fuzzy case for Telugu vowel.

Hidden Layer(w_{ji}) to Output Layer (w_{kj})					
0.017	0.017	0.017	0.017	0.017	0.017
0.014	0.014	0.014	0.014	0.014	0.014
0.006	0.006	0.006	0.006	0.006	0.006
0.001	0.001	0.00	0.001	0.001	0.001
0.029	0.029	0.029	0.029	0.029	0.029
0.005	0.005	0.005	0.005	0.005	0.005

Similarly, the same procedure of knowledge extraction can also be applied to the other real life data sets. It may be noted that NFRGNN with initial weights in random real numbers case can be referred to as fuzzy MLP (Ref. 3) where the initial weights are real numbers between the interval [-0.5, 0.5]. Table 4 shows the experimental results of NFRGNN for all the aforesaid real-life data sets.

Table 4. Experimental results for NFRGNN.

Dataset name	Initial Weights	Average Accuracy	Average Precision	Average Recall	Macro f_1
Telugu vowel	Random	83.08	0.82	0.80	0.81
	Crisp	85.38	0.85	0.82	0.83
	Fuzzy	85.90	0.86	0.83	0.84
Sonar	Random	73.12	0.73	0.72	0.72
	Crisp	74.73	0.75	0.74	0.74
	Fuzzy	75.81	0.76	0.75	0.75
Glass	Random	62.63	0.56	0.55	0.55
	Crisp	63.68	0.57	0.55	0.55
	Fuzzy	67.37	0.59	0.56	0.57
Image Segmentation	Random	93.33	0.93	0.93	0.93
	Crisp	93.33	0.93	0.93	0.93
	Fuzzy	93.76	0.94	0.94	0.94
SPECTF Heart	Random	78.66	0.65	0.59	0.61
	Crisp	80.75	0.70	0.59	0.61
	Fuzzy	82.01	0.74	0.62	0.64
Letter Recognition	Random	79.65	0.81	0.80	0.80
	Crisp	73.13	0.76	0.73	0.73
	Fuzzy	80.70	0.82	0.81	0.81
Spam base	Random	83.07	0.84	0.83	0.83
	Crisp	88.00	0.87	0.86	0.87
	Fuzzy	88.38	0.88	0.87	0.88
Web data	Random	48.00	0.35	0.36	0.35
	Fuzzy	57.00	0.73	0.48	0.50

Examining the experimental results from Table 4, for Telugu vowel data, initial weights in random case, crisp case and fuzzy case, NFRGNN provides the significant accuracy of 83.08%, 85.38%, 85.98%, respectively. The optimal performance evaluation measures of NFRGNN with initial weights in random, crisp and fuzzy cases are 0.81, 0.83, 0.84, respectively. For all the other data sets the corresponding figures can be found from the remaining part of the experimental results. An observation was seen that fuzzy MLP with fifty hidden nodes gives better result than NFRGNN where weights in crsip case with twenty six hidden nodes for letter recognition data. However NFRGNN where weights in fuzzy case gives the superior result than that of fuzzy MLP. It was seen that NFRGNN with five hidden nodes (crisp case) is unable to classify the web data from all the classes. Hence these results are not included in Table 4. The experimental results show that the performance of NFRGNN (initial connection weights in both fuzzy and crisp cases), in terms of percentage accuracy, macro averaged f_1 measure, is superior to fuzzy MLP (initial weights of NFRGNN in random case) for all data sets. These figures of NFRGNN with initial weights in fuzzy case indicate that the overlapping between the input patterns of the classes are much resolved than that of initial weights in crisp and random cases.

The performance of our method is compared with well known existing methods: one is robust fuzzy granular neural network (RFGNN) from (Ref. 4) and other is rough fuzzy MLP (rfMLP) from (Ref. 5), we apply on Telugu vowel data, as an example. In rfMLP, Method I produces four reducts combing all the 6 classes, and each reduct represents a set of six decision rules corresponding to 6 vowel classes. In Method II, we have considered one reduct for each class representing its decision rule; thereby generating six decision rules for six vowel classes. The results of rfMLP with three layered knowledge based network and RFGNN for Telugu vowel at the end of 1500^{th} epoch are presented in Table 5.

Table 5. Recognition scores for FRGNN, rfMLP and NRFGNN.

Dataset name	Method	Average Accuracy	Average Precision	Average Recall	Macro f_1
Telugu vowel	RFGNN	81.79	0.76	0.84	0.78
	rfMLP (Method I)	83.97	0.82	0.81	0.81
		82.95	0.80	0.79	0.80
		82.69	0.81	0.79	0.80
		83.08	0.81	0.79	0.80
	(Method II)	83.08	0.81	0.80	0.80
	NFRGNN (Fuzzy Case)	85.90	0.86	0.83	0.84

For Telugu vowel data, the recognition scores for RFGNN and rfMLP are 81.79% and 83.97% respectively. In contrast, NFRGNN gives the maximum recognition score of 85.98%. The optimal performance evaluation measures of RFGNN, rfMLP and NFRGNN are 0.78, 0.81, 0.84, respectively. Based on these results, we can say that the performance of NFRGNN (initial connection weights in fuzzy case), in terms of percentage accuracy, macro averaged f_1 measure, is superior to rfMLP and RFGNN. It was observed that the time complexity of NFRGNN is less than rfMLP and RFGNN for all the data sets.

Finally, we can conclude that the difference among the NFRGNN, rfMLP and RFGNN, as stated above, is likely to be a statistical significant.

7. Conclusion

In this paper, we have presented the design of a novel granular neural network architecture by integrating fuzzy rough sets with multilayer perceptron using back propagation algorithm. We have examined two special types of granular computations. One is induced by low, medium, or high fuzzy granules and the other one is classes of granulation structures induced by a set of fuzzy equivalence granules based on a sequence of fuzzy reflexive relations. With respect to classes of granulation structures, one can obtain stratified fuzzy rough set approximations that can be used to determine the dependency factors of all conditional attributes to obtain initial weights of our proposed network. The incorporation of granule concepts at input and initial weights stages, and membership values at output stage of the conventional MLP also helps the resulting NFRGNN to efficiently handle uncertain and ambiguous information.

The performance of NFRGNN for fuzzy classification of real data sets is found to be superior to rfMLP and RFGNN trained on the Telugu vowel data, as an example. The NFRGNN architecture is a useful application of granular computing to real world classification problems.

References

1. L. A. Zadeh, "Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic," *Fuzzy Sets and Systems*, **19**, 111-127, (1997).
2. L. A. Zadeh, "Fuzzy logic, neural networks, and soft computing," *Commn. ACM*, **37**, 77-84, (1994).
3. S. K. Pal and S. Mitra, "Multilayer perceptron, fuzzy Sets, and classification," *IEEE Trans. on Neural Networks*, **3**, 683-697, (1992).
4. G. Avatharam and S. K. Pal, "Robust granular neural network, fuzzy granules and classification," *Proc. LNCS, Springer Verlag*, **6401**, 220-227, (2010).
5. M. Banerjee, S. Mitra, and S. K. Pal, "Rough fuzzy MLP: knowledge encoding and classification," *IEEE Trans. on Neural Networks*, **9**, 1203-1216, (1998).
6. S. Disk and A. Kandel, "Granular computing in neural networks in Granular computing: An emerging paradigm," *W. Pedrycz, Ed., NY: Physica Verlag*, 275-305, (2001).
7. L. A. Zadeh, "Fuzzy sets," *Inf. Control*, **8**, 338-353, (1965).
8. Z. Pawlak, "Rough Sets: Theoretical aspects of reasoning about data," *Dordrecht, The Netherlands: Kluwer*, (1991).
9. D. Dubios and H.Prade, "Rough fuzzy sets and fuzzy rough sets," *Int. Journal of General Systems*, **17**, 91-209, (1990).
10. P. Lingras and R. Jensen, "Survey of rough and fuzzy hybridization," *Proc. Intl. Conf. on Fuzzy Systems*, 1-6, (2007).
11. M. Banerjee and S. K. Pal, "Roughness of a fuzzy set," *Information Sciences*, **93**, 235-246, (1996).
12. R. Jensen and Q. Shen, "New approaches to fuzzy-rough feature selection," *IEEE Trans. on Fuzzy Systems*, **17**, 824-838, (2009).
13. C. Cornelis, R. Jensen, G. Hurtado, and D. Slezak, "Attribute selection with fuzzy decision reducts," *Information Sciences*, **180**, 209-224, (2010).
14. L. A. Zadeh, "From computing with numbers to computing with words - from manipulation of measurements to manipulation of perceptions," *IEEE Trans. on Circuits and Systems*, **46**, 105-119 (1999).
15. S. K. Pal and D. Dutta Majumder, "Fuzzy mathematical approach to pattern recognition," *John Wiley (Halsted), New York*, (1986).
16. D. Sen and S. K. Pal, "Generalized rough sets, entropy and image ambiguity measures," *IEEE Trans. on Systems, Man, and Cybernetics - Part B*, **39**, 117-128, (2009).
17. A. M. Radzikowska and E. E. Kerre, "A comparative study of fuzzy rough sets," *Fuzzy Sets and Systems*, **126**, 137-156, (2002).
18. S. K. Pal and D. Dutta Majumdar, "Fuzzy sets and decision making approaches in vowel and speaker recognition," *IEEE Trans. on Systems, Man, and Cybernetics*, **7**, 625-629, (1977).
19. R. Jensen and Q. Shen, "Fuzzy-rough attribute reduction with application to web categorization," *Fuzzy Sets and Systems*, **141**, 469-485, (2004).
20. G. Salton and M. J. McGill, "An introduction to modern information retrieval," *McGrawHill, New York*, (1983).
21. Y. Q. Zhang, B. Jin, and Y. Tang, "Granular neural networks with evolutionary interval learning," *IEEE Trans. on Fuzzy Systems*, **16**, 309-319, (2008).

The work was done while Prof. S. K. Pal was a J.C. Bose Fellow of the Government of India.