Theoretical Computer Science 412 (2011) 5834–5853

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Fuzzy rough granular neural networks, fuzzy granules, and classification

Avatharam Ganivada[a,*], Soumitra Dutta[b], Sankar K. Pal[a]

[a] *Center for Soft Computing Research, Indian Statistical Institute, Kolkata 700108, India*
[b] *INSEAD, Blvd de Constance, Fontainebleau 77305, France*

ARTICLE INFO

Keywords:
Granular computing
Fuzzy granules
Fuzzy tolerance relation
Rule-based neural networks
Fuzzy pattern classification

ABSTRACT

We introduce a fuzzy rough granular neural network (FRGNN) model based on the multilayer perceptron using a back-propagation algorithm for the fuzzy classification of patterns. We provide the development strategy of the network mainly based upon the input vector, initial connection weights determined by fuzzy rough set theoretic concepts, and the target vector. While the input vector is described in terms of fuzzy granules, the target vector is defined in terms of fuzzy class membership values and zeros. Crude domain knowledge about the initial data is represented in the form of a decision table, which is divided into subtables corresponding to different classes. The data in each decision table is converted into granular form. The syntax of these decision tables automatically determines the appropriate number of hidden nodes, while the dependency factors from all the decision tables are used as initial weights. The dependency factor of each attribute and the average degree of the dependency factor of all the attributes with respect to decision classes are considered as initial connection weights between the nodes of the input layer and the hidden layer, and the hidden layer and the output layer, respectively. The effectiveness of the proposed FRGNN is demonstrated on several real-life data sets.

1. Introduction

Natural computation is a discipline that builds a bridge between computer science and natural science. It deals mainly with the methodologies and models that take their inspiration from nature (or are based on natural phenomena) for problem solving, use computers (or computational techniques) to synthesize natural phenomena, or employ natural materials (e.g., molecules) for computation. The constituent technologies for performing these tasks include cellular automata, artificial neural networks (ANNs), evolutionary algorithms, swarm intelligence, artificial immune systems, fractal geometry, artificial life, DNA computing, granular computing and perception-based computing. For example, artificial neural networks attempt to emulate the information representation / processing scheme and discrimination ability of biological neurons in the human brain, and they possess characteristics like adaptivity, robustness, ruggedness, speed, and optimality. Similarly, evolutionary algorithms are a biologically-inspired tool based on powerful paradigms from the natural world. They mimic some of the processes observed in natural evolution such as crossover, selection, and mutation, leading to a stepwise optimization of organisms. On the other hand, perception-based computing provides the capability to compute and reason with perception-based information as humans do to perform a wide variety of physical and mental tasks without any measurement and computation. Reflecting the finite ability of the sensory organs and (finally the brain) to resolve details, perceptions are inherently fuzzy-granular (f-granular) [1]. That is, the boundaries of perceived classes are unsharp and the values of the attributes they can take are granulated (a clump of indistinguishable points/objects) [2,3].

* Corresponding author.
E-mail address: avatharg@yahoo.co.in (A. Ganivada).

0304-3975/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2011.05.038

Granulation is also a computing paradigm, among others such as self-reproduction, self-organization, functioning of the brain, Darwinian evolution, group behavior, cell membranes, and morphogenesis, that are abstracted from natural phenomena. A good survey on natural computing explaining its different facets is provided in [4]. Granulation is inherent in human thinking and reasoning processes. Granular computing (GrC) provides an information processing framework, where computation and operations are performed on information granules, and it is based on the realization that precision is sometimes expensive and not very meaningful in modeling and controlling complex systems. 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.

The present article describes a granular neural network within the natural computing paradigm, where the structure of granulation is defined on fuzzy rough sets. In this consortium, artificial neural networks, fuzzy logic, and rough sets work synergistically, and use the principle of granular computing. The system exploits the tolerance for imprecision, uncertainty, approximate reasoning, and partial truth under the soft computing framework, and is capable of achieving tractability, robustness, and close resemblance with human-like (natural) decision making for pattern recognition in ambiguous situations [5].

Several authors have described different granular neural network architectures [31] in the past few years. Zhang et al. described granular neural networks in [32], incorporating fuzzy input data to neural networks. Vasilakos and Stathakis explained land-use classification from remote sensing images by fuzzy neural networks [33]. Granular neural networks using fuzzy sets as their formalism and an evolutionary training algorithm are reported in [34]. A good introduction to neuro-fuzzy inference systems may be found in [35]. One of the earlier neuro-fuzzy systems for classification, named a fuzzy neural network, was developed by Pal and Mitra [10]. As a part of determining the initial weights, Dick and Kandel built a novel neuro-fuzzy system architecture called a linguistic neural network based on information granules to simplify the knowledge representation in the neural networks [36]. Banerjee et al. described a knowledge-based network in [13], where domain knowledge is extracted from data in the form of decision rules using rough set theoretic techniques. Szczuka developed a rule-based rough neural network in [37], where rough sets are used mainly for generating the initial weights of the network.

Integration of fuzzy sets [6,7] and rough sets [14–17] under fuzzy rough computing or rough fuzzy computing has recently drawn the attention of researchers. Many relationships have been established to extend and integrate the underlying concepts of these two methodologies judiciously to deal with additional aspects of data imperfection, especially in the context of granular computing. The main purpose of such hybridization [19] is to provide a high degree of flexibility [20], robust solutions and advanced tools for data analysis [21], and a framework for efficient uncertainty handling [22]. This rough fuzzy paradigm may also be considered as a tool for modeling the aforesaid f-granular characteristics of perception-based computing. In rough set theory, one starts with crisp equivalence classes, in the same way that fuzzy equivalence classes are central to the fuzzy rough sets approach. Each equivalence class may be used as a granule. The concept of crisp equivalence classes can be extended to fuzzy equivalence classes by the inclusion of a fuzzy tolerance relation on the universe, which determines the extent that two elements are similar in a relation. Fuzzy rough sets [11], based on a fuzzy tolerance relation, provide a means by which discrete or real-valued noisy data can be effectively reduced without any additional information about the data (such as thresholds on a particular domain of universe) for its analysis. The granulation structure produced by an equivalence class provides a partition of the universe. The intension of it 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 notion of positive degree of each object, and this is used to define the dependency factor of each conditional attribute, all of which are then used to extract the domain knowledge about the data.

The granular network proposed in this paper integrates fuzzy rough sets with a fuzzy neural network that uses the dependency factors of all conditional attributes in the form of initial weights of the network (or network parameters), and neurons with a logistic activation function. The proposed system, called a fuzzy rough granular neural network (FRGNN), stores the domain knowledge about the data as a pattern of connection weights between simple processing units. It uses the theory of granulation structures in fuzzy rough sets [11,12], based on a fuzzy similarity relation, defining the dependency factors of attributes from the decision table. We considered three layers in the FRGNN. The appropriate number of hidden nodes in the network is determined by the number of classes, unlike other methods [13,36]. The number of nodes of the input layer is determined by the $3n$-dimensional linguistic features (in terms of fuzzy properties *low*, *medium*, and *high*) of the pattern. Clearly, if the input pattern contains n-dimensional features, the input layer is determined by $3n$ neurons.

The components of the input vector consist of membership values to the overlapping partitions of the linguistic properties *low*, *medium* or *high*, corresponding to each input feature. In this way an n-dimensional feature space is characterized by a $3n$-dimensional linguistic feature space. Each linguistic term is associated with a granule. The $3n$-dimensional linguistic feature space gives us conditional attributes which are presented in the decision table along with the decision class attributes. The decision table is divided into c decision tables corresponding to c classes. From the c decision tables, the dependency factor of each conditional attribute and the average value of all the dependency factors of the conditional attributes with respect to the decision classes are determined as the initial weights between the input layer and the hidden layer, and the hidden layer and the output layer, respectively, of the network. While the input vector that is presented to the network is in terms of a $3n$-dimensional linguistic vector, the output vector is in terms of a class membership value and zeros. This provides a scope for enhancing the robustness of the network in tackling ambiguous patterns and overlapping classes. Superiority of the model is demonstrated on various real-life datasets.

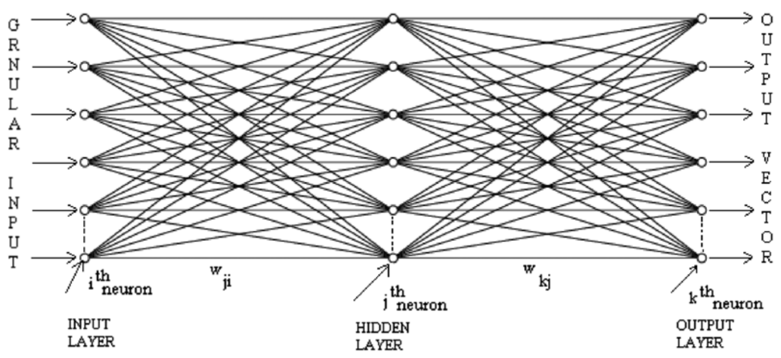


Fig. 1. Architecture of an FRGNN with a granular input layer, and hidden and output layers with the same number of nodes.

The paper is organized as follows: A brief description of the FRGNN architecture with the back-propagation algorithm is provided in Section 2. We describe the input vector and output vector of the FRGNN in Section 3. After recalling some preliminaries of granulations and approximations in rough sets and fuzzy sets, the proposed granulation structure in a fuzzy rough set theoretic framework is provided in Section 4. This includes both crisp and fuzzy cases for selecting the initial weights. Based on this granulation structure, we describe the knowledge encoding and network configuration of the FRGNN in Section 5. In Section 6, the FRGNN model is implemented on several real-life data sets for classification. This includes a comparison with some other versions of the fuzzy MLP (multilayer perceptron) integrated with rough sets. The paper is concluded in Section 7.

2. FRGNN architecture

In this section, we describe the architecture of an FRGNN based on a multilayer perceptron [8,9] using a back-propagation algorithm. The neurons are arranged in a layered feed-forward neural network shown in Fig. 1. Each neuron in the hidden layer is fully connected to the neurons in the next layer and the previous layer. The input layer is composed of non-computational units, each of which receives a single input, and distributes it to all the neurons in the next layer. A granular input vector is supplied as the input to the neurons in the input layer, and the output of the input layer is supplied as activation values to the neurons in the hidden layer. Similarly, the output of the hidden layer is provided as activation values to the output layer. During training, the node in the output layer corresponding to a class is clamped to the membership value of a pattern corresponding to that class, while the other nodes are clamped to zeros. The dependency factor of each conditional attribute and the average value of all the dependency factors of conditional attributes, with respect to the decision classes from all the decision tables, are encoded into the network as initial weights between the nodes of the input layer and the hidden layer, and the hidden layer and the output layer, respectively.

The back-propagation algorithm used in the FRGNN is described as follows.

Input

D, a data set consisting of the training patterns in granular form and their associated target vectors in terms of membership values and zeros.

η, the learning rate.

α, a momentum term.

$b = b_j$, the bias b is kept constant at each node j in the hidden and output layers network, creating a granular feed-forward network.

Output

A trained neural network.

Method

```
1. Initial weights are determined by fuzzy rough sets based on a fuzzy similarity relation, among the nodes (units)
   of all layers in the network;
2. While the terminating condition is not satisfied{
3. for each training pattern
      Propagate the inputs forward:
4. for each unit j of the input layer {
5. $x_j = I_j$; } here, the output of an input unit is its actual input value
6. for each unit j of the hidden or output layer, compute the net input of each unit j with respect to the previous
   layer, i {
7. $I_j = \sum_i w_{ji} x_i + b$; }
8. Apply logistic activation function to compute the output of each unit j {
9. $\phi(x_j) = \frac{1}{1+e^{-I_j}}$; }
```

```
    Back propagation:
10. for each unit $j$ in the output layer, compute the error {
11. $Error_j = \phi(x_j)(1-\phi(x_j))(T_j-\phi(x_j))$, where $T_j$ denotes the target value for each unit $j$ in the output layer
12. for each unit $j$ in the hidden layer, compute the error with respect to the next higher layer (unit $k$ in the output layer)
13. $\gamma_j = \phi(x_j)(1-\phi(x_j))\sum_k Error_k w_{jk}$; }
14. for each weight $w_{ij}$ in the network {
15. $\Delta w_{ij} = (\eta)x_i\gamma_j$;
16. $\Delta w_{ij}(t) = (\eta)x_i\gamma_j + (\alpha)\Delta w_{ij}(t-1)$; }
17. for each constant value of bias $b$ in the network {
        $\Delta b = (\eta)\gamma_j$;
        $b(t) += \Delta b(t-1)$;} }
```

Here, the momentum term α is used to escape local minima in the weight space. The value of the bias b is kept constant at each node of the hidden and output layers of the network, and t denotes an epoch, i.e., the number of times the training data will be presented to the network for updating the weight parameters w_{ij} and bias b. The resulting trained network is used for classifying the test patterns.

3. Input vector 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 include parking a car, driving in heavy traffic, and understanding speech. For performing such tasks, one needs perceptions of size, distance, weight, speed, time, direction, smell, color, shape, force, etc. But a fundamental difference between such measurements on the one hand and perception on the other is that the measurements are crisp numbers, whereas perceptions are fuzzy numbers or, more generally, fuzzy granules [23].

A formal definition of a fuzzy granule is a group of objects defined by the generalized constraint form 'X is r R' where 'R' is a constrained relation, 'r' is a random set constraint, which is a combination of probabilistic and possibilistic constraints, and 'X' is a fuzzy set valued random variable which takes the values *low*, *medium*, and *high*. Using fuzzy set theoretic techniques [7,24], a pattern point x, belonging to the universe U, may be assigned a grade of membership with a membership function $\mu_A(x)$ to a fuzzy set A. This is defined as

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

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

$$\pi(x,c,\lambda) = \begin{cases} 2\left(1-\frac{\|x-c\|_2}{\lambda}\right)^2, & \text{for } \frac{\lambda}{2} \le \|x-c\|_2 \le \lambda, \\ 1-2\left(\frac{\|x-c\|_2}{\lambda}\right)^2, & \text{for } 0 \le \|x-c\|_2 \le \frac{\lambda}{2}, \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

where $\lambda > 0$ is a scaling factor (radius) of the π function with c as a central point, and $\|\cdot\|_2$ denotes the Euclidian norm. The π membership function has been explained in the article by Pal and Mitra [10].

3.1. Incorporation of granular concept

An n-dimensional pattern is represented as a $3n$-dimensional linguistic vector [7]

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

where μ indicates the value of π membership function corresponding to a fuzzy granule *low*, *medium* or *high* along each feature axis. The fuzzy granule is quantified as

$$low \equiv \left\{\frac{0.689}{L}, \frac{0.980}{M}, \frac{0.998}{H}\right\},$$
$$medium \equiv \left\{\frac{0.054}{L}, \frac{0.991}{M}, \frac{0.217}{H}\right\}, \text{ and}$$
$$high \equiv \left\{\frac{0.979}{L}, \frac{0.919}{M}, \frac{0.199}{H}\right\}.$$

When the input is numerical, we use the π membership function of (2) with appropriate values of the center C and scaling factor λ. Their selection is explained in Section 3.2.

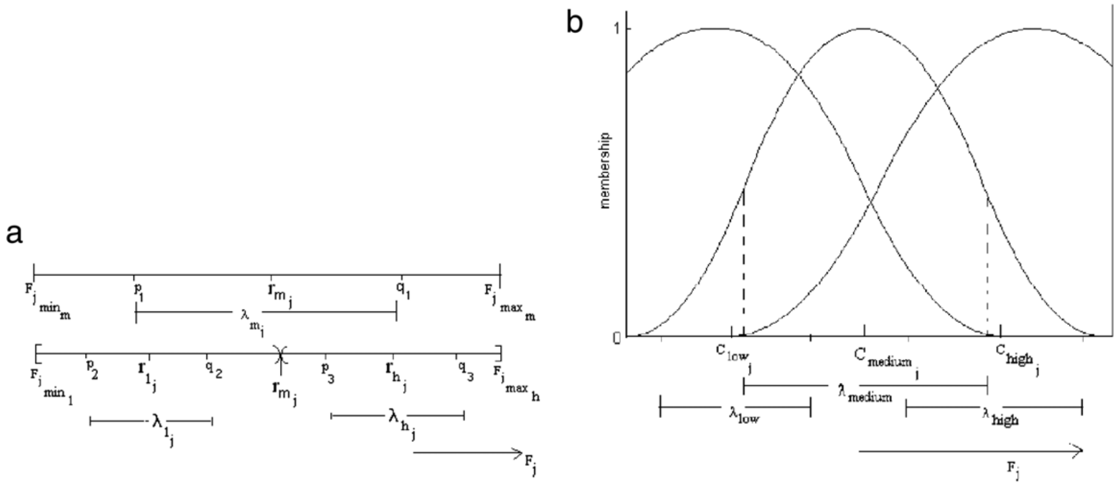


Fig. 2. Linguistic forms: *low*, *medium*, and *high*. (a) Parameters, (b) overlapping linguistic π sets.

3.2. Choice of parameters of π functions for numerical features

Let $\{F_{ij}\}$ for $i = 1, 2, \ldots, s, j = 1, 2, \ldots, n$, represent a set of s patterns with n features of a given data set, and $F_{j_{min_m}}$ and $F_{j_{max_m}}$ denote the minimum and maximum values along the jth feature considering all the s patterns. Initially, the average of feature values of all the s patterns along the jth feature F_j is considered as the center of the linguistic term *medium* along that feature and denoted by r_{m_j}, as shown in Fig. 2(a). Then, the average values (along the jth feature F_j) of the patterns having the label values in the ranges $[F_{j_{min_m}}, r_{m_j})$ and $(r_{m_j}, F_{j_{max_m}}]$ are defined as the means of the linguistic terms *low* and *high*, and denoted by r_{l_j} and r_{h_j}, respectively. Similarly, considering the patterns having label values in the ranges $[F_{j_{min_m}}, r_{m_j})$ and $(r_{m_j}, F_{j_{max_m}}]$ along jth axis, we define $F_{j_{min_l}} = F_{j_{min_m}}$, $F_{j_{max_l}} = r_{m_j}$, $F_{j_{min_h}} = r_{m_j}$, and $F_{j_{max_h}} = F_{j_{max_m}}$. Then the center C and corresponding scaling factor λ for the linguistic terms *low*, *medium*, and *high* along the jth feature F_j are as follows.

$$\begin{aligned}
&C_{medium_j} = r_{m_j},\\
&p_1 = C_{medium_j} - \frac{F_{j_{max_m}} - F_{j_{min_m}}}{2},\\
&q_1 = C_{medium_j} + \frac{F_{j_{max_m}} - F_{j_{min_m}}}{2},\\
&\lambda_{m_j} = q_1 - p_1,\\
&\lambda_{medium} = \frac{\sum_{j=1}^{n} \lambda_{m_j}}{n}.
\end{aligned} \tag{4}$$

$$\begin{aligned}
&C_{low_j} = r_{l_j},\\
&p_2 = C_{low_j} - \frac{F_{j_{max_l}} - F_{j_{min_l}}}{2},\\
&q_2 = C_{low_j} + \frac{F_{j_{max_l}} - F_{j_{min_l}}}{2},\\
&\lambda_{l_j} = q_2 - p_2,\\
&\lambda_{low} = \frac{\sum_{j=1}^{n} \lambda_{l_j}}{n}.
\end{aligned} \tag{5}$$

$$\begin{aligned}
&C_{high_j} = r_{h_j},\\
&p_3 = C_{high_j} - \frac{F_{j_{max_h}} - F_{j_{min_h}}}{2},\\
&q_3 = C_{highj} + \frac{F_{j_{max_h}} - F_{j_{min_h}}}{2},\\
&\lambda_{h_j} = q_3 - p_3,\\
&\lambda_{high} = \frac{\sum_{j=1}^{n} \lambda_{h_j}}{n}.
\end{aligned} \tag{6}$$

Clearly, the λ-value for a particular linguistic term along the axis would remain the same. Eqs. (4)–(6) automatically ensure that each input feature value along the jth axis for a pattern $\overrightarrow{F}_i$ is assigned three membership values corresponding to the three-dimensional (3D) granular space of Eq. (3) in such a way that at least one of the $\mu_{low(F_{i1})}(\overrightarrow{F}_i)$, $\mu_{medium(F_{i1})}(\overrightarrow{F}_i)$ and $\mu_{high(F_{i1})}(\overrightarrow{F}_i)$ is greater than 0.5. In other words, this allows a pattern $\overrightarrow{F}_i$ to have a strong membership to at least one of the linguistic properties *low*, *medium*, or *high*. This representation is shown diagrammatically in Fig. 2(b).

3.3. Class memberships as output vectors

The membership of the ith pattern to a class c_k is defined as

$$\mu_k(\overrightarrow{F}_i) = \frac{1}{1 + (\frac{Z_{ik}}{f_d})^{f_e}}, \tag{7}$$

where Z_{ik} is the weighted distance, and f_d and f_e are the denominational and exponential fuzzy generators [29] controlling the amount of fuzziness in the class membership. Obviously, the class membership lies in [0, 1]. We use

$$Z_{ik} = \sqrt{\sum_{j=1}^{n}\left[\sum_{p=1}^{3}\frac{1}{3}(\mu_p(F_{ij}) - \mu_p(o_{kj}))^2\right]}, \quad \text{for } k = 1, 2, \ldots, c, \tag{8}$$

where o_{kj} is the center of the jth feature vector from the kth class, and c is the number of classes. In the fuzziest case, we may use a fuzzy modifier, namely, a contrast internification (INT) operator [7,10], to enhance the contrast in membership values of each pattern within that class in order to decrease the ambiguity in taking a decision.

3.4. Applying the membership concept to the target vector

The target vector at the output layer is defined by a membership value and zeros Eq. (9). For example, if a training pattern belongs to the kth class, its desired output vector would have only one non-zero membership value corresponding to the kth node representing that class and zero value for the remaining nodes in the output layer. Therefore, for the ith training pattern $\overrightarrow{F_i}$ from the kth class, we define the desired output of the kth output node, considering the fuzziest case, as

$$d_k = \begin{cases} \mu_{INT}(\overrightarrow{F_i}), & \text{if the } i\text{th pattern is from } k\text{th class representing the } k\text{th output node,} \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

The network then back propagates the errors with respect to the desired membership values at the output stage.

4. Granulations and approximations in fuzzy rough sets

Here, we describe first some preliminaries on granulations based on rough sets and fuzzy sets. Then, the proposed structure of granulation on fuzzy rough sets is explained for two different cases.

4.1. Rough set theory: granulation by partitions

The granulation structure used in rough set theory is typically a partition of the universe. In rough set analysis [14–17], data represented as an information system is a pair $(U, \mathcal{A})$, where U is a non-empty finite set, called the universe, and $\mathcal{A}$ is a non-empty finite set of attributes. For every 'a' in $\mathcal{A}$, we define a mapping $a : U \to V_a$, where V_a is the value set of 'a' over U. An information system can be defined as an attribute value table, in which rows and columns are labeled by objects of the universe and attributes, respectively. For every $B \subseteq \mathcal{A}$, the B-indiscernibility relation R_B is defined as

$$R_B = \{(x, y) \in U^2 \quad | \quad \forall a \in B, \quad a(x) = a(y)\}, \tag{10}$$

where R_B is an equivalence relation. It partitions the universe U into disjoint subsets, and each partition may be viewed as a granule consisting of indistinguishable elements. It is also referred to as an equivalence granule $[\tilde{x}]_{R_B}$ in [18]. Given $A \subseteq U$ an arbitrary set, it may not be possible to describe U directly using the equivalence granules $[\tilde{x}]_{R_B}$. In this case, one may characterize A by a pair of lower and upper approximations

$$R_B \downarrow A = \{x \in U | [\tilde{x}]_{R_B} \subseteq A\}, \tag{11}$$

$$R_B \uparrow A = \{x \in U | [\tilde{x}]_{R_B} \cap A \neq \emptyset\}. \tag{12}$$

The pair $(R_B \downarrow A, R_B \uparrow A)$ is called a rough set. In information system, a decision system is characterized by $(U, \mathcal{A} \cup d)$, where d ($d \notin \mathcal{A}$) is called a decision attribute and its equivalence classes $[\tilde{x}]_{R_d}$ are called decision classes (decision granules). Given $B \subseteq \mathcal{A}$, the B-positive region can be defined as

$$POS_B = \bigcup_{x \in U} R_B \downarrow [\tilde{x}]_{R_d}. \tag{13}$$

The positive region contains all the objects of U that can be classified into the equivalence granules of R_B using the information in the attribute B. The degree of dependency of an attribute in B is then measured by

$$\gamma_B = \frac{|POS_B|}{|U|}, \tag{14}$$

where $|\cdot|$ denotes the cardinality of a set U.

4.2. Fuzzy sets: granulation by partitions

In the context of fuzzy rough set theory [20,21], fuzzy set theory [7,24] allows that an object belongs to a set, and a couple of objects belong to a relation, to a given degree. Recall that Eq. (1) is defined as a fuzzy set in U. A fuzzy relation R in U is a mapping $U \times U \to [0, 1]$, where the mapping is expressed by the membership function $R(x, y)$ of the 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 the fuzzy set R_y defined by

$$R_y(x) = R(x, y),$$

for all x in U.

In fuzzy rough set theory, a similarity between objects in U is modeled by a fuzzy tolerance relation R; i.e.,

$$R(x, x) = 1 \quad \text{(reflexive)},$$
$$R(x, y) = R(y, x) \quad \text{(symmetry), and}$$
$$T(R(x, y)R(y, z)) \le R(x, z) \quad (T\text{-transitivity}),$$

for all x, y, z in U. Given a t-norm (or a T-norm), R is then called a fuzzy T-equivalence relation or a fuzzy similarity relation (fuzzy tolerance relation). The fuzzy similarity relations are commonly considered to measure the approximate equality of objects, which is explained in Section 4.3. In general, for the fuzzy tolerance relation R, we call R_y a fuzzy T-equivalence class (fuzzy equivalence granule) of y. The fuzzy logical counterparts of the connectives [24] are used in the generalization of lower approximation (11) and upper approximation (12) in fuzzy rough set theory. We now recall some definitions. A t-norm (triangular norm) $T : [0, 1]^2 \to [0, 1]$ satisfies $T(1, x) = x$. We use T_M and T_L to represent t-norms, and these are defined as

$$T_M(x, y) = \min(x, y), \quad \text{and}$$
$$T_L(x, y) = \max(0, x + y - 1) \quad (\text{Lukasiewicz } t\text{-norm}),$$

for all $x, y \in [0, 1]$. On the other hand, a mapping $I : [0, 1] \times [0, 1] \to [0, 1]$ satisfies $I(0, 0) = 1, I(1, x) = x$ for all $x \in [0, 1]$, where I is an implicator. For all $x, y \in [0, 1]$, the implicators I_M and I_L are defined by

$$I_M(x, y) = \max(1 - x, y) \quad \text{(Kleene–Dienes implicator), and}$$
$$I_L(x, y) = \min(1, 1 - x + y) \quad \text{(Lukasiewicz implicator).}$$

4.3. Fuzzy rough sets: granulation by partitions

Research on fuzzifying lower and upper approximations in the spirit of Pawlak [14] emerged in late 1980s. The proposals had emerged due to Nakamura [27], Dubois and Prade [20], and Banerjee and Pal [22], who drew inspiration from an earlier publication by Cerro and Prade [26]. Several other researchers have reported their contributions on hybridization of fuzzy sets and rough sets [11,12,19,25,28]. In doing so, the following two principles were adopted for fuzzyfication of the formulae (11) and (12) for the lower and upper approximations of a set.

1. A 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) with membership degrees in [0, 1].
2. Usually, 'object indistinguishability' (for instance, with respect to their attribute values in an information system) is described by means of an equivalence relation R in U in Pawlak's rough approximation. Rather than doing so, the approximation equality of objects can be represented by a fuzzy similarity relation R in generalized approximation space. As a result, objects are categorized into different classes or granules with "soft" boundaries based on the similarity among them.

In fuzzy rough sets [12], an information system is a pair $(U, \mathcal{A})$, where $U = \{x_1, x_2, \ldots, x_s\}$ and $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ are finite non-empty sets of objects and conditional attributes, respectively. In our work, the values of conditional attributes can be quantitative (real valued). Let 'a' be a quantitative attribute in $\mathcal{A}$. We express the fuzzy similarity relation R_a between any two objects x and y in U with respect to the attribute 'a' as

$$R_a(x, y) = max\left(min\left(\frac{a(y) - a(x) + \sigma_a}{\sigma_a}, \frac{a(x) - a(y) + \sigma_a}{\sigma_a}\right), 0\right), \tag{15}$$

where σ_a denotes the standard deviation of the attribute 'a'. R_a is also called a fuzzy a-indiscernibility relation. It may be noted that the relation R_a does not necessarily satisfy the T-transitivity property. For any $B \subseteq \mathcal{A}$, the fuzzy B-indiscernibility relation R_B based on the fuzzy similarity relation R is induced by

$$R_B(x, y) = \underbrace{T\ (R_a(x, y))}_{a \in B}, \tag{16}$$

where T represents a t-norm. For each $y \in U$, the fuzzy tolerance class of R_B is defined by $R_{B_y}(x) = R_B(x, y)\ \forall\ x$ in U. The fuzzy tolerance classes of R_B can be used to approximate the fuzzy sets (called concepts) in U.

A decision system $(U, \mathcal{A} \cup \{d\})$ is a special kind of information system, in which d $(d \notin \mathcal{A})$ is called a decision attribute and it can be qualitative (discrete valued). Based on these values, the set U is partitioned into a number of non-overlapping decision concepts $R_d(x_k)$, $k = 1, 2, \ldots, c$, where each decision concept corresponds to a decision class. Each object $x_i \in U$ is classified by the decision classes. Each decision class may be represented by a crisp set or a fuzzy set. For a qualitative attribute 'a' in {d}, the decision classes are defined in the following two methods.

Method I (crisp case): crisp way of defining decision classes

$$R_a(x, y) = \begin{cases} 1, & \text{if a(x) = a(y),} \\ 0, & \text{otherwise,} \end{cases} \tag{17}$$

for all x, y in U. The crisp-valued decision class implies that objects in the universe U corresponding to the decision equivalence granule would take values only from the set {0, 1}. A fuzzy set $A \subseteq U$ can be approximated only by constructing the lower and upper approximations of A with respect to crisp decision classes.

In real-life problems, the data are generally ill defined, with overlapping class boundaries. Each pattern used in the fuzzy set $A \subseteq U$ may belong to more than one class. To model such data, we extend the concept of a crisp decision granule into a fuzzy decision granule by inclusion of the fuzzy concept to crisp decision granules. The fuzzy decision classes are defined as follows.

Method II (fuzzy case): fuzzy way of defining decision classes

Consider a c-class problem domain where we have c decision classes of a decision attribute in a decision system. Let the n-dimensional vectors O_{kj} and V_{kj}, j = 1,2, …, n, denote the mean and standard deviation of the data for the kth class in the given decision system. The weighted distance of a pattern $\overrightarrow{F_i}$ from the kth class is defined as

$$Z_{ik} = \sqrt{\sum_{j=1}^{n} \left[\frac{F_{ij} - O_{kj}}{V_{kj}}\right]^2}, \quad \text{for } k = 1, 2, \ldots, c, \tag{18}$$

where F_{ij} is the value of the jth component of the ith pattern. The parameter $\frac{1}{V_{kj}}$ acts as the weighting coefficient, such that the larger the value of V_{kj}, the less is the importance of the jth feature in characterizing the kth class. Note that when the value of a feature for all the patterns in a class is the same, the standard deviation of those patterns along that feature will be zero. In that case, we consider V_{kj} = 0.000001 (a very small value, instead of zero for the sake of computation) so that the weighting distance Z_{ik} becomes high and the membership values of the ith pattern to the kth class along that feature become low (Eq. (7)).

It may be noted that when a pattern $\overrightarrow{F}_i$ has different membership values corresponding to c decision classes then its decision attribute becomes quantitative, i.e., each pattern has varying membership value. The quantitative decision attribute can be made qualitative (as in Method I, crisp case) in two ways, namely, (i) by computing the average of the membership values over all the patterns in the kth class to its own class, and assigning it to each pattern $\overrightarrow{F}_i$ in its kth decision class, and (ii) by computing the average of the membership values over all the patterns in the kth class to the other classes, and assigning it to each pattern $\overrightarrow{F}_i$ in other decision classes (other than the kth class). So the average membership value (qualitative) of all the patterns in the kth class to its own class is defined as

$$D_{kk} = \frac{\sum_{i=1}^{m_k} \mu_k(\overrightarrow{F}_i)}{|m_k|}, \quad \text{if } k = r, \tag{19}$$

and the average membership values (qualitative) of all the patterns in the kth class to other decision classes are defined as

$$D_{kr} = \frac{\sum_{i=1}^{m_k} \mu_r(\overrightarrow{F}_i)}{|m_k|}, \quad \text{if } k \neq r, \tag{20}$$

where $|m_k|$ indicates the number of patterns in the kth class, and $k, r = 1, 2, \ldots, c$.

For a qualitative attribute 'a' $\in \{d\}$, the fuzzy decision classes are defined as

$$R_a(x, y) = \begin{cases} D_{kk}, & \text{if a(x) = a(y),} \\ D_{kr}, & \text{otherwise,} \end{cases} \tag{21}$$

for all x, y in U. Here, D_{kk} corresponds to an average membership value of all the patterns that belong to the same class ($k = r$), and D_{kr} corresponds to an average membership values of all the patterns from classes other than k ($k \neq r$). The way of implementing the aforesaid Methods I and II is explained through examples in Section 5.2.

For computing the lower and upper approximations of the fuzzy set $A \subseteq U$ with a fuzzy similarity relation R under fuzzy logic connectives (a t-norm T and an implicator I), we extend Eqs. (11) and (12) with the definitions given in [12] as

$$(R \downarrow A)(y) = \inf_{x \in U} I(R(x, y), A(x)), \tag{22}$$

$$(R \uparrow A)(y) = \sup_{x \in U} T(R(x, y), A(x)), \tag{23}$$

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

$$POS_B(y) = \left(\bigcup_{x \in U} R_B \downarrow R_d x \right)(y), \tag{24}$$

for all $x, y \in U$. The positive region of a fuzzy set characterizes a region with maximum membership value. Considering y belonging to a fuzzy set which is a subset of U, the fuzzy positive region Eq. (24) can be simplified as

$$POS_B(y) = (R_B \downarrow R_d x)\,(y). \tag{25}$$

The method of simplification is explained in [12]. The degree of dependency of d on the set of attributes $B \subseteq \mathcal{A}$ is defined as

$$\gamma_B = \frac{\sum_{x \in U} POS_B(x)}{|U|}, \tag{26}$$

where $|\cdot|$ denotes the cardinality of a set U, and γ is $0 \leq \gamma \leq 1$. The fuzzy set $A \subseteq U$ is said to be completely dependent on B if $\gamma = 1$.

It may be noted that the aforesaid procedure of determining the dependency factor of each conditional attribute with respect to the decision granule in either the crisp case or the fuzzy case enables the proposed FRGNN to encode its initial weight parameters from the training samples. The initial weights in the fuzzy case (Method II) are seen to provide better performance in handling the overlapping class boundaries than those of the crisp case (Method I). These are explained in the following sections.

5. Configuration of the FRGNN using fuzzy rough sets

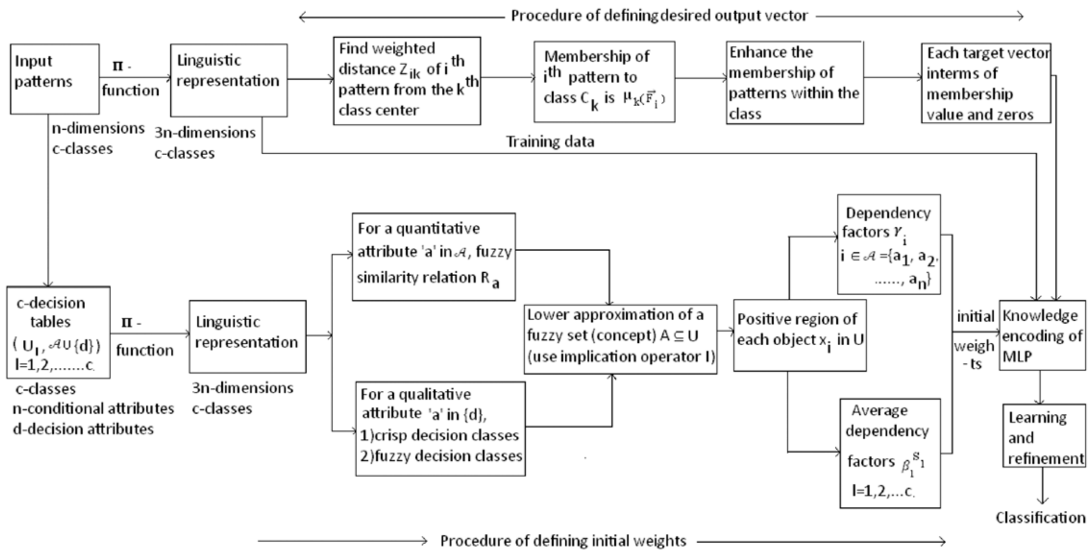
In this section, we first show how the decision tables can be used to explain the concept of granulation by partitions and fuzzy rough set approximations based on a fuzzy similarity relation. Based on this principle, the initial weights of the FRGNN are then determined; thereby providing a knowledge-based network. Such a network is found to be more efficient than other similar types of granular neural network [13,36] and the conventional MLP [9]. During training, these networks search for a set of connection weights that corresponds to some local minima. Note that there may be a large number of such minima corresponding to various good solutions. Therefore, if we can initially set weights of the network so as to correspond nearby one such solution, the searching space may be reduced and learning thereby becomes faster. Further, the architecture of the network can be made simpler by fixing the number of nodes in the hidden layer based on the class information. These are the characteristics that the proposed FRGNN is capable of achieving. The knowledge encoding procedure is described in the next section using the concepts of fuzzy rough sets and a fuzzy similarity relation (Fig. 3).

5.1. Knowledge encoding procedure

Let $S = (U, \mathcal{A} \cup \{d\})$ be a decision table, with $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ its set of conditional attributes, and with decision attributes $\{d\}$, where $U = \{x_1, x_2, \ldots, x_s\}$, its set of objects, form c classes and objects having labeled values corresponding to each n-dimensional conditional attribute. We split the decision table $S = (U, \mathcal{A} \cup \{d\})$ into c decision tables $S_l = (U_l, \mathcal{A} \cup \{d\})$, $l = 1, 2, \ldots, c$, corresponding to c classes, and the objects are added to each decision table from all the c classes succession. Moreover, S_l satisfies the following conditions:

$$U_l \neq \emptyset, \quad \bigcup_{l}^{c} U_l = U, \quad \bigcap_{l=1}^{c} U_l = \emptyset. \tag{27}$$

The size of each S_l, $l = 1, 2, \ldots, c$, is dependent on the available number of objects from all the classes. If all the classes have an equal number of patterns, then the number of objects that will be added to each S_l will be same; otherwise, it will be different.



Procedure of defining desired output vector
Input patterns
Π - function
Linguistic representation
Find weighted distance Z_ik of i th pattern from the k th class center
Membership of i th pattern to class C_k is μ_k(F_i)
Enhance the membership of patterns within the class
Each target vector interms of membership value and zeros
n-dimensions c-classes
3n-dimensions c-classes
Training data
c-decision tables (U_l, 𝒜∪{d}) l=1,2,.......c
Π - function
Linguistic representation
c-classes n-conditional attributes d-decision attributes
3n-dimensions c-classes
For a quantitative attribute 'a' in 𝒜, fuzzy similarity relation R_a
For a qualitative attribute 'a' in {d}, 1)crisp decision classes 2)fuzzy decision classes
Lower approximation of a fuzzy set (concept) A ⊆ U (use implication operator I)
Positive region of each object x_i in U
Dependency factors γ_i i ∈ 𝒜 ={a_1, a_2, ..., a_n}
Average dependency factors β_l^{S_l} l=1,2,...c.
initial weigh-ts
Knowledge encoding of MLP
Learning and refinement
Classification
Procedure of defining initial weights

Fig. 3. Block diagram of knowledge encoding.

Let us consider the case of feature F_j for S_l, $l = 1, 2, \dots, c$. The ith representative pattern $\overrightarrow{F}_i$ is mapped to a point in the 3D 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 Eq. (3). In this manner, an n-dimensional attribute valued decision table can be transformed into a 3n-dimensional attribute valued decision table. We apply the following procedure for each decision table S_l, $l = 1, 2, \dots, c$.

Step 1:
Obtain additional granulation structures using the fuzzy similarity relation, defined in Eq. (15), on each conditional attribute by generating a fuzzy similarity matrix.

Step 2:
Use Step 1 to compute lower approximations, defined in Eq. (22), of each concept for each conditional attribute with respect to the decision classes, defined in Eq. (17) or (21), based on the fuzzy logic connective (Lukasiewicz implicator).

Step 3:
Calculate the fuzzy positive region, defined in Eq. (25), of each object for each conditional attribute.

Step 4:
Calculate the degree of dependency, defined in Eq. (26), of each conditional attribute, and then the resulting values are determined as initial weights between the nodes of the input layer and hidden layer.

Step 5:
Calculate the degree of dependency, defined in Eq. (26), of each conditional attribute corresponding to the objects within the class with respect to each decision class and average value of all those degree of dependencies, and then the resulting values are determined as initial weights between the nodes of the hidden layer and the output layer.

Let us now design the initial structure of the three-layered FRGNN. The number of input layer nodes consists of the 3n-dimensional attribute values and the output layer is represented by c classes. The hidden layer nodes are modeled with the class information. Next, we explain the procedure of encoding the initial weights of the network. Let the dependency degrees for conditional attributes of a decision table S_1, for instance, be γ_i, $i \in \mathcal{A} = \{a_1, a_2, \dots, a_n\}$. The weight w_{li} between an input node i and a hidden node l is defined as follows, for instance, when $l = 1$:

$$\gamma_i^{S_1} = \frac{\sum_{x \in U} POS_i(x)}{|U|}. \tag{28}$$

Let β_l denote the connection weight between a hidden node and an output node. The weight w_{kl} between the hidden node l and the output node k, for instance, when $k = 1$, is defined as follows. Given c decision classes for a decision table $\{d_1, d_2, d_3, \dots, d_c\}$,

$$\beta_l^{S_1} = \frac{\sum_{i=1}^{n} \gamma_i^l}{|n|}, \tag{29}$$

Table 1
Dataset.

U	a	b	c	d
1	−0.4	−0.3	−0.5	1
2	−0.4	0.2	−0.1	2
3	−0.3	−0.4	0.3	1
4	0.3	−0.3	0	2
5	0.2	−0.3	0	2
6	0.2	0	0	1

Table 2
Decision table.

U	L_1	M_1	H_1	L_2	M_2	H_2	L_3	M_3	H_3	d
1	0.875	0.395062	0	0.929687	0.924383	0	0.125	0.347222	0	1
2	0.875	0.395062	0	0	0.260802	0.125	0	0.986111	0.382812	2
3	0.5	0.697531	0	0.382812	0.739198	0	0.125	0.875	0	1
4	0	0.302469	0.5	0.929687	0.924383	0	0	0.875	0.929688	2
5	0	0.604938	0.875	0.929687	0.924383	0	0	0.875	0.929688	2
6	0	0.604938	0.875	0	0.813272	0.125	0	0.875	0.929688	1

where γ_i^l is defined as

$$\gamma_i^l = \frac{\sum_{x \in U_{d_l}} POS_i(x)}{|U_{d_l}|}, \quad l = 1, 2, \ldots, c. \tag{30}$$

A similar procedure is applied to the rest of the decision tables S_l. Then the degrees of dependency of conditional attributes, $\gamma_i^{S_l}$, for $l = 1, 2, \ldots, c$, are used as the initial weights between the nodes of the input layer and the hidden layer. The initial weight between nodes of the hidden and output layers is set to $\beta_l^{S_l}$. The connection weights, so encoded, are refined by training the network on a set of patterns supplied as input.

5.2. Example

We apply the operations of fuzzy rough sets based on the similarity relation on an example dataset as given in Table 1 to determine the initial weights of the FRGNN.

Crisp case

Each conditional attribute in Table 1 can be transformed into a 3D granular space. Then the resulting decision table is shown in Table 2.

Apply Step 1 of Section 5.1 to Table 2. The resulting similarity matrices of the conditional attributes L_1 and M_1 are as follows:

$$R_{L_1}(x, y) = \begin{pmatrix} 1 & 1 & 0.555 & 0 & 0 & 0 \\ 1 & 1 & 0.555 & 0 & 0 & 0 \\ 0.555 & 0.555 & 1 & 0.407 & 0.407 & 0.407 \\ 0 & 0 & 0.407 & 1 & 1 & 1 \\ 0 & 0 & 0.407 & 1 & 1 & 1 \\ 0 & 0 & 0.407 & 1 & 1 & 1 \end{pmatrix}$$

$$R_{M_1}(x, y) = \begin{pmatrix} 1 & 1 & 0.691 & 0.905 & 0.785 & 0.785 \\ 1 & 1 & 0.691 & 0.905 & 0.785 & 0.785 \\ 0.691 & 0.691 & 1 & 0.596 & 0.905 & 0.905 \\ 0.905 & 0.905 & 0.596 & 1 & 0.691 & 0.691 \\ 0.785 & 0.785 & 0.905 & 0.691 & 1 & 1 \\ 0.785 & 0.785 & 0.905 & 0.691 & 1 & 1 \end{pmatrix}.$$

Similarly, the similarity matrices for the remaining attributes $R_{H_1}(x, y)$, $R_{L_2}(x, y)$, $R_{M_2}(x, y)$, $R_{H_2}(x, y)$, $R_{L_3}(x, y)$, $R_{M_3}(x, y)$, $R_{H_3}(x, y)$ can be determined. We then calculate the lower approximations (from Step II) of the concepts $x_0 = \{1, 3, 6\}$ and $x_1 = \{2, 4, 5\}$ for every conditional attribute with respect to the decision classes $R_d(x, y)$, where y belongs to x_0 and x_1, and x belongs to U.

Apply Step 2 from Section 5.1 to the concept x_0.

$(R_{L_1} \downarrow R_d x)(y) = \inf_{x \in \mathbf{U}} I\{R_{L_1}(x, y), R_d(x, y)\}$.

For object 1, this is
$(R_{L_1} \downarrow R_d x)(1) = \inf_{x\in\mathbf{U}} I\{R_{L_1}(x, 1), R_d(x, 1)\},$
$= \min\{I(1, 1), I(1, 0), I(0.555, 1), I(0, 0), I(0, 0), I(0, 1)\},$
$= 0.0.$
For object 3, this is
$(R_{L_1} \downarrow R_d x)(3) = \min\{I(0.555, 1), I(0.555, 0), I(1, 1), I(0.407, 0), I(0.407, 0), I(0.407, 1)\},$
$= 0.444.$
For object 6, $(R_{L_1} \downarrow R_d x)(6) = 0.0.$
Similarly, apply Step 2 from Section 5.1 to the concept x_1.

$$(R_{L_1} \downarrow R_d x)(2) = 0.0, \qquad (R_{L_1} \downarrow R_d x)(4) = 0.0, \qquad (R_{L_1} \downarrow R_d x)(5) = 0.0.$$

For any qualitative attribute 'a' ∈ {d}, the lower approximations are equivalent to positive degrees of objects in both the concepts x_0 and x_1. The resulting dependency degree of a conditional attribute L_1 is $\gamma_{\{L_1\}}$ = 0.074.

Calculating the dependency degrees for the remaining conditional attributes from Table 2, we have

$$\begin{aligned} &\gamma_{\{M_1\}} = 0.031, && \gamma_{\{H_1\}} = 0.074,\\ &\gamma_{\{L_2\}} = 0.077, && \gamma_{\{M_2\}} = 0.506,\\ &\gamma_{\{H_2\}} = 0.000, && \gamma_{\{L_3\}} = 0.042,\\ &\gamma_{\{M_3\}} = 0.111, \quad \textit{and} && \gamma_{\{H_3\}} = 0.233. \end{aligned}$$

These resulting dependency degrees are considered as the initial connection weights from nine input layer nodes to one hidden layer node of the FRGNN corresponding to class 1. Next, we define the connection weights from the hidden layer nodes to one output layer node as follows.

The positive degrees of objects in the concept $x_0 = \{1, 3, 6\}$ with respect to the decision class $R_d(x_0)$ for the attribute L_1 are

$$POS_{L_1}(1) = 0.0, \qquad POS_{L_1}(3) = 0.444, \qquad POS_{L_1}(6) = 0.0.$$

Hence, the resulting degree of dependency is

$$\gamma_{\{L_1\}}(x_0) = \frac{0.0 + 0.444 + 0.0}{3} = 0.148.$$

The positive degrees of objects with respect to the decision class $R_d(x_1)$ for the attribute L_1 are

$$POS_{L_1}(2) = 0.0, \qquad POS_{L_1}(4) = 0, \qquad POS_{L_1}(5) = 0.0.$$

Hence, the resulting degree of dependency is

$$\gamma_{\{L_1\}}(x_1) = \frac{0.0 + 0.0 + 0.0}{3} = 0.0.$$

The dependency degrees for the rest of the attributes with respect to each decision class are as follows.

$$\begin{aligned} &\gamma_{\{M_1\}}(x_0) = 0.031, && \gamma_{\{M_1\}}(x_1) = 0.031,\\ &\gamma_{\{H_1\}}(x_0) = 0.0, && \gamma_{\{H_1\}}(x_1) = 0.148,\\ &\gamma_{\{L_2\}}(x_0) = 0.155, && \gamma_{\{L_2\}}(x_1) = 0.0,\\ &\gamma_{\{M_2\}}(x_0) = 0.104, && \gamma_{\{M_2\}}(x_1) = 0.168,\\ &\gamma_{\{H_2\}}(x_0) = 0.0, && \gamma_{\{H_2\}}(x_1) = 0.0,\\ &\gamma_{\{L_3\}}(x_0) = 0.183, && \gamma_{\{L_3\}}(x_1) = 0.0,\\ &\gamma_{\{M_3\}}(x_0) = 0.183, && \gamma_{\{M_3\}}(x_1) = 0.038,\\ &\gamma_{\{H_3\}}(x_0) = 0.310, \quad \textit{and} && \gamma_{\{H_3\}}(x_1) = 0.155. \end{aligned}$$

The average dependency degrees of all the conditional attributes with respect to the decision classes are characterized by $\gamma(x_0) = 0.113$ and $\gamma(x_1) = 0.060$. These values are used to represent the initial connection weights of two hidden layer nodes to one output layer node of the FRGNN that corresponds to class 1.

Fuzzy case

Each conditional attribute in Table 1 can be transformed into the $3n$-dimensional granular space of Eq. (3). Then the resulting values are shown in Table 3. The average membership values of Eqs. (19) and (20) are presented in Table 3 under the decision attribute columns D_{kk} and D_{kr}.

We then calculate the lower approximations (from Step II) of the concepts $x_0 = \{1, 3, 6\}$ and $x_1 = \{2, 4, 5\}$ for every conditional attribute with respect to the fuzzy decision classes $R_d(x, y)$, where y belongs to x_0 and x_1, and x belongs to U.

Table 3
Decision table.

U	L_1	M_1	H_1	L_2	M_2	H_2	L_3	M_3	H_3	d	D_{kk}	D_{kr}
1	0.875	0.395	0	0.929	0.924	0	0.125	0.347	0	1	0.257	0.180
3	0.5	0.697	0	0.382	0.739	0	0.125	0.875	0	1	0.257	0.180
6	0	0.604	0.875	0	0.813	0.125	0	0.875	0.929	1	0.257	0.180
2	0.875	0.395	0	0	0.260	0.125	0	0.986	0.382	2	0.276	0.084
4	0	0.302	0.5	0.929	0.924	0	0	0.875	0.929	2	0.276	0.084
5	0	0.604	0.875	0.929	0.924	0	0	0.875	0.929	2	0.276	0.084

For the concept $x_0 = \{1, 3, 6\}$,
$(R_{L_1} \downarrow R_d x)(y) = \inf_{x\in\mathbf{U}} I\{R_{L_1}(x, y), R_d(x, y)\}$.
For object 1, this is

$$\begin{aligned}(R_{L_1} \downarrow R_d x)(1) &= \inf_{x\in\mathbf{U}} I\{R_{L_1}(x, 1), R_d(x, 1)\},\\ &= \min\{I(1, 0.257), I(1, 0.180), I(0.555, 0.257),\\ &\qquad I(0, 0.180), I(0, 0.180), I(0, 0.257)\},\\ &= 0.180.\end{aligned}$$

For object 3, this is

$$\begin{aligned}(R_{L_1} \downarrow R_d x)(3) &= \inf_{x\in\mathbf{U}} I\{R_{L_1}(x, 3), R_d(x, 3)\},\\ &= \min\{I(0.555, 0.257), I(0.555, 0.180), I(1, 0.257),\\ &\qquad I(0.407, 0.180), I(0.407, 0.180), I(0.407, 0.257)\},\\ &= 0.257.\\ (R_{L_1} \downarrow R_d x)(6) &= \inf_{x\in\mathbf{U}} I\{R_{L_1}(x, 6), R_d(x, 6)\},\\ &= \min\{I(0, 0.257), I(0, 0.180), I(0.407, 0.257),\\ &\qquad I(1, 0.180), I(1, 0.180), I(1, 0.257)\},\\ &= 0.180.\end{aligned}$$

For the concept $x_1 = \{2, 4, 5\}$,

$$(R_{L_1} \downarrow R_d x)(2) = 0.084, \qquad (R_{L_1} \downarrow R_d x)(4) = 0.084, \qquad (R_{L_1} \downarrow R_d x)(5) = 0.084.$$

The resulting degree of dependency of the conditional attribute L_1 is $\gamma_{\{L_1\}}$ = 0.145. Calculating the dependency degrees for the remaining features from Table 3 results in

$$\begin{aligned}&\gamma_{\{M_1\}} = 0.160, &&\gamma_{\{H_1\}} = 0.164,\\ &\gamma_{\{L_2\}} = 0.145, &&\gamma_{\{M_2\}} = 0.190,\\ &\gamma_{\{H_2\}} = 0.132, &&\gamma_{\{L_3\}} = 0.158,\\ &\gamma_{\{M_3\}} = 0.164, \quad \textit{and} &&\gamma_{\{H_3\}} = 0.190.\end{aligned}$$

These resulting dependency degrees are used as the initial connection weights between the input layer nodes to one node in the hidden layer of the FRGNN that corresponds to class 1. Similarly, one can determine from Table 3 the initial connection weights from the hidden layer nodes to one output layer node of the FRGNN corresponding to class 1. The resulting average degrees of dependency of all the conditional attributes with respect to the decision classes are characterized by $\gamma(x_0)$ = 0.209 and $\gamma(x_1)$ = 0.113.

So far, we have discussed the procedure for determining the connection weights corresponding to one decision table. If a similar procedure is applied to the other decision table in both crisp and fuzzy cases then a fully connected FRGNN with initial weight parameters would be generated.

6. Implementation and experimental results

The proposed FRGNN which incorporates fuzzy granular concepts at various levels has been implemented in C on several real-life data sets. We describe their characteristics in Table 4.

The speech data "vowel" deals with 871 Indian Telugu vowel sounds [29]. These were uttered in a consonant–vowel–consonant context by three male speakers in the age group 30–35 years. The data set has three features, F_1, F_2, and F_3, corresponding to the first, second, and third vowel format frequencies obtained through spectrum analysis of speech data. Fig. 4(a) shows a two-dimensional (2D) projection of the 3D feature space of the six vowel classes in the F_1–F_2 plane. All the other data sets (such as pima Indian, sonar, and thyroid) are taken from the UCI Machine Learning Repository [30]. Fig. 4(b) shows a 3D projection of the 60D feature space of the two classes of sonar data in the F_1–F_2–F_3 space. Fig. 5(a) shows a 3D projection of the 18D feature space of the four classes of lymphography data in the F_1–F_2–F_3 space. Fig. 5(b) shows a 3D projection of the 5D feature space of the three classes of thyroid data in the F_1–F_2–F_3 space.

During learning, we have used an n-fold cross-validation design with stratified sampling. The training is done on $(n-1)$ folds of the data selected randomly from each of the classes. The remaining one-fold data is considered as the test set. This is repeated n times, and the overall performance of the FRGNN is computed taking an average over n sets of folds. In our

Table 4
Data set characteristics.

Dataset name	Number of patterns	Features	Classes	Origin
Telugu vowel	871	3	6	[29]
Pima Indian	768	8	2	UCI
Sonar	208	60	2	UCI
Lymphography	148	18	4	UCI
Glass	214	9	6	UCI
Ionosphere	351	33	2	UCI
Thyroid	215	5	3	UCI
Wine	178	12	3	UCI

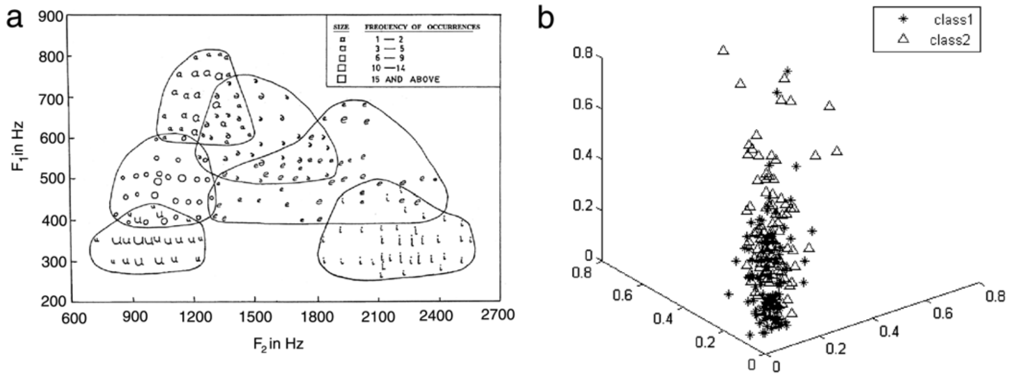


Fig. 4. (a) Vowel data in the F_1–F_2 plane. (b) Sonar data in the F_1–F_2–F_3 space.

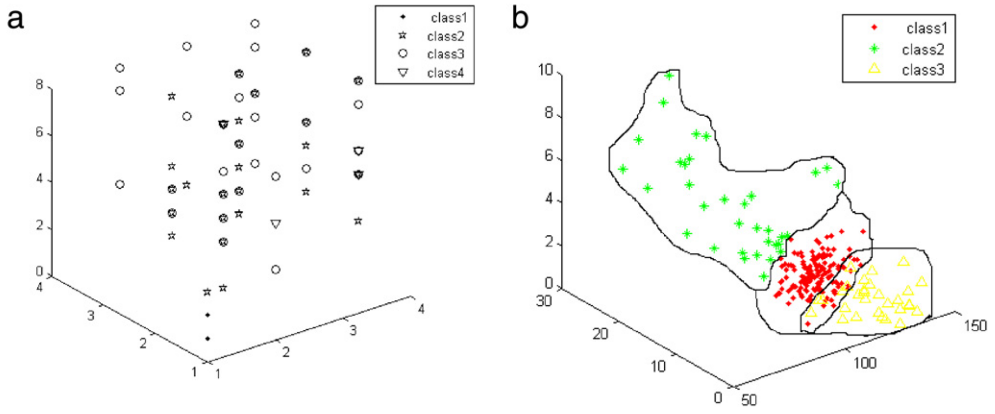


Fig. 5. (a) Lymphography data in the F_1–F_2–F_3 space. (b) Thyroid data in the F_1–F_2–F_3 space.

experiment, we considered $n = 10$ or 9 based on the size of the data sets. The parameters in Eq. (7) were chosen as $f_d = 6$ and $f_e = 1$ for all the data sets. However, the momentum parameter α, learning rate η, and the bias b of the FRGNN traversed a range of values between 0 and 1, and had different values depending on the folds used for learning. For example, in the case of vowel data, the appropriate values were $\alpha = 0.98$, $\eta = 0.0858$, $b = 0.00958$ in the crisp case, and $\alpha = 0.958$, $\eta = 0.06558$, $b = 0.0958$ in the fuzzy case. It is observed that the FRGNN converges to a local minimum at the 2000th epoch for glass data and ionosphere data. For the vowel data and the remaining data sets, the proposed FRGNN converges to a local minimum at the 1500th epoch.

Before providing the performance of the FRGNN on the test sets for all the data, we explain the implementation process for its training on Telugu vowel data, as an example. Here, the FRGNN has nine nodes in the input layer, and six nodes in each of hidden and output layers. The data is first transformed into a 3D granular space using Eq. (3). The appropriate values of the center C and scaling factor λ for each feature of granules, e.g., *low*, *medium*, or *high*, are determined as $C_{low_1} = 368.932$, $C_{medium_1} = 470.482$, $C_{high_1} = 583.616$, $C_{low_2} = 1110.323$, $C_{medium_2} = 1514.684$, $C_{high_2} = 2047.021$, $C_{low_3} = 2359.322$, $C_{medium_3} = 2561.021$, $C_{high_3} = 2755.891$, and $\lambda_{low} = 586.666$, $\lambda_{medium} = 1300.000$, $\lambda_{high} = 666.666$. Then, the 3D patterns have numerical components which are mapped into a 9D granular space with components $L_1, M_1, H_1, L_2, M_2, H_2, L_3, M_3, H_3$, while the desired vector has components $d_1, d_2, d_3, d_4, d_5, d_6$ corresponding to the six vowel classes. Table 5 provides some examples of a granular input vector $\overrightarrow{F}_i$, and the desired output vector d_k, $k = 1, 2, \ldots, 6$, for a set of sample patterns.

Table 5
Input target vectors for a set of sample patterns presented to the proposed model.

Input features			Input vector									Desired vector					
F_1	F_2	F_3	L_1	M_1	H_1	L_2	M_2	H_2	L_3	M_3	H_3	d_1	d_2	d_3	d_4	d_5	d_6
700	1500	2600	0.37	0.94	0.93	0.22	0.99	0.06	0.66	0.99	0.89	0.9	0.0	0.0	0.0	0.0	0.0
650	1200	2500	0.54	0.96	0.98	0.95	0.88	0.00	0.88	0.99	0.70	0.0	0.9	0.0	0.0	0.0	0.0
300	2100	2600	0.97	0.96	0.63	0.00	0.59	0.98	0.66	0.99	0.89	0.0	0.0	0.9	0.0	0.0	0.0
400	1150	2500	0.99	0.99	0.84	0.99	0.84	0.00	0.88	0.99	0.70	0.0	0.0	0.0	0.9	0.0	0.0
500	2000	2750	0.90	0.99	0.96	0.00	0.72	0.99	0.22	0.95	0.99	0.0	0.0	0.0	0.0	0.9	0.0
400	900	2700	0.99	0.99	0.84	0.74	0.55	0.00	0.35	0.97	0.98	0.0	0.0	0.0	0.0	0.0	0.9

Table 6
The FRGNN with initial connection weights in the crisp case for Telugu vowel data.

Input to hidden layer (w_{ij})						Hidden to output layer (w_{jk})					
0.0051	0.0045	0.0019	0.0002	0.0003	0.0014	0.0883	0.1574	0.0453	0.0346	0.0179	0.0258
0.0236	0.0085	0.0021	0.0016	0.0008	0.0006	0.0523	0.1377	0.0290	0.0348	0.02523	0.04833
0.03815	0.01357	0.0061	0.0133	0.0001	0.0004	0.1524	0.1084	0.0617	0.0876	0.0224	0.0348
0.0027	0.0020	0.0013	0.0029	0.0018	0.0023	0.1247	0.0586	0.0119	0.01556	0.0185	0.0141
0.0096	0.0057	0.0061	0.0076	0.0046	0.0066	0.0976	0.0081	0.0215	0.0195	0.0182	0.0425
0.0029	0.0097	0.0030	0.0030	0.0006	0.0002	0.0350	0.0112	0.0169	0.0407	0.0025	0.0596
0.0026	0.0055	0.0009	0.0023	0.0029	0.0071						
0.0004	0.0018	0.0015	0.0005	0.0011	0.0051						
0.0060	0.0068	0.0045	0.0026	0.0025	0.0059						

Table 7
The FRGNN with initial connection weights in the fuzzy case for Telugu vowel data.

Input to hidden layer (w_{ij})						Hidden to output layer (w_{jk})					
0.0747	0.0832	0.0759	0.0849	0.0937	0.0892	0.0936	0.097	0.1214	0.1347	0.1294	0.1371
0.0634	0.06202	0.0661	0.0713	0.0761	0.0757	0.0648	0.0615	0.0881	0.1101	0.0813	0.0950
0.0642	0.0655	0.0741	0.0868	0.0737	0.0798	0.0362	0.0472	0.0477	0.0603	0.0532	0.0463
0.1117	0.0948	0.1024	0.1079	0.1104	0.0862	0.0166	0.0082	0.0237	0.0172	0.0197	0.0179
0.0778	0.0893	0.0981	0.1114	0.1232	0.0962	0.1588	0.1616	0.1601	0.1636	0.1629	0.1663
0.1044	0.0890	0.1286	0.1106	0.1047	0.12675	0.0793	0.0675	0.0904	0.078	0.0836	0.0844
0.0681	0.0768	0.0868	0.0899	0.0763	0.09251						
0.06290	0.0620	0.0824	0.0697	0.0695	0.0845						
0.0768	0.0713	0.0870	0.0941	0.0765	0.0855						

Knowledge extraction procedure for Telugu vowel data:

Let the decision table $S = (U, \mathcal{A} \cup \{d\})$ be used to represent the entire training data set. The decision table S is then divided into six decision tables corresponding to six vowel classes, namely, $S_l = (U_l, \mathcal{A} \cup \{d\})$. Each S_l, $l = 1, 2, \ldots, 6$, need not be of the same size. Each S_l is transformed into a 3D granular space with Eq. (3). We apply the knowledge encoding procedure (which was explained before with respect to the data set in Table 1) for each decision table S_l, and the resulting domain knowledge is then encoded into the proposed FRGNN in the form of initial connection weights. Tables 6 and 7 provide the initial connection weights of the FRGNN in the crisp case (Method I of Section 4.3) and fuzzy case (Method II of Section 4.3), respectively. The values of the fuzzifiers f_d and f_e in Eq. (7) were considered to be 1. The complete FRGNN architecture thus generated for the vowel data is shown in Fig. 6. The network refines its initial weight parameters using the training samples. The trained network is then used to classify the test pattern set.

Note that, if the number of training patterns in any class is less than the number of classes, then it may not be possible to define the average dependency factors of all the conditional attributes with respect to a decision class. To avoid this, we include some training patterns with zero attribute value to that particular class so as to make the number of training patterns in that class equal to the number of classes.

Performance of the FRGNN on test sets:

The experimental results of the FRGNN on all the real-life data sets are shown in Table 8. The results correspond to three types of the initial weights of the FRGNN. These are (i) random numbers in [−0.5, 0.5], (ii) the crisp case (Method 1 of Section 4.3) and (iii) the fuzzy case (Method 2 of Section 4.3). One may note that considering the random initial weights (i.e., type (i)) makes the FRGNN equivalent to a fuzzy MLP [10].

The performance of the FRGNN is seen to vary over different folds. For example, in the ten-fold cross-validation, for Telugu vowel data with initial weights in the random case, the crisp case, and the fuzzy case, the recognition scores of the FRGNN are seen to vary between 79.01% (minimum) and 88.51% (maximum) with an average accuracy of 84.75%; 81.61% (minimum) and 90.80% (maximum) with an average accuracy of 86.55%; and 85.06% (minimum) and 94.25% (maximum) with an average accuracy of 87.71%, respectively. In the nine-fold cross-validation, for Pima Indian data with initial weights in the random case, the crisp case, and the fuzzy case, these figures are found to be 72.94% (minimum) and 83.53% (maximum)

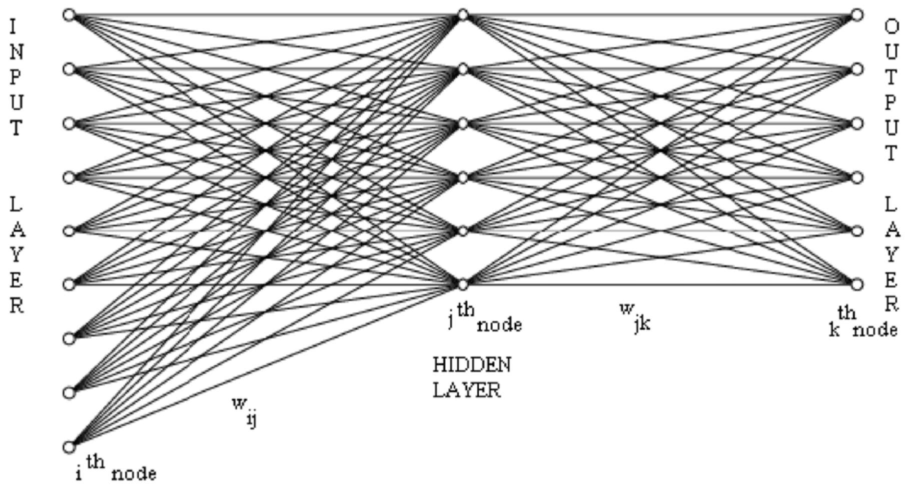


Fig. 6. The FRGNN for Telugu vowel data.

Table 8
Experimental results for the FRGNN.

Dataset name	# folds	Initial weights case	Each fold accuracy										Average accuracy
Telugu vowel	10	Random	82.76	85.06	83.91	85.06	85.06	86.21	86.21	79.01	88.51	85.06	84.75
		Crisp	85.06	85.06	90.8	87.36	87.36	86.21	85.06	81.61	88.51	88.51	86.55
		Fuzzy	86.21	86.21	94.25	89.66	86.21	86.21	86.21	85.06	88.51	88.51	87.71
Pima Indian	9	Random	75.29	82.35	70.59	80	72.94	74.12	72.94	74.12	78.82	–	75.69
		Crisp	77.65	83.53	74.12	81.18	74.12	75.29	76.47	74.12	78.82	–	77.26
		Fuzzy	77.65	83.53	74.12	81.18	74.12	76.47	76.47	74.12	84.71	–	78.04
Sonar	9	Random	86.96	86.96	73.91	86.96	91.3	82.61	82.61	86.96	86.96	–	85.03
		Crisp	91.3	91.3	73.91	86.96	91.3	82.61	82.61	86.96	91.3	–	86.47
		Fuzzy	95.65	91.3	78.26	86.96	100	86.96	78.26	86.96	95.65	–	88.89
Lymphography	9	Random	87.5	81.25	75	75	81.25	81.25	75	87.5	75	–	79.86
		Crisp	87.5	87.5	75.00	75.00	87.5	81.25	81.25	93.75	81.25	–	83.33
		Fuzzy	100	87.5	75	81.25	93.75	81.25	81.25	100	81.25	–	86.81
Glass	10	Random	71.43	71.43	76.19	76.19	76.19	71.43	76.19	71.43	71.43	71.43	73.34
		Crispe	71.43	71.43	80.95	71.43	80.95	76.19	71.43	76.19	80.95	76.19	75.71
		Fuzzy	71.43	76.19	80.95	76.19	80.95	76.19	71.43	71.43	71.43	85.71	76.19
Ionosphere	10	Random	91.43	88.57	91.43	91.43	88.57	88.57	88.57	97.14	88.57	94.29	90.86
		Crisp	88.57	94.29	91.43	91.43	91.43	91.43	88.57	97.14	88.57	94.29	91.72
		Fuzzy	88.57	94.29	91.43	91.43	91.43	91.43	91.43	97.14	91.43	94.29	92.29
Thyroid	10	Random	95.45	95.45	95.45	95.24	95.45	86.36	90.48	100.00	100.00	100.00	95.39
		Crisp	95.45	95.45	95.45	95.24	95.45	90.48	95.24	100.00	100.00	100.00	96.28
		Fuzzy	95.45	95.45	95.45	95.24	95.45	90.48	95.24	100.00	100.00	100.00	96.28
Wine	10	Random	95.45	95.45	90.91	100	90.91	95.45	95.45	100.00	100.00	100.00	96.36
		Crisp	95.45	95.45	90.91	100	90.91	100.00	100.00	100.00	100.00	100.00	97.27
		Fuzzy	95.45	95.45	90.91	100	90.91	100.00	100.00	100.00	100.00	100.00	97.27

with an average accuracy of 75.69%; 74.12% (minimum) and 90.80% (maximum) with an average accuracy of 77.26%; and 74.12% (minimum) and 83.53% (maximum) with an average accuracy of 78.04%, respectively. For sonar data with nine-fold validation, the corresponding figures are found to be 73.91% (minimum) and 91.30% (maximum) with an average accuracy of 85.03%; 73.91% (minimum) and 91.30% (maximum) with an average accuracy of 86.47%; and 78.26% (minimum) and 100.00% (maximum) with an average accuracy of 88.89%.

The generalization ability of the FRGNN is seen to depend on the characteristics of the data. If the correlation among input variables is high, the generalization ability is not so high, and vice versa. For example, the generalization ability of the FRGNN for Telugu vowel, pima Indian, sonar, lymphography, and glass data is below 90% because of the high correlation among their input variables, whereas it is above 90% for thyroid, wine, and ionosphere data, which have comparatively less correlation among the input variables.

From Table 8, we can conclude that the performance of the FRGNN with initial connection weights corresponding to both fuzzy and crisp cases is superior to that of the fuzzy MLP (i.e., the FRGNN with initial weights corresponding to the

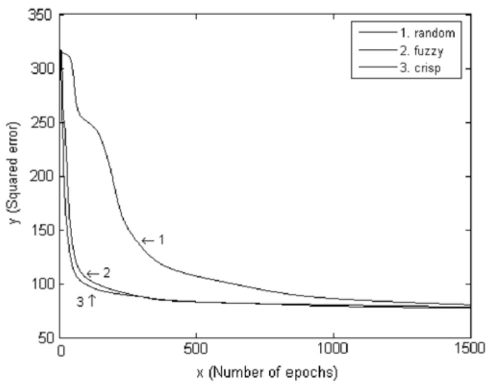


Fig. 7. Comparison of squared errors of the FRGNN with initial weights for Telugu vowel data. 1: random case, 2: crisp case, 3: fuzzy case.

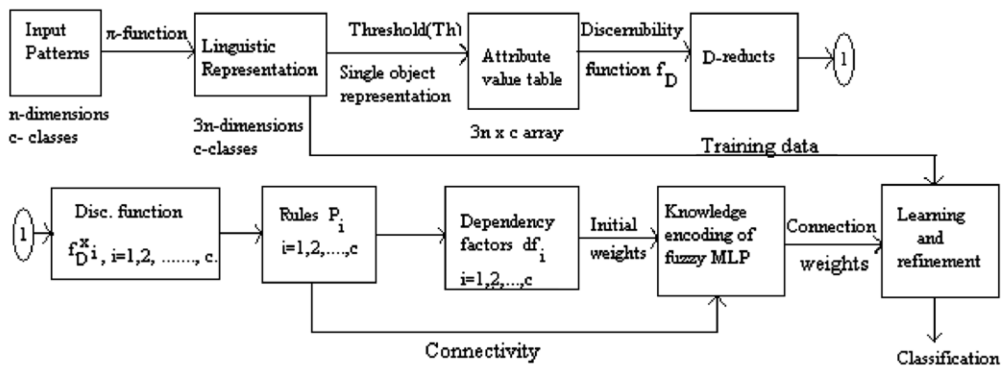


Fig. 8. Block diagram of the knowledge encoding procedure in Method I.

random case) for all the data sets. Weight selection by Method II of Section 4.3 (fuzzy case) results in better performance than Method I of Section 4.3 (crisp case). Further, their difference is more apparent for overlapping classes; for example, see Fig. 5(a) for lymphography data, where the difference is seen to be more than 3%.

Fig. 7 presents the variation of the squared error with the number of epochs carried out during training of the FRGNN with initial weights in random, crisp, and fuzzy cases. This comparative result is shown, as an example, only for Telugu vowel data. Here, the results correspond to one fold over ten folds. As expected, the squared error decreases with the number of epochs. The error drops significantly at lower number of epochs for the crisp and fuzzy cases than the random case of weight selection, because the former two cases enable the FRGNN to start learning from a much better position than the latter case. Among the crisp and fuzzy cases, although the crisp one is seen to provide slightly better results for the particular fold of vowel data (considered in Fig. 7), in overall performance, the fuzzy case of weight selection is the best for all the data sets (see Table 8).

Comparison with the rough fuzzy MLP:

In order to compare the performance of our network against an existing well-known network of a rough fuzzy MLP [13], we consider the Telugu vowel data, as an example. The domain knowledge encoding procedure in the rough fuzzy MLP is explained in the following two methods.

Method I

Fig. 8 shows a complete algorithm for Method I. Here, the data is first transformed into a 3D linguistic space. A threshold *Th* ($0.5 \leq Th < 1$) is imposed on the resultant linguistic data in order to make them binary (0 or 1). The most representative template, i.e., the one with the maximum number of occurrences, is selected from the set of all the templates in the resultant binary-valued data to serve as an object in a decision table. Similarly, the other most representative templates are selected from the remaining classes of the resultant binary-valued data. Then the resultant information is represented in the decision table. Knowledge reduction now consists of eliminating superfluous values of the conditional attributes by computing their generalized decision reducts (D-reducts). For each D-reduct, the discernibility matrix is defined such that the discernibility functions are obtained from the discernibility matrix. Each discernibility function gives rise to a decision rule. These rules are used to define the dependency factors which are encoded into the conventional MLP.

Method II

Fig. 9 shows a complete algorithm for Method II, in which the data is first transformed into a 3D linguistic space and made binary, as in Method I. The binary-valued data is represented in a decision table S. Then the decision table S is divided into S_i, $i = 1, 2, \ldots, c$, corresponding to c decision attributes, where c is the number of classes. For each decision table S_i, objects are

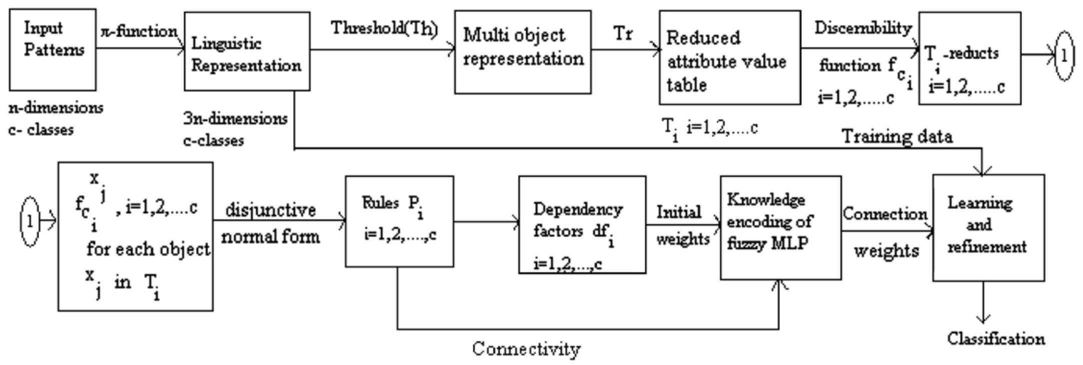

Input Patterns
π-function
n-dimensions c- classes
Linguistic Representation
3n-dimensions c-classes
Threshold(Th)
Multi object representation
Tr
Reduced attribute value table
T_i i=1,2,....c
Discernibility function f_ci i=1,2,.....c
T_i -reducts i=1,2,.....c
Training data
f_ci^xj , i=1,2,....c for each object x_j in T_i
disjunctive normal form
Rules P_i i=1,2,....,c
Dependency factors df_i i=1,2,...,c
Initial weights
Knowledge encoding of fuzzy MLP
Connection weights
Learning and refinement
Classification
Connectivity

Fig. 9. Block diagram of the knowledge encoding procedure in Method II.

Table 9
Experimental results for the rough fuzzy MLP.

Dataset name	# folds	Method	Each fold accuracy										Average accuracy
Telugu vowel	10	Method I	80.46	89.66	82.76	86.21	83.91	87.36	83.91	86.21	87.36	90.8	85.86
			83.91	83.91	87.36	90.80	80.46	87.6	83.91	82.76	88.51	89.66	85.86
			83.91	87.36	82.76	87.36	82.76	87.36	85.06	81.61	87.36	88.51	86.41
			83.91	85.06	81.61	89.66	80.46	88.51	83.91	82.76	89.66	90.80	85.63
	10	Method II	81.61	83.91	87.36	88.51	87.36	87.36	87.36	83.91	88.51	89.66	86.55

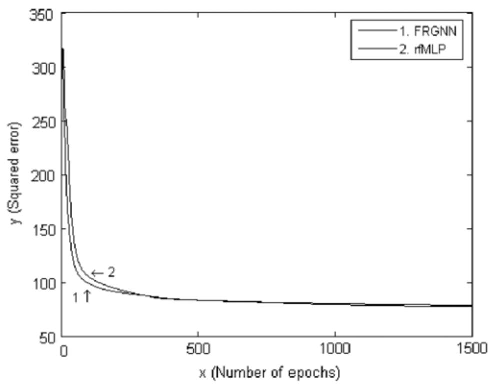


Fig. 10. Comparison of squared errors of the FRGNN and rough fuzzy MLP.

grouped into different sets based on the number of occurrences of the patterns of same attribute values. The objects in the decision table are arranged in decreasing order based on the frequency of occurrences of the set of multi-objects. A threshold value (Tr) is determined based on the frequency of occurrences of the set of multi-objects. All objects having frequency of occurrences less than the threshold (Tr) are eliminated from the decision table. Thus the reduced decision table T_i is created with the help of a threshold (Tr). For each T_i, a discernibility matrix is defined. A discernibility function f_{c_i}, $i = 1, 2, \ldots, c$, is generated corresponding to the discernibility matrix. Based on the discernibility matrix, T_i-decision reducts are generated using the disjunctive normal form. For each T_i-decision reduct, the discernibility matrix is defined, so is the discernibility function $f_{c_i}^{x_j}$ for each object x_j in T_i. Each discernibility function gives rise to a decision rule. The rules, thus formed, are used to determine the dependency factors df_i, $i = 1, 2, \ldots, c$, which are then considered as the initial connection weights of the conventional MLP.

Table 9 demonstrates the performance of the rough fuzzy MLP, with one hidden layer as in the FRGNN, for classification of vowels after 1500 epochs for a typical set of ten folds. The parameters of the membership function (f_d and f_e), learning rate (η), momentum parameter (α), and bias b were assigned the same values as in the FRGNN for the purpose of fair comparison. Note that Method I produces four reducts combing all the six classes, and each reduct represents a set of six decision rules corresponding to six vowel classes. On the other hand, in Method II, we have obtained one reduct for each class representing its decision rule, thereby generating six decision rules for six vowel classes. That is why Table 9 has four rows for Method I but one row for Method II. Comparing Tables 8 and 9, we can say that the performance of the FRGNN with weights in the fuzzy case is better than those of both Methods I and II of the rough fuzzy MLP.

In Fig. 10, we present the comparison of squared errors of the FRGNN (with initial weights in the fuzzy case) and the rough fuzzy MLP (Method II) for one of the ten-folds, considered in Table 9, for which both the FRGNN and rough fuzzy MLP have given the best classification accuracy within their respective sets of folds for vowel data. In the FRGNN, the number of nodes in the hidden layer is equal to the number of classes, which is six in the case of vowel data. Interestingly, the same number of nodes was also obtained automatically in the rough fuzzy MLP using its rough dependency factors. Again, the minimum values in both the cases were reached at 1500th epoch. Considering the variation of average accuracy rate, Fig. 10 further supports the earlier findings regarding the superiority of the FRGNN over the rough fuzzy MLP for Telugu vowel data. Although the comparative result is shown here only for vowel data, to restrict the size of the paper, the same observation holds good for other data sets too.

Salient points of the difference between the FRGNN and the rough fuzzy MLP:

In the rough fuzzy MLP, basically rough sets are integrated with a fuzzy MLP for encoding domain knowledge in network parameters, whereas it is fuzzy rough sets which are integrated with a fuzzy MLP in the FRGNN.

As stated in Method II of the rough fuzzy MLP, a threshold value (Tr) is used to eliminate the noisy patterns from input data. In contrast, no such threshold value is required in the FRGNN, and the problem of removing noisy patterns is resolved by incorporating the concept of fuzzy rough sets in determining the initial weights of the network. Again, in Method II of the rough fuzzy MLP, it was found that the attribute reducts could not be deducted for class 3 of the thyroid data since the reduced attribute table contained only one pattern after applying the threshold (Tr) to the samples of class 3. So, the decision rules could not be generated for class 3. This problem did not occur in the FRGNN.

In the rough fuzzy MLP, the decision rules are generated for each reduct of the reduct set, and the dependency factors of these rules are mapped into the connection weights of the network. In contrast, no such attribute reducts are generated in the FRGNN, and the dependency factors of all the conditional attributes and the average value of all the dependency factors of all conditional attributes, with respect to the decision classes, are defined in the form of initial connection weights of the network.

In the rough fuzzy MLP, the network architecture is modeled for each reduct of the reduct set, and every such network gives a recognition score for the test set. The maximum recognition score computed over them is then considered as the performance measure of the rough fuzzy for the concerned test set. In contrast, only one network architecture is modeled in the FRGNN corresponding to the entire training data, and the network is seen to perform better than the rough fuzzy MLP for the same test set.

7. Conclusions

In this paper, we have presented the design procedure of a new granular neural network model in a natural computing framework by integrating the concept of fuzzy rough sets with a multilayer perceptron (MLP) using a back-propagation algorithm. Granularity is incorporated both at the input level and in determining the weights of the network. The network accepts input in terms of fuzzy granules (*low*, *medium* and *high*), and provides output decisions in terms of class membership values and zeros. The dependency factors, generated by the object space, are partitioned into fuzzy equivalence granules based on fuzzy similarity relations, and then used in the form of initial connection weights of the network instead of random numeric connection weights. This investigation not only demonstrates a way of integrating fuzzy rough sets with a fuzzy neural network, but also provides a methodology that is capable of generating a granular neural network architecture and improving its performance. The fuzzy rough set provides a means by which discrete real-valued noise data can be effectively reduced without the need for any user-supplied information. Additionally, the fuzzy partitions corresponding to each attribute can be automatically derived by fuzzy similarity relations and fuzzy logical connectives.

We have examined two special types of granular computation; one is induced by *low*, *medium*, and *high* fuzzy granules and the other has classes of granulation structures induced by a set of fuzzy equivalence granules based on a fuzzy similarity relation. With respect to the classes of granulation structures, stratified fuzzy rough set approximations are obtained to determine the dependency factors of all conditional attributes to obtain the initial weights of the FRGNN. The incorporation of granularity at various levels of the conventional MLP helps the resulting FRGNN to efficiently handle uncertain and ambiguous input information. This is demonstrated with extensive experimental results on a set of eight real-life data sets with varying dimension and size. The FRGNN was found to be superior to a rough fuzzy MLP which uses rough sets, rather than fuzzy rough sets, for knowledge encoding. Although the comparative results are shown only for vowel data, the same observation holds for all other data sets considered in the experiment. The FRGNN architecture reflects a useful application of granular computing to real-world classification problems.

Acknowledgements

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

References

[1] L.A. Zadeh, A new direction in AI: towards a computational theory of perceptions, AI Magazine 22 (2001) 73–84.
[2] L.A. Zadeh, Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, Fuzzy Sets and Systems 19 (1997) 111–127.
[3] S.K. Pal, Computational theory perception (CTP), rough-fuzzy uncertainty analysis and mining in bioinformatics and web Intelligence: a unified framework, LNCS Transactions on Rough Set 5946 (2009) 106–129.

[4] L. Kari, G. Rozenberg, The many facets of natural computing, Communications of the ACM 51 (10) (2008) 72–83.
[5] L.A. Zadeh, Fuzzy logic, neural networks, and soft computing, Communications of the ACM 37 (1994) 77–84.
[6] L.A. Zadeh, Fuzzy sets, Information and Control 8 (1965) 338–353.
[7] S.K. Pal, D. Dutta Majumder, Fuzzy Mathematical Approach to Pattern Recognition, John Wiley, Halsted, New York, 1986.
[8] R.P. Lippmann, An introduction to computing with neural networks, IEEE Magazine on Acoustics, Speech, and Signal Processing 61 (1987) 4–22.
[9] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall, New Jersey, 1999.
[10] S.K. Pal, S. Mitra, Multilayer perceptron, fuzzy sets, and classification, IEEE Transactions on Neural Networks 3 (5) (1992) 683–697.
[11] R. Jensen, Q. Shen, New approaches to fuzzy-rough feature selection, IEEE Transactions on Fuzzy Systems 17 (2009) 824–838.
[12] C. Cornelis, R. Jensen, G. Hurtado, D. Slezak, Attribute selection with fuzzy decision reducts, Information Sciences 180 (2010) 209–224.
[13] M. Banerjee, S. Mitra, S.K. Pal, Rough fuzzy MLP: knowledge encoding and classification, IEEE Transactions on Neural Networks 9 (6) (1998) 1203–1216.
[14] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning about Data, Kluwer, Netherlands, 1991.
[15] Z. Pawlak, A. Skowron, Rough sets and boolean reasoning, Information Sciences 177 (2007) 41–73.
[16] Z. Pawlak, A. Skowron, Rough sets: some extensions, Information Sciences 177 (2007) 28–40.
[17] Z. Pawlak, A. Skowron, Rudiments of rough sets, Information Sciences 177 (2007) 3–27.
[18] Y.Y. Yao, Stratified rough sets and granular computing, in: Proc. International Conference on Fuzzy Information Processing Society, 1999, pp. 800–804.
[19] S.K. Pal, A. Skowron (Eds.), Rough–Fuzzy Hybridization: A New Trend in Decision Making, Springer-Verlag, Singapore, 1999.
[20] D. Dubois, H. Prade, Rough fuzzy sets and fuzzy rough sets, International Journal of General Systems 17 (1990) 91–209.
[21] P. Lingras, R. Jensen, Survey of rough and fuzzy hybridization, in: Proc. IEEE International Conference on Fuzzy Systems, 2007, pp. 1–6.
[22] M. Banerjee, S.K. Pal, Roughness of a fuzzy set, Information Sciences 93 (1996) 235–246.
[23] L.A. Zadeh, From computing with numbers to computing with words — from manipulation of measurements to manipulation of perceptions, IEEE Transactions on Circuits and Systems 46 (1999) 105–119.
[24] G.J. Klir, T. Folger, Fuzzy Sets, Uncertainty and Information, Prentice Hall, New Jersey, 1988.
[25] D. Sen, S.K. Pal, Generalized rough sets, entropy and image ambiguity measures, IEEE Transactions on Systems, Man and Cybernetics - Part B 39 (1) (2008) 117–128.
[26] L. Fariñas del Cerro, H. Prade, in: A. Di Nola, A.G.S. Ventre (Eds.), Rough sets, two fold fuzzy sets and logic-fuzziness in indiscernibility and partial information, Springer-Verlag, Koln, 1986, pp. 103–120.
[27] A. Nakamura, Fuzzy rough sets, Notes on Multiple-Valued Logic in Japan 9 (1988) 1–8.
[28] J.S. Mi, Y. Leung, H.Y. Zhao, T. Feng, Generalized fuzzy rough sets determined by a triangular norm, Information Sciences 178 (16) (2008) 3203–3213.
[29] S.K. Pal, D. Dutta Majumder, Fuzzy sets and decision making approaches in vowel and speaker recognition, IEEE Transaction on Systems, Man, and Cybernetics 7 (1977) 625–629.
[30] D.J. Newman, S. Hettich, C.L. Blake, C.J. Merz, UCI repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science, 1998. http://archive.ics.uci.edu/ml/.
[31] S.K. Pal, L. Polkowski, A. Skowron (Eds.), Rough-neural Computing: Techniques for Computing with Words, Springer-Verlag, Germany, 2004, pp. 698–699.
[32] Y.Q. Zhang, M.D. Fraser, R. Gagliano, A. Kandel, Granular neural networks for numerical–linguistic data fusion and knowledge discovery, IEEE Transactions on Neural Networks 11 (2000) 658–667.
[33] A. Vasilakos, D. Stathakis, Granular neural networks for land use classification, Soft Computing 9 (2005) 332–340.
[34] Y.Q. Zhang, B. Jin, Y. Tang, Granular neural networks with evolutionary interval learning, IEEE Transactions on Fuzzy Systems 16 (2008) 309–319.
[35] J.S.R. Jang, ANFIS: adaptive-network-based fuzzy inference systems, IEEE Transaction on Systems, Man, and Cybernetics 23 (1993) 665–685.
[36] S. Dick, A. Kandel, Granular computing in neural networks, in: W. Pedrycz (Ed.), Granular Computing: An Emerging Paradigm, Physica Verlag, New York, 2001, pp. 275–305.
[37] M. Szczuka, Refining classifiers with neural networks, International Journal of Computer and Information Sciences 16 (2001) 39–56.