Int. J. Appl. Math. Comput. Sci., 2000, Vol.10, No.4, 723–745

NEURAL NETWORKS IN THE FRAMEWORK OF GRANULAR COMPUTING

WITOLD PEDRYCZ*,**

The study is concerned with the fundamentals of granular computing and its application to neural networks. Granular computing, as the name itself stipulates, deals with representing information in the form of some aggregates (embracing a number of individual entitites) and their ensuing processing. We elaborate on the rationale behind granular computing. Next, a number of formal frameworks of information granulation are discussed including several alternatives such as fuzzy sets, interval analysis, rough sets, and probability. The notion of granularity itself is defined and quantified. A design agenda of granular computing is formulated and the key design problems are raised. A number of granular architectures are also discussed with an objective of dealineating the fundamental algorithmic and conceptual challenges. It is shown that the use of information granules of different size (granularity) lends itself to general pyramid architectures of information processing. The role of encoding and decoding mechanisms visible in this setting is also discussed in detail along with some particular solutions. Neural networks are primarily involved at the level of numeric optimization. Granularity of information introduces another dimension to the neurocomputing. We discuss the role of granular constructs in the design of neural networks and knowledge representation therein. The intent of this paper is to elaborate on the fundamentals and put the entire area in a certain perspective while not moving into specific algorithmic details.

Keywords: information granulation, pyramid architectures, encoding and decoding, neural networks, learning, knowledge representation

1. Introduction

Last years saw a rapid growth of interest in so-called granular computing and computing with words as one among its realizations (Zadeh, 1996; 1997). In a nutshell, granular computing is geared toward representing and processing basic chunks of information — information granules (Pedrycz, 1998; Zadeh, 1979). Information granules, as the name itself stipulates, are collections of entities, usually originating at the numeric level, that are arranged together due to their similarity, functional adjacency, indistinguishability or alike. The process of forming information granules is referred

* University of Alberta, Department of Electrical and Computer Engineering, Edmonton, Canada, e-mail: pedrycz@ee.ualberta.ca

** Polish Academy of Sciences, Systems Research Institute, 01–447 Warsaw, Poland.

to as information granulation. No matter how this granulation proceeds and what fundamental technology becomes involved therein, there are several essential factors that drive all pursuits of information granulation. These factors include

- A need to split the problem into a sequence of more manageable and smaller subtasks. Here granulation serves as an efficient vehicle to modularize the problem. The primary intent is to reduce an overall computing effort.
- A need to comprehend the problem and provide with a better insight into its essence rather than get buried in all unnecessary details. In this sense, granulation serves as an abstraction mechanism that reduces an entire conceptual burden. As a matter of fact, by changing the 'size' of the information granules, we can hide or reveal a certain amount of details one intends to deal with during a certain design phase.

The long-lasting tradition of computing using some specific information granules is a visible testimony that some specific versions of granular computing are omnipresent indeed. As a matter of fact, as we discuss in this study, digital-to-analog transformation leading to digital computing for the analog world is just a highly representative (albeit quite specific) instance of granular computing. By tradition (and the associated technology dominant at that time), we have embarked on the digital world of computing. To interact with the continuous (analog) world, we use set-based granulation (more specifically, interval-valued granulation). This specific type of granulation comes under the name of analog-to-digital conversion.

Information granules may arise as a phenomenon of inherent nonuniqueness associated with the problem at hand. As a simple example, one can resort himself to any inverse problem; the type of characteristics involved (as the functions may be non-invertible) gives rise to relations and as a result, a collection of information granules rather than single numeric quantities. Dropping some input variable in a model may also lead to the same effect of granular information.

We may witness (maybe not always that clearly and profoundly) that the concept of granular computing tends to permeate a number of significant endeavors. The reason is quite straightforward. Granular computing as opposed to numeric computing is *knowledge-oriented*. Numeric computing is *data-oriented*. Undoubtedly, knowledge-inclined processing is a cornerstone of data mining, intelligent databases, hierarchical control, etc.

While the idea of granular computing has been advocated and spelled out in the realm of fuzzy sets (and seems to be a bit biased in this way), there are a number of fundamental formal frameworks that can be exploited as well. Several alternative paths to follow include interval analysis, rough sets and probabilistic environments, to name a few dominant and most visible options.

The diversity of the formal means used for information granulation and further processing of the resulting information granules has a common denominator. All of these environments share the same research agenda that attempts to address the fundamentals of granular computing.

A way of constructing information granules and describing them in an analytical fashion is a common problem no matter which path (probability or set-theoretic) we follow. The question as to the definition of the 'size', 'capacity' or 'dimension' of the information granule is of primordial interest. How to measure the granularity of the constructed information granules? How to relate this granularity with computational complexity? Those are open questions in the framework of granular computing that still await solid answers.

What are sound methodologies when operating on information granules? How to evaluate (validate and verify) granular constructs? What would be appropriate measures of relevance of granular models? These are fundamental issues posed in the case of numeric modeling and discussed in detail. The same suite of questions expressed in the case of granular architectures begs for further thorough investigations.

There is an intriguing question as to a way of navigating between constructs (models) developed at various levels of information granularity. Is the structure developed with the use of 'large' information granules useful when more specific results are required? It is apparent that when forming information granules, the contributing elements lose their identity that is essentially a non-recoverable process. Now, how this could effect the results of computing involving bigger information granules? If we want to recover the details, how efficient could be our attempt? What are the limits of this reconstruction? These aspects boil down to the mechanisms of encoding and decoding granular information. When any datum enters a system operating at a certain level of information granularity, it becomes encoded. As a result it becomes 'accepted' (tuned) to the level of information granularity present within this system. Once the system tends to communicate its results, these need to be decoded. In other words, encoding and decoding are interfaces between worlds (systems) operating at various levels of information granularity. We have already encountered this scheme in digital processing: encoding corresponds to the analog-to-digital (A/D) conversion whereas the decoding comes under the name of digital-to-analog (D/A) conversion.

Fuzzy modeling has emerged as an interesting, attractive, and powerful modeling environment applied to numerous system identification tasks. Granular computing forms a useful environment supporting all modeling pursuits and adding another dimension to the modeling itself. The key features being emphasized very often in this setting concern a way in which fuzzy sets enhance or supplement the existing identification schemes. It was Zadeh (1979) first who has introduced the concept of fuzzy models and fuzzy modeling. The enhancements of system modeling conceived within this framework take place at the conceptual level as well as at the phase of detailed algorithms. In a nutshell, fuzzy models are concerned with the modeling pursuit that occurs at the level of linguistic granules (fuzzy sets or fuzzy relations) rather than the one that happens at a detailed and purely numeric level encountered in other modeling approaches. What fuzzy sets offer in system modeling is another more general and holistic view at the resulting model that gives rise to their augmented interpretation and better utilization. From a computational point of view, fuzzy sets are inherently nonlinear (viz. their membership functions are nonlinear mappings). As a consequence of such a nonlinear character, one may anticipate that this feature augments the representation power of the fuzzy models. There have been a substantial number of various schemes of fuzzy modeling along with specific algorithmic variations that help eventually capture some specificity of the problem at hand and contribute to the efficiency of the overall identification schemes, cf. (Kruse *et al.*, 1994; Tsoukalas and Uhrig, 1997). Quite often, in order to take advantage of numeric experimental data, the modeling algorithms resort themselves to a vast spectrum of neurofuzzy techniques, see e.g., (Bortolan, 1998; Buckley and Hayashi, 1984; Harris *et al.*, 1993; Jang, 1993; Jang *et al.*, 1997; Kasabov, 1996; Kruse *et al.*, 1994; Pedrycz, 1997; Tsoukalas and Uhrig, 1997).

Granulation is a necessary prerequisite that is required to take advantage of discrete models (where by 'discrete' we mean granular) such as finite-state machines, Petri nets, and alike.

Indisputibly, neural networks enjoy a rapid growth that has materialized in a vast number of architectures, learning algorithms and applications. By and large, neurocomputing is inherently geared to numeric computing. In a nutshell, neural networks support the bottom-up development approach: we start with clouds of numeric data that are captured into more concise nonlinear mappings through intensive learning.

The objective of this study is twofold. First, we raise fundamental issues of granular computing as a new and unified paradigm of information processing, elaborate on a family of possible formal frameworks and formulate the key design problems associated with this form of computing. Second, we show how granular computing gives rise to a new broad category of granular neural networks and granular neurocomputing.

2. Granular Computing: An Information Processing Pyramid

In granular computing, we operate on information granules. Information granules exhibit different levels of granularity. Depending upon the problem at hand, we usually group granules of similar 'size' (that is granularity) together in a single layer. If more detailed (and computationally intensive) processing is required, smaller information granules are sought. Then these granules are arranged in another layer. In total, the arrangement of this nature gives rise to the information pyramid. As portrayed schematically in Fig. 1, in granular processing we encounter a number of conceptual and algorithmic layers indexed by the 'size' of information granules. Information granularity implies the usage of various techniques that are relevant for the specific level of granularity. Alluding to system modeling, we can refine Fig. 1 by associating the layers of the information processing pyramid with the pertinent most commonly used classes of processing and resulting models:

- at the lowest level we are concerned with numeric processing; this is a domain completely overwhelmed by numeric models such as differential equations, regression models, neural networks, etc.,
- at the intermediate level we encounter larger information granules (viz. those embracing more individual elements),
- the highest level can be solely devoted to symbol-based processing and as such invokes well-known concepts of finite state machines, bond graphs, Petri nets,

qualitative simulation, etc. Note that some of these classes emerge at the intermediate level of information granularity and at that level their conceptual and symbolic fabric is usually augmented with some numeric component.

The general characteristics of the principle of granular computing can be enumerated as shown in Table 1.

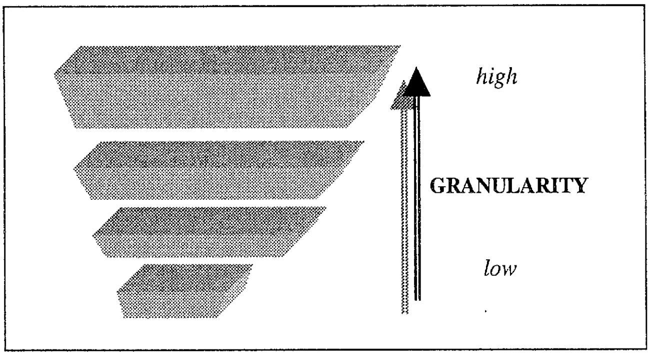


Fig. 1. An information-processing pyramid (the respective layers are indexed by the corresponding level of information granularity).

Table 1. The fundamental features of granular computing.

Allow for multiple abstraction levels (granularity levels)
Allow for several methods of traversing various levels of hierarchy (encoding-decoding mechanisms)
Allow for nonhomogeneous methods (differential or difference equations, Petri nets, finite state machines)

3. Information Granulation

In this section, we look into the underlying rationale behind information granulation and discuss various means supporting the construction of information granules. The starting point is to look at any linguistic model as an association of information granules (linguistic terms) defined over some variables of the system. Quite descriptively, one may allude to such linguistic granules or linguistic landmarks as being a focal point of all modeling activities. Linguistic granules are viewed as linked collections of objects (data points, in particular) drawn together by the criteria of indistinguishability, similarity or functionality. Such collections can be modeled in several formal environments including set theory, rough sets, random sets, shadowed sets or fuzzy sets.

Informally speaking, information granules (Zadeh, 1979; 1996; 1997) are viewed as linked collections of objects (data points, in particular) that are drawn together by the criteria of indistinguishability, similarity or functionality. Information granules and the ensuing process of information granulation constitute a vehicle of abstraction leading to the emergence of concepts.

Granulation of information is an inherent and omnipresent activity of human beings carried out with the intent of better understanding of the problem. In particular, granulation of information is aimed at splitting the problem into several manageable chunks. In this way, we partition the problem into a series of well-defined subproblems (modules) of a far lower computational complexity than the original one. Granulation is related to the notion of *abstraction*. Likewise in abstraction, we are concerned with forming general concepts by identifying similarity between elements belonging to the same category.

Granulation occurs everywhere; the examples are numerous and they originate from various areas:

- We granulate information over time by forming information granules over predefined time intervals. For instance, one computes a moving average with its confidence intervals.

- In any computer model we granulate memory resources by subscribing to the notion of pages of memory as its basic operational chunks (then we may consider various swapping techniques to facilitate an efficient access to individual data items).

- We granulate information available in the form of digital images—the individual pixels are arranged into larger entities and processed as such. This leads us to various issues of scene description and analysis.

- In describing any problem, we tend to shy away from numbers. Instead, we tend to use aggregates and building rules (*if-then statements*) that dwell on them.

- We live in an inherently analog world. Computers, by tradition and technology, perform processing in a digital world. Digitization of this nature (that dwells on set theory – interval analysis) is an example of information granulation.

- All mechanisms of data compression are examples of information granulation that is carried in a certain sense.

Overall, there is a profound diversity of the situations that call for information granulation. There is also a panoply of possible formal vehicles to be used to capture the notion of granularity and provide with a suitable algorithmic framework in which all granular computing can be efficiently completed. In the ensuing section, we elaborate on those commonly encountered in the literature. Examples of such formal environments include set theory, rough sets, random sets, shadowed sets or fuzzy sets.

The idea of information granules and the size of information granules themselves gives rise to an important and application-driven issue of usefulness of information granules. One should stress that the size of information granules is dictated by the particular application or a certain category of users. This effect of variable usefulness of information granules versus their level of granularity is illustrated in Fig. 2. In these considerations, we express granularity through measures such as cardinality or σ-count, i.e.

$$\mathrm{Card}\,(A) = \int_{\mathbb{X}} A(x)\,\mathrm{d}x,$$

where A is an information granule under consideration.

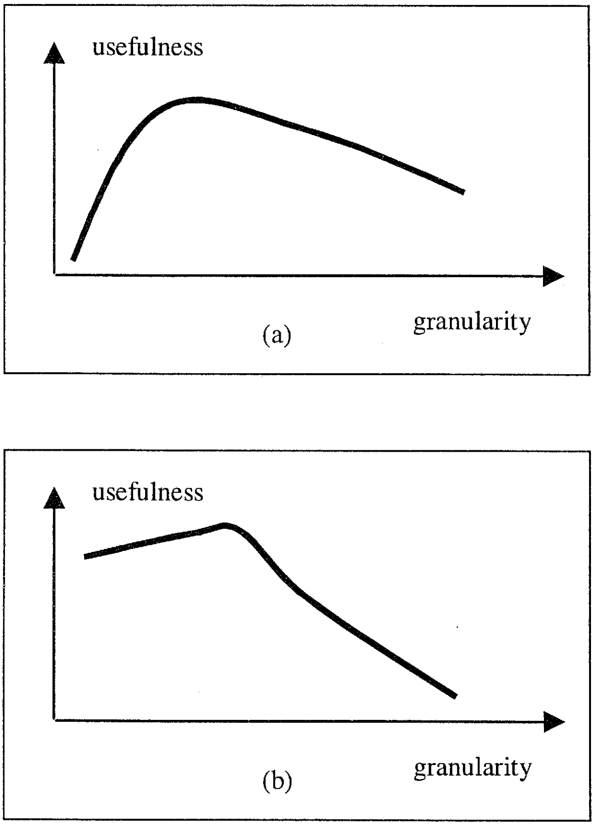
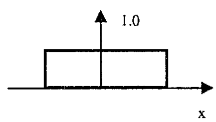
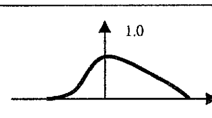
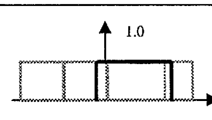
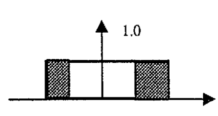


Fig. 2. Usefulness of information granules as a function of their granularity.

These two figures exhibit quite distinct patterns of behavior: in Fig. 2(a) we observe that the the usefulness of information is quite low when dealing with information granules of lower granularity while the usefulness does not change drastically when moving towards higher granularity. An opposite pattern appears in Fig. 2(b).

There are a number of fundamental approaches to information granules and information granulation. Table 2 summarizes the key formal frameworks of information granulation, identifies the notation being used therein and underlines the main features of such approaches.

Table 2. Selected models of information granulation and their brief characterization.

Formal model of information granules	Notation; $\mathbb{X}$ is a universe of discourse (space)	Example	Characterization and applications
Set theory	$\mathcal{P}(\mathbb{X})$	 $A: \mathbb{X} \to \{0,1\}$	Basic model of information granules. It generalizes and encapsulates a collection of elements. Sets are described by two-valued characteristic functions. Elements in sets are indistinguishable. For $\mathbb{X} = \mathbb{R}$ set theory gives rise to interval analysis.
Fuzzy sets	$\mathcal{F}(\mathbb{X})$	 $A: \mathbb{X} \to [0,1]$	Model of concepts with continuous rather than abrupt boundaries. Membership function captures a notion of partial membership.
Rough sets	$\mathcal{R}(\mathbb{X})$	 $A = \langle \mathbb{X}, A_*, A^* \rangle$ where A_* and A^* are lower and upper bounds	Dwell on the notion of indiscernibility relation. Concepts are described by their upper and lower bounds.
Shadowed sets	$\mathcal{S}(\mathbb{X})$	 $A: \mathbb{X} \to \{0,1,[0,1]\}$	Model of concepts with ill-defined boundaries. The boundaries are captured through intervals of possible membership values rather than a single specific membership value. Shadowed sets could be induced by fuzzy sets and provide an efficient computing vehicle. Shadowed sets bridge fuzzy sets and rough sets.

4. Fundamental Issues of Traversing Information Pyramid: Encoding and Decoding

Granular computing supports modeling activities carried out at various levels of information granularity, cf. Fig. 3.

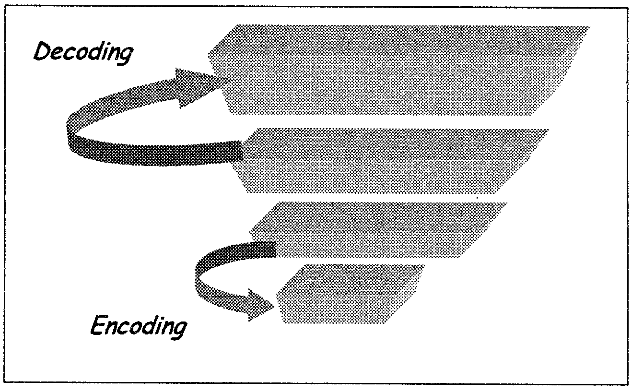


Fig. 3. Decoding and encoding information granules as a vehicle of traversing the information pyramid.

The ability to traverse through the layers characterized by different sizes of information granules is one of the dominant features of the modeling pursuits discussed in this framework. Each modeling layer indexed by the assumed level of granularity, comes with its own repertoire of modeling techniques. For instance, for the highest level of information granularity, viz. numeric data, we are dealing with differential equations and regression models as basic vehicles of system modeling. Commonly used neural networks fall under the same category. When moving towards a non-numeric layer where some information granules of lower granularity are formed, we encounter a diversity of models such as Petri nets, finite state machines, bond graphs, constraint-based, etc. Depending on the specific form of granulation, we subsequently allude to fuzzy Petri nets, probabilistic Petri nets, etc.

The layers communicate between themselves. They receive data from other layers, complete computing (processing) and return the results to some other layers. These communication mechanisms are referred to as encoding and decoding, respectively. The role of the encoder is to transform the input information entering the given layer. The objective of the decoder is to convert the information granules produced by the given layer into the format acceptable by the destination layer. Depending on the problem at hand and the formalism of information granulation being used, a specific naming comes into play.

The general formulation of the encoding-decoding problem can be delineated as follows: Develop encoding (Enc) and associated decoding (Dec) algorithms such that the following relationship is satisfied:

$$\mathrm{Dec}\left(\mathrm{Enc}\,(X)\right) = X$$

for all information granules (X) defined in a certain formal framework of information granulation and for a broad range of sizes of the information granules involved. In a limit case numeric granules are also included. Note, however, that the decoding-encoding scheme could be very demanding and one may not be able to meet the equality. More practically, we request that the design of these transformation should minimize the associated transformation error meaning that we are interested in minimizing the expression involving the distance $\|\cdot\|$ between the original information granule and its transformation

$$\left\|\mathrm{Dec}\left(\mathrm{Enc}\,(X)\right) - X\right\| \to \mathrm{Min}$$

over a given range of granularity of X's involved there and for a fixed granulation environment.

The A/D and D/A conversions form an interesting illustration to the formulation of the problem given above, see Fig. 4. We have:

A/D: $\mathrm{Enc}\,(X)\colon\ X = \{x\} \in \mathbb{R} \to X \in \mathcal{P}(\mathbb{R})$ (the resulting granules are intervals in $\mathbb{R}$; depending on how the intervals are formed, one encounters either a uniform quantization or a non-uniform one).

D/A: $\mathrm{Dec}\,(X)\colon\ X \in \mathcal{P}(\mathbb{R}) \to X = \{x'\} \in \mathbb{R}$ (usually a quantization error occurs so we never obtain the original numeric entity, $x \neq x'$).

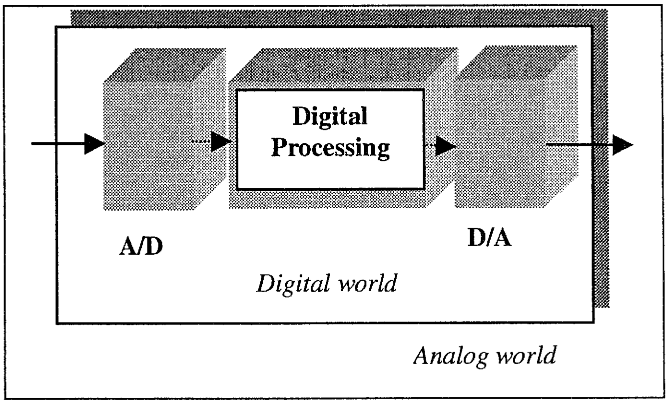


Fig. 4. Digital processing as an example of commonly encountered granular computing; note a role of A/D and D/A converters utilized as the encoding and decoding modules.

The A/D and D/A conversions can be revisited and generalized in the framework of fuzzy sets, $\mathcal{F}(\mathbb{X})$. This leads to the following formulation of the problem:

A/D: $\mathrm{Enc}(X): \quad X = \{x\} \in \mathbb{R} \to X \in \mathcal{F}(\mathbb{R})$ (the resulting granules are fuzzy sets in $\mathbb{R}$; depending on how they are formed, one encounters either a uniform quantization or a non-uniform linguistic discretization of $\mathbb{X}$).

D/A: $\mathrm{Dec}(X): \; X \in \mathcal{P}(\mathbb{R}) \to X = \{x'\} \in \mathbb{R}$ (usually a quantization error; it can be avoided by selecting a proper family of fuzzy sets. The zero error occurs for the triangular fuzzy sets with overlap between successive membership functions).

In fuzzy controllers, the process of converting numeric data into the format accepted by the inference engine is called *fuzzification*. This is the name used for the encoding mechanism. The decoding is referred to as a *defuzzification* scheme.

One may also envisage a mixed form of information granules, namely they may originate from different formal environments of information granulation.

5. Hybrid Models of Information Granules and Interoperability of Various Platforms of Granular Computing

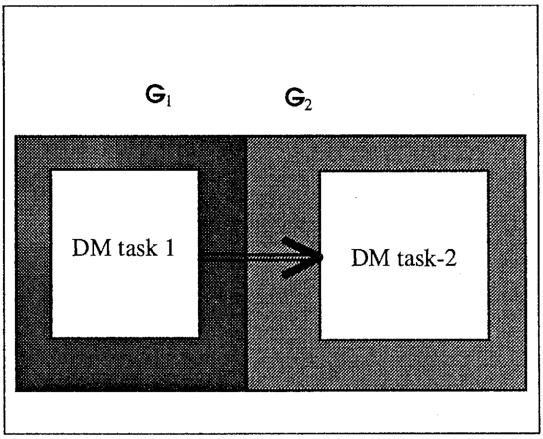
Various models of information granules and granulation processes themselves are crucial in the realization of interoperability when dealing with various platforms of granular computing. We illustrate this important concept in the setting of data mining. Information granules, no matter what formal framework they are supported by, are used as front and back end interfaces of the data mining computing machine. The need for the studies of the hybrid models of information granules arises when we are faced with an issue of interoperability between various tasks or subsystems of data mining that could be realized in various frameworks of granular computing. As an example, consider the situation visualized in Fig. 5(a). One data mining task, say T_1, is realized in the setting of information granules in the setting $\boldsymbol{G}_1$. The other one is developed in the granular environment $\boldsymbol{G}_2$. The results of the first task need to be communicated to the second module. This inherently gives rise to concept of the hybrid models of information granularity. For instance, assume that $\boldsymbol{G}_2$ dwells on set theory. Now if $\boldsymbol{G}_1$ generates the results in the form of fuzzy sets, this type of communication gives rise to fuzzy rough sets. Interestingly, even though $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ could exploit the same formalism of granular information, the communication between these two modules produces rough sets (Pawlak, 1982). This arises as a result of a certain level of granularity of data. As visualized in Fig. 5(b), 'X is A' is a result of passing a message to the second task. This, in turn, invokes the representation of A in terms of the family of sets. As a consequence, even though we have sets at both ends, the representation of A emerges as a rough set. Put it in a different way: rough sets are just the outcome of the communication at the granular level. In more detail, cf. Fig. 5, X is transformed into the following form:

$$X \in \boldsymbol{P}(\mathbb{X}) \Rightarrow X^* \in \boldsymbol{R}(\mathbb{X})$$

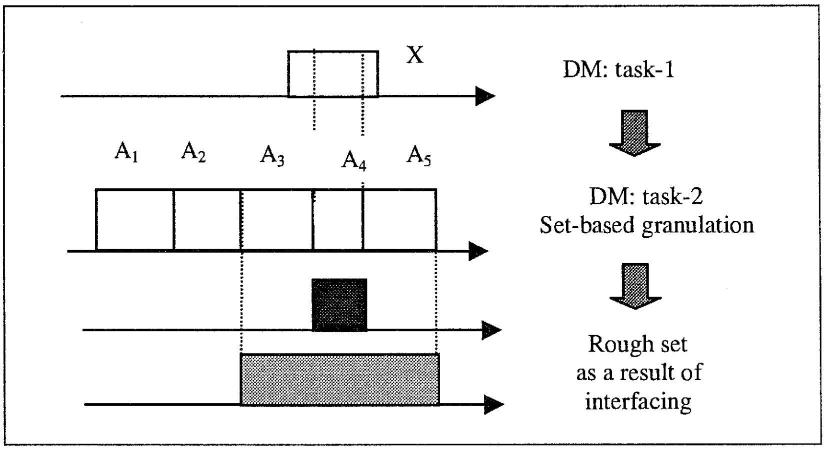
with $\boldsymbol{P}$ and $\boldsymbol{R}$ being the families of sets and rough sets defined in $\mathbb{X}$. The lower and upper bound of X^* is expressed as

$$X_* = \{A_4\}, \quad X^* = \{A_3, A_4, A_5\}.$$

One can justify the origin and usage of some other hybrid models in the same way.



(a)



(b)

Fig. 5. Two data mining tasks realized with the aid of different formalisms of information granulation: (a) a general scheme of communication, and (b) rough sets arising as an effect of communication between DM tasks accomplished in the granular setting implemented by sets.

6. Selected Approaches to Information Granulation

In this section, we elaborate on two selected approaches to information granulation showing how numeric data can be efficiently converted into information granules. The first one is a simple method of a direct transformation of data into chunks of knowledge. The second one alludes to fuzzy clustering.

6.1. Granulation with Fuzzy Sets: From Numeric Data to Triangular Fuzzy Numbers

This approach concentrates on the construction of triangular membership functions based on current experimental numeric data. The granulation of data is carried out according to the following scheme:

- Generate randomly a search region in the entire data space. Both the location of its center and the corresponding spread are generated randomly.

- Collect the respective data and consider each variable in the problem separately. Determine a median value of such a numeric sample (our preference is in the median because of its robustness). The median becomes the modal value of the triangular fuzzy set (fuzzy number). To determine the spread of the information granule, we follow a conservative approach and admit all data to the triangular fuzzy set, see Fig. 6. It could well be that the spreads obtained in this way are too broad, yet this construction helps us incorporate as many data as possible meaning that the resulting information granule conveys a significant experimental justification (obviously, some other variations of this method are possible but will be computationally more intensive).

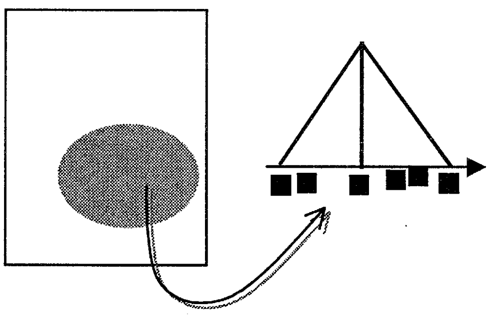


Fig. 6. From numeric data to an information granule described by a triangular fuzzy set (fuzzy number). A shadowed region on the left hand side of the diagram represent a randomly generated search region. The small boxes on the right-hand side illustrate the distribution of the experimental numeric data found therein.

6.2. Clustering Techniques

Clustering is often regarded as a synonym of information granulation. As underlined in the literature, the intent of clustering, no matter how the detailed algorithm looks like, is to find a structure in the data and reveal clusters—information granules in the data set. It is important to emphasize the following features of the clustering method (essentially these observations are pertinent to most of the grouping methods):

- Clustering is a direction-free data analysis; it discovers relations or fuzzy relations not distinguishing between independent and dependent variables (in our study this pertains to potential inputs and outputs of the neural network).

- Clustering is very much driven by the distance function ($\|\cdot\|$) forming the objective function to be optimized. The choice of the distance could be critical to the form of the 'discovered' structure in the data. Interestingly enough, even though clustering concerns unsupervised learning, by imposing the distance function, we predefine a focus of search and specify the geometry of information granules one is looking for. For instance, the Euclidean distance promotes the ellipsoidal-like form of clusters while the Hamming distance focuses the search on hyperboxes in the data space.

Granular neural networks distinguish between input and output variables. Clustering methods that adhere to their relational (direction-free) format could not be the preferred option (as a matter of fact the lack of the already mentioned directionality mechanism may contribute to an overly strong averaging effect). To incorporate the directionality component, we complete clustering based on a series of contexts—fuzzy sets defined in the output space (output variable). Afterwards the clustering is realized for the individual contexts. This limits the scope of clustering by confining it to the respective linguistic granules. These granules are viewed as fuzzy sets defined in the output space. Their choice can be predetermined by the designer of the neural network who can select them having some particular modeling objectives in mind.

The detailed description of the algorithm can be found in the literature (Pedrycz, 1996). It dwells on the well-known FCM method (Bezdek, 1981) and modifies it to the current problem by adding the context-based mechanism. The main computing stages include iterative determination of the prototypes and partition matrix of the clusters.

Let us denote the data to be clustered by $\boldsymbol{x}_k$, $k = 1, 2, \ldots, N$. The number of clusters is equal to c. Moreover, $B(y_k)$ stands for the value of the context B for the k-th data point in the output space, y_k. A few words of explanation are helpful here. The context is a fuzzy set defined by the designer. Say, we are interested in clustering data from the standpoint (context) of medium positive values of the output. Then this term is modeled as a fuzzy set, say with a Gaussian membership function

$$B(y) = \exp\left(-\frac{(y-m)^2}{\sigma^2}\right)$$

(here m and σ are the two parameters of this Gaussian fuzzy set of context). The original data point $(\boldsymbol{x}_k, y_k)$ in the output space invoke the context to the degree $B(y_k)$ thus leaving the pair $(\boldsymbol{x}_k, B(y_k))$ to be used by the clustering algorithm.

The construction of the clusters is guided by the performance index (objective function) that is expressed as

$$Q = \sum_{i=1}^{c} \sum u_{ik}^2 \|\boldsymbol{x}_k - \boldsymbol{v}_i\|^2,$$

where $\|\boldsymbol{x}_k - \boldsymbol{v}_i\|^2$ is a distance function between the data point and the i-th prototype $(\boldsymbol{v}_i)$. Note that, in general, a fuzzification factor (m) assumes any value greater than 1. In this setting the objective function reads as follows:

$$Q = \sum_{i=1}^{c} \sum u_{ik}^m \|\boldsymbol{x}_k - \boldsymbol{v}_i\|^2$$

(in the sequel, we confine ourselves to $m = 2$ as the option being commonly used).

The minimization of this objective function is an example of the constraint-driven minimization

$$\operatorname{Min} Q \text{ subject to } U \in \boldsymbol{U}$$

with $\boldsymbol{U}$ being a family of fuzzy partitions

$$\boldsymbol{U} = \left\{ \boldsymbol{u}_{ik} \middle| \sum_{i=1}^{c} u_{ik} = B(y_k) \text{ and } 0 < \sum_{k=1}^{N} u_{ik} < N \right\}.$$

Not going into details (that could be found elsewhere (Pedrycz, 1996)), we summarize the main algorithmic steps. The algorithm is iterative, starts from a certain partition matrix and iterates through new partition matrices and prototypes until a given stopping criterion is met. The prototypes of the clusters are determined in the form

$$\boldsymbol{v}_i = \frac{\sum_{k=1}^{N} u_{ik}^2 \boldsymbol{x}_k}{\sum_{k=1}^{N} u_{ik}^2}.$$

The partition matrix U arises in the form

$$u_{ik} = \frac{B(y_k)}{\sum_{j=1}^{c} \left(\frac{\|\boldsymbol{x}_k - \boldsymbol{v}_i\|}{\|\boldsymbol{x}_k - \boldsymbol{v}_j\|} \right)^2}.$$

As mentioned, the algorithm is iterative and the computations of the prototypes and the partition matrix are completed up to the point when the differences between the two successive partition matrices are below a certain threshold value.

7. The Design of Granular Neural Networks: General Development Practices

In a nutshell, the development of the granular neural networks involves two main phases, cf. Fig. 7:

- granulation of numeric data; at this level a collection of information granules is formed,
- the construction of the neural network; now any learning that takes place with the neural network is based on the information granules rather than original data.

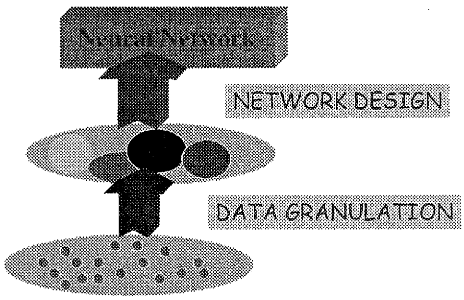


Fig. 7. The development process of granular neural networks perceived as a two-phase design; the construction of the network (including its learning) is followed by data granulation.

The role of the information granules in the development of neural networks has not been dealt with intensively. There has been a lot of research focused on so-called neurofuzzy systems. The name stipulates that information granules implemented in terms of fuzzy sets are used there. This is true to some extent. The point is that information granules do not get involved explicitly in the construction of the neural network. They do not contribute to the condensation of the original learning data. They rather retain the dimensionality of the available data. The proposed modification concerns an introduction of a strongly nonlinear effect through the characteristics of the information granules (say, membership functions). The primary intent is to simplify the learning architecture or/and enhance the learning processes. Being more specific, information granules do contribute to the transformation (deformation) of the original data space and positioning all data there. The classic example of this type of transformation concerns radial basis function (RBF) neural networks (Golden, 1996). One may refer to a number of interesting findings concerning the equivalence between such neurofuzzy networks and classic neural architectures, cf. (Hayashi *et al.*, 1993; Ishibuchi, 1996). The receptive fields are just examples of the membership functions of fuzzy relations. When distributed properly, they make this classification problem linearly separable. Note however, that this transformation phase, even though it relies on the usage of the information granules, does not contribute to any reduction of the size of the training data. Quite often, the new space exhibits a higher dimensionality than the original one (as this is the case for the exclusive-OR problem).

As advocated in Fig. 7, the development process of the granular neural networks involves two fundamental phases:

- First, the data (that are usually numeric) are *condensed* in the form of some information granules and as such are made available to the neural network.
- Second, the resulting information granules are subsequently 'seen' by the neural network and as such used for all training purposes. As a consequence, the neural network is not exposed to the original data of a far higher granularity and far more numerous as the information granules.

Owing to the fundamental role of the information granules, the ensuing neural networks will be referred to as *granular* neural networks. Depending upon the size of the information granules, one can envisage four possible options as outlined in Fig. 8.

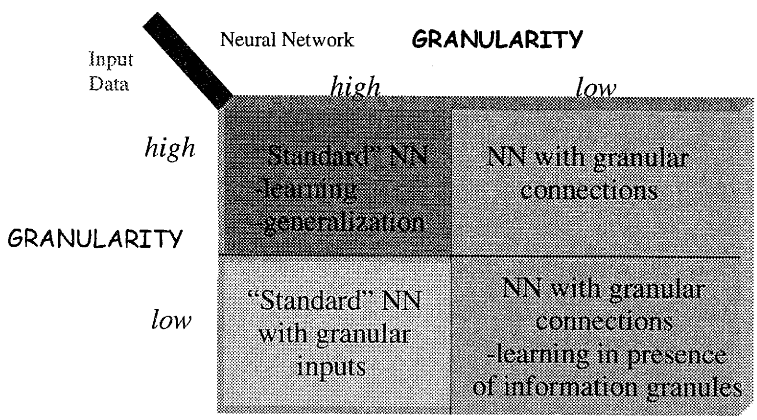


Fig. 8. A general taxonomy of granular neural networks.

This taxonomy sheds light on the key classes of granular neural networks (as a matter of fact, this classification can be easily supported by a number of architectures in the area of neurocomputing). Two criteria dealing with the granularity of the network itself (viz. connections) and the data (those exploited for the training purposes or used afterwards) of which are studied here:

(i) *High* granularity of the structure (connections) along with the *high* granularity of input data. In limit, when we encounter numeric quantities, this entry of the matrix corresponds to the 'standard' numerically-driven neural networks. They come with all training methods available in the literature. The aspects of generalization of the network is placed in the context of numeric generalization meaning that one is interested in the predictive abilities of the network in cases of new numeric entries.

(ii) We may train the network using numeric data (data of *high* granularity) and then use it in the environment where the granularity of data is *low* (the entry indicated as 'a standard neural networks with granular inputs'). Here we have to make sure that the network is capable of accepting nonnumeric data when operating on them. This has to be accomplished through a certain encoding mechanism. In common, this encoding is embodied through a specialized input (preprocessing) layer. The granularity of inputs implies a reduced granularity of the output. The generalization capabilities include also granulation issues, viz. an ability of the network to cope with inputs of varying granularity.

(iii) We may have situations where the network itself has been designed with the use of granular connections (*low* granularity of the underlying neural architecture) whereas the current inputs are of *high* granularity (say, numeric entries). There is a residual effect of granulation: even though the input is numeric, the output of such a neural network may exhibit low granularity (being the reflection of the nonnumeric connections). For these networks, there is no need to develop any specific encoding mechanisms to assure their proper interaction with the environment.

(iv) Finally, the most challenging situation emerges where we are concerned with the structure of the network with the connections of *low* granularity when one has to operate in presence of granular information again of a fairly *low* granularity.

8. Selected Architectures of Granular Neural Networks

The proposed granular neural networks as far as their topologies and respective learning schemes are concerned, are directly linked with the way in which the information granules are constructed. It is worth underlining that quite often the intensity of training depends on the way in which the information granules were developed. Furthermore, the common task embraces a way of encoding the information granules within the topology of the neural network.

8.1. Granular Neural Networks with Parametric Encoding

The underlying architecture is implied by the form of the information granules available for the training of the network. They are regarded to be homogeneous in terms of their representation of the information granules. In other words, we confine ourselves to a certain parametric representation of the information granules. For example, in the case of the triangular fuzzy numbers we represent each input and output variable as a triple of three entries (bounds and modal value) (Bortolan, 1998), see Fig. 9.

The learning of the neural network here has to be done from scratch in the sense we are provided with granular input-output data but there are no prior relationships specified between them. The architecture lends itself to the standard neural network with numeric connections. The learning algorithms here are well-known and fully developed. Note that the use of the granular data leads to the reduction of the size

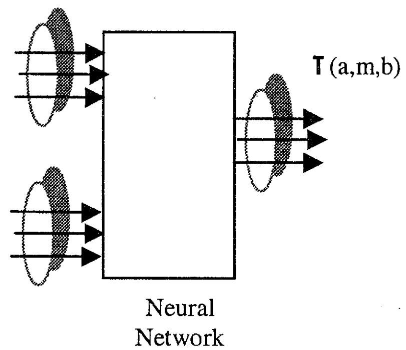


Fig. 9. An architecture of the granular neural network with parametric encoding—the case of triangular fuzzy numbers

of the original numeric data set. At the same time the dimensionality of the problem increases as we need a number of parameters to represent each granule (in the case of the triangular fuzzy numbers this triples the number of inputs).

8.2. Classes of Granular Neural Networks Exploiting Results of Cluster-Based Information Granulation

The context-based clustering leaves us with the number of contexts and induced clusters. The links (associations) between these entities are assumed by the method but not quantified at all. What we are provided with once the contextual clustering has been completed is a structure one can portray in Fig. 10. This figure summarizes the associations as being formed by the clustering method. What has been constructed in this manner, becomes eventually the most descriptive and least restrictive (but yet operational) realization of the granular neural network. Still a lot of details may be missing at this stage.

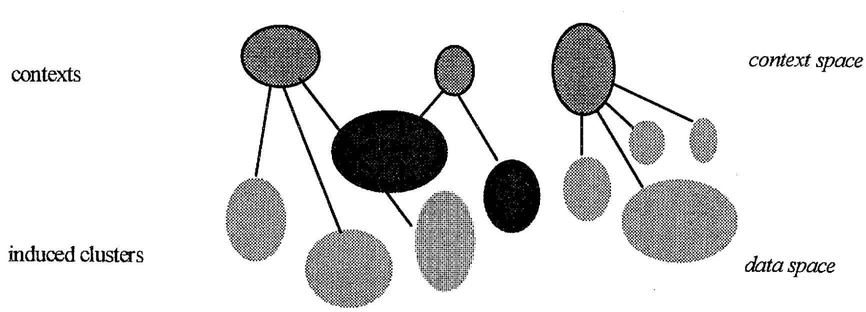


Fig. 10. Results of context-based clustering as a blueprint of the granular neural network.

The skeleton of the granular neural network (that is highly heterogeneous) involves the following computational layers:

- the input layer formed by the induced clusters (from the functional standpoint, they are similar to the radial basis functions used in RBF neural networks),
- the activation levels of the corresponding clusters are combined together in the output layer which is composed of a single summation unit with granular connections being the contexts (fuzzy sets) used in the context-based clustering scheme. The granularity of the connections requires computations involving this effect. We can write them down in a concise manner as

$$Y = \oplus \sum_{i=1}^{p} z_i B_i.$$

Where z_i denotes an activation level of the i-th context—these are the values produced by the previous layer. The addition completed here involves operations on information granules rather than plain numbers. Evidently, the granular neural network produces nonnumeric outputs even for a numeric form of the input.

More specifically, for the contexts being specified as triangular fuzzy numbers, say $B_i = \boldsymbol{T}(a_{i-}, a_i, a_{i+})$, the output of the network is again a triangular fuzzy number, $Y = \boldsymbol{T}(y_-, y_m, y_+)$ with the bounds computed based upon the bounds of the individual contexts. The calculations adhere to the principles of fuzzy arithmetic (Kandel, 1986; Pedrycz, 1998).

Interestingly, there are no provisions as to any further learning of the granular neural network as all its components have already been provided (namely the contexts forming the connections of the neuron in the output layer as well as the form of the receptive fields- induced clusters). To enhance the neural architecture and come up with some adaptable connections, we modify the network by adding one intermediate (hidden) layer dealing with the calibration of the activation levels of the individual clusters, see Fig. 11. Simple summation nodes (each for the corresponding context) are a viable enhancement of the neural architecture.

9. Conclusions

We have discussed the fundamentals of granular computing viewed as a new unified paradigm of processing information granules. Granular computing subsumes commonly encountered numeric processing as its special (limit) case.

The research agenda of granular computing includes a series of key and well-defined methodological and algorithmic issues:

- Construction of information granules. This deals both with the selection of the formal framework of information granulation and detailed estimation procedure producing information granules. The latter dwells on the usage of the setting in which the granules are constructed.

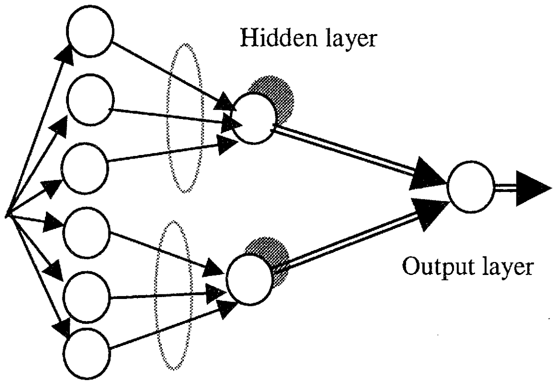


Fig. 11. An enhanced topology of the granular neural network realized by an addition of a hidden layer of linear processing units.

- Characterization of the dimension (granularity) of information granules. This task is crucial as providing us with a better insight as to the essence of the granulation process and its implications both at the level of processing and comprehension.

- The development of the encoding and decoding mechanisms. These are essential to the functioning of any granular architecture. The encoding and decoding schemes are essential to the performance of granular computing. Interestingly enough, the essence of information compatibility expressed in terms of its granularity is inherently related with granular computing and nonexistent within other environments.

- The issues of interoperability are crucial to the design of systems operating within the realm of various formalisms of information granularity.

In this study, we have introduced granular neural networks. These constructs arise at the junction of neurocomputing and neurocomputing and tend to reconcile some of the difficulties encountered in the development of neural networks. The computational aspect of training neural networks in the presence of huge data sets has been effectively addressed by applying some data granulation techniques. The methods of data granulation themselves were discussed as well.

The points worth underlining are the following:

- Granular neural networks can be realized in many possible ways depending on the type of information granules as well as an intensity of the learning processes.

- When looking into the training of the granular neural network, one should be clearly aware of the two dimensions of the training set, namely its *representing* abilities in terms of space coverage as well as the granulation level of the training data themselves. In contrast, when dealing with numeric training data, we have to deal only with the first of these aspects.

- The two-phase development process highlighted in this study, calls for the formation of the information granules and the neural manipulation of these afterwards. This helps reduce all learning burden associated with all large training data sets.
- The granulation mechanism supports an important feature of problem modularization (and as such is very much in line with the existing neural structures such as correlation neural networks (Golden, 1996)). The granular neural network can serve as a skeleton (blueprint) of the overall architecture that could be easily refined afterwards and lead to the detailed construction of the numeric neural networks developed at the local basis.

The intent of this study was to introduce the concept of the granular neural networks and discuss their fundamental properties not necessarily covering all optimization tasks.

Acknowledgment

The support from the Natural Sciences and Engineering Research Council of Canada (NSERC) and Alberta Software Engineering Research Consortium (ASERC) is gratefully acknowledged.

References

Bezdek J.C. (1981): *Pattern Recognition with Fuzzy Objective Function Algorithms.* — New York: Plenum Press.

Bortolan G. (1998): *An architecture of fuzzy neural networks for linguistic processing.* — Fuzzy Sets Syst., Vol.100, No.2, pp.197–215.

Buckley J. and Hayashi Y. (1984): *Fuzzy neural networks: A survey.* — Fuzzy Sets Syst., Vol.66, No.1, pp.1–14.

Golden R.M. (1996): *Mathematical Methods for Neural Network Analysis and Design.* — Cambridge, MA: MIT Press.

Harris C.J., Moore C.G. and Brown M. (1993): *Intelligent Control — Aspects of Fuzzy Logic and Neural Nets.* — Singapore: World Scientific.

Hayashi H., Buckley J.J. and Czogała E. (1993): *Fuzzy neural network with fuzzy signals and weights.* — Int. J. Intell. Syst., Vol.8, No.3, pp.527–537.

Ishibuchi H. (1996): *Development of fuzzy neural networks*, In: Fuzzy Modelling. Paradigms and Practice (W. Pedrycz, Ed.). — Dordrecht: Kluwer Acad. Publ., pp.185–202.

Jang J.S.R. (1993): *ANFIS: Adaptive network based fuzzy inference systems.* — IEEE Trans. Syst. Man Cybern., Vol.23, No.4, pp.665–685.

Jang J.S.R., Sun C.T. and Mizutani E. (1997): *Neuro-Fuzzy and Soft Computing.* — Upper Saddle River, NJ: Prentice Hall.

Kandel A. (1986): *Fuzzy Mathematical Techniques with Applications.* — Reading, MA: Addison-Wesley.

Kasabov N. (1996): *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering.* — Cambridge, MA: MIT Press.

Kruse R., Gebhardt J. and Klawonn F. (1994): *Foundations of Fuzzy Systems.* — Chichester: Wiley.

Pawlak Z. (1982): *Rough sets.* — Int. J. Comp. Sci., Vol.11, No.3, pp.341–356.

Pedrycz W. (1995): *Fuzzy Sets Engineering.* — Boca Raton, FL: CRC Press.

Pedrycz W. (1996): *Conditional Fuzzy C - Means.* — Pattern Recogn. Lett., Vol.17, No.4, pp.625–632.

Pedrycz W. (1997): *Computational Intelligence: An Introduction.* — Boca Raton, FL: CRC Press.

Pedrycz W. and Gomide F. (1998): *An Introduction to Fuzzy Sets: Analysis and Design.* — Cambridge, MA: MIT Press.

Tsoukalas L.H. and Uhrig R.E. (1997): *Fuzzy and Neural Approaches in Engineering.* — New York: J. Wiley.

Zadeh L.A. (1979): *Fuzzy sets and information granularity*, In: Advances in Fuzzy Set Theory and Applications (Gupta M.M., Ragade R.K., Yager R.R., Eds.). — Amsterdam: North Holland, pp.3–18.

Zadeh L.A. (1996): *Fuzzy logic = Computing with words.* — IEEE Trans. Fuzzy Syst., Vol.4, No.2, pp.103–111.

Zadeh L.A. (1997): *Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic.* — Fuzzy Sets Syst., Vol.90, No.2, pp.111–117.