Expert Systems with Applications 41 (2014) 2729–2741

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Perception granular computing in visual haze-free task

Hong Hu, Liang Pang *, Dongping Tian, Zhongzhi Shi

Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Science, Beijing 100080, China

A R T I C L E   I N F O

Keywords:
Granular computing
Leveled granular system
Fuzzy logic
Machine learning
Haze free
Brain-like computer

A B S T R A C T

In the past decade, granular computing (GrC) has been an active topic of research in machine learning and computer vision. However, the granularity division is itself an open and complex problem. Deep learning, at the same time, has been proposed by Geoffrey Hinton, which simulates the hierarchical structure of human brain, processes data from lower level to higher level and gradually composes more and more semantic concepts. The information similarity, proximity and functionality constitute the key points in the original insight of granular computing proposed by Zadeh. Many GrC researches are based on the equivalence relation or the more general tolerance relation, either of which can be described by some distance functions. The information similarity and proximity depended on the samples distribution can be easily described by the fuzzy logic. From this point of view, GrC can be considered as a set of fuzzy logical formulas, which is geometrically defined as a layered framework in a multi-scale granular system. The necessity of such kind multi-scale layered granular system can be supported by the columnar organization of the neocortex. So the granular system proposed in this paper can be viewed as a new explanation of deep learning that simulates the hierarchical structure of human brain. In view of this, a novel learning approach, which combines fuzzy logical designing with machine learning, is proposed in this paper to construct a GrC system to explore a novel direction for deep learning. Unlike those previous works on the theoretical framework of GrC, our granular system is abstracted from brain science and information science, so it can be used to guide the research of image processing and pattern recognition. Finally, we take the task of haze-free as an example to demonstrate that our multi-scale GrC has high ability to increase the texture information entropy and improve the effect of haze-removing.

1. Introduction

Lin (2012) pointed out that Granulation seems to be a natural methodology deeply rooted in human thinking. Many daily things are routinely granulated into sub-things (Lin, 2012). In the IEEE-GrC2006 conference of information about the granular computing (GrC), the outline of GrC is defined as a general computation theory for effectively using granules such as classes, clusters, subsets, groups and intervals to build an efficient computational model for complex applications with huge amounts of data, information and knowledge (Zadeh, 1997). Just as the scholars summarized in the IEEE-GrC2006 conference though the label is relatively recent, the basic notions and principles of GrC, though under different names, have appeared in many related fields, such as information hiding in programming, granularity in artificial intelligence, divide and conquer in theoretical computer science, interval computing, cluster analysis, fuzzy and rough set theories, neutrosophic computing, quotient space theory, belief functions, machine learning, databases and many others (Bargiela & Pedrycz, 2006). The above definition of GrC is too augmental and the subjects about classes, clusters, subsets, groups and intervals have already studied by artificial intelligence and mathematics for a long time. What is really new point for GrC? We think that the new or main point of the GrC lies in the original insight of GrC proposed by Zadeh, in which there are three basic concepts that underlie human cognition: granulation, organization and causation. Informally, granulation involves decomposition of whole into parts; organization involves integration of parts into whole; and causation involves association of causes with effects. Granulation of an object A leads to a collection of granules of A, with a granule being a clump of points (objects) drawn together by indistinguishability, similarity, proximity or functionality (Zadeh, 1997). In this original insight of GrC, Zadel pointed out three important aspects about GrC: (1) the GrC is a main character of human cognition, (2) so called GrC is based on indistinguishability, similarity, proximity or functionality, (3) there is a close relationship among granulation, organization and causation. Based on these points, we think that it is necessary to find some key points of GrC in human cognition. There are two kinds of GrC research: perception-level and knowledge-level. A perception-level GrC does a series feature transformation and tries to find meta-knowledge implied in samples; a knowledge GrC tries to process knowledge or structure

* Corresponding author. Tel.: +86 01062600505.
E-mail addresses: huhong@ict.ac.cn (H. Hu), pangl@ics.ict.ac.cn (L. Pang), tiandp@ics.ict.ac.cn (D. Tian), shizz@ics.ict.ac.cn (Z. Shi).

0957-4174/$ - see front matter
http://dx.doi.org/10.1016/j.eswa.2013.11.006

information based on metal-knowledge. In this paper we focus on the perception-level.

Indistinguishability, similarity and proximity can be described by equivalence relation or tolerance relation, and these relations can be described by some kind distance functions. From the relevant literature, it is easy to see that many GrC researches focus on classification and clustering (Yao, 2000, 2001; Zhang & Zhang, 2003, 2004a). Zhang and Zhang (2003, 2004a, 2004b, 2005) use the quotient space theory to try to study indistinguishability and similarity. Yao (2001) extends the equivalent class to rough approximation set. The quotient space structure described by equivalence relation is used to probe the structure of granules such as classes, clusters, subsets, groups etc. In a more general way, Lin (1998, 2007) and Yao (1998, 1999) use binary relations and neighborhood systems to study indistinguishability and similarity respectively, the geometric concepts: partitions, covering and topology, and neighborhood can be described by binary relations in the algebra. Pedrycz, Hirota, Pedrycz, and Dong (2012) define granular on fuzzy sets and discuss several operations and their granular consistency (Pedrycz et al., 2012). Lin (2012) gives out a summary about the history of granular computing, he discusses all formal description of granular computing and some further directions e.g. GrC, databases and data mining, GrC and clouding computing etc. In fact, GrC should be discussed in the framework of human cognition from perception to pattern recognition and knowledge processing. Although the concept of granular computing has been proposed more than ten years, only few people pay attention to this subject. In fact granular computing pays much more attention to the leveled computing of intelligence. Just as Yao (2006) pointed out: "Granules in the family are called focal elements of discussion at the level. Each level is represented by a plane. While granules at the same level are of similar nature, granules at different levels may be very different. Consequently, we may use different vocabularies and languages for descriptions at different levels." The leveled computation revealed by granular computing is very important for machine learning, e.g. the famous approach deep learning. The term deep learning gained much attraction in the mid-2000s after a publication by Bargiela and Pedrycz (2006), Castro (1995). Nowadays it becomes a huge wave of technology trend for big data and artificial intelligence. Deep learning simulates the hierarchical structure of human brain, processes data from lower level to higher level, and gradually composes more and more semantic concepts. These facts mean that deep learning has a close relationship with the granular computing.

All the granular computing researches aforementioned, in general, neglect information transformation and feature abstraction, which are very important for deep learning. In this paper, we propose a novel framework of granular system, which has ability to process information transformation and object-background separation. We take the haze-free task as an example to validate the ability of our granular system.

These facts mean that the deep learning has a close relationship with the granular computing proposed by us.

The main contributions of this paper include:

(1) The basic notions and principles of GrC termed with different names, but they have appeared in many related fields, such as information hiding in programming, granularity in artificial intelligence, divide and conquer in theoretical computer science, interval computing, cluster analysis, fuzzy and rough set theories, neurotrophic computing, quotient space theory, belief functions, machine learning, databases, and many others, so old version of Granular computing is just an abstraction of old methods, in this paper we give a novel concrete model for granular computing which has a multi scale layered structure from feature abstraction to classification.

(2) Our novel granular system has a close relation with deep learning, so it develops a new focus for deep learning. It is the first time that fuzzy logic is introduced for leveled feature abstraction in deep learning.

(3) Although fussy logic is often mentioned in granular computing, for example,fuzzy logic and rough set technique are used by Lin (1999) for word computing, and Liu, Xiong, and Wu (2012) use fuzzy lattices in the classification based on hyperspherical granular computing (Liu et al., 2012). In this paper, fuzzy logic is not only used for describing granular similar to hyperspherical granular in Liu et al. (2012), but also for feature abstraction and classification. For this purpose, we propose a novel and effective approach which is combined fuzzy logical designing, PSVM and back propagation.

(4) The granular computing proposed by us gives a novel approach for the task of haze-free, the experiments' result show that this approach is sound.

The rest of this paper is organized as follows: In Section 2,we try to give out a formal definition of granular system and granular computing based on the a tolerance relation, which is described by fuzzy logical formula; in Section 3, we discuss the algorithm to design a granular system; in Section 4, we give out a concrete example of designing a granular system for haze-free task; at last Section 5 is the discussion and looking forward to the future.

2. Granular system based on tolerance relation

The difference between a granular system based on equivalence relation and a granular system based on tolerance relation is that an equivalence relation will divide a space into nonoverlapping covering while a tolerance relation will create overlapping covering of this space.

Yiyu Yao proposed granular computing paradigm for concept learning in which two learning strategies are investigated. A global attribute-oriented strategy searches for a good partition of a universe of objects and a local attribute–value-oriented strategy searches for a good covering (Yao & Deng, 2013). In this paper, granular computing is started from feature vectors e.g. images, not attributes. In order to simulation perception procession of our cognition, we define a set of multi-scale nested convex regions with a corresponding computing based on this set. There are two main purposes to build such a granular system based on tolerance relation:

(1) Granular systems are designed to describe similarity and proximity of information, which can be described by tolerance relation. Granular systems based on tolerance relation can be viewed as a topological structure built by topological bases on the topological space (X, τ) induced from a metric space (X, dis) by the metric *dis*. Granular systems based on tolerance relation can be used to describe domain stricture, which represents indistinguishability, similarity and proximity of examples.Classification is determined by the indistinguishability, similarity and proximity of information.
There are two kind similarity among examples-static similarity and dynamical similarity.
   (a) If elements of classes are distributed in standard convex regions, we can use some kind distance function to describe classes distribution domains. In this case, similarity between two objects can be intuitively described by distance functions. If $dis(x, y)$ is a distance function

in the n-dimensional space R^n and c is a point in R^n, the formula $dis(c, y) < r$ described a convex region D in R^n which takes c as its center. Every point y in this region is equal to $y = c + e$, here e is some kind noise, if D is just a ball, e can be viewed as white noise which has an amplitude less than r.We denote such kind similarity as static similarity.

(b) The dynamical similarity is different from static similarity, if one object O_1 continuously changes to another object O_2, e.g. a tadpole continuously grow up to a frog, then O_1 and O_2 are dynamical similar, i.e. if all elements in a class A are dynamical similar, the distribution domain of A is a connected domain. Dynamical similarity will cause distribution domain become very complicate and have a nonlinear borderline. Although,a dynamical similarity may cause the inner class difference larger than the among classes difference, it can also be described by equivalence relation or tolerance relation. The difference of a granular system based equivalence relation and granular system based on tolerance relation is that an equivalence relation will divide a space into non overlapping covering and a tolerance relation will create overlapping covering of this space. The relation described by the formula $dis(x, y) < r$ is the special case of a tolerance relation.

(2) The main purpose of information transformation in pattern recognition is to recognize or classify different objects from their mixture, so information transformation used in pattern recognition should be taken place in a granular system which describes static similarity and dynamical similarity. This is the main point we will discuss in this paper.

Now we try to use fuzzy logical formula based on distance function to define granular systems. There are three distance axioms: (1) $dis(a,a) = 0$; (2) $dis(a,b) = dis(b,a)$; (3) $dis(a,b) + dis(b,c) \geqslant dis(a,c)$. $dis(x,y) < r$ defines a tolerance relation and $dis(x,y) = 0$ defines an equivalent relation, and every tolerance relation can be viewed as an abstract distance which does not obey (d_3), so any granular systems based on equivalent relation and tolerance relation can be described by distance function in a geometrical way.

Definition 1 (*A simple fuzzy logical formula based on distance function*). A simple fuzzy logical formula based on distance function $sp(\alpha, c|dis, d, \omega)$ is denoted as:

$$sp(a, c|dis, d, \omega) = m(dis(a, c|\omega), d) \tag{1}$$

where $dis(a, c|\omega) = dis(a \circ \omega, c \circ \omega)$, $x \circ \omega = (x_1\omega_1, \ldots, x_n\omega_n)$, and $dis()$ is point to point distance function, and ω is a weight vector for dimensions of feature vector space $S, m()$ can be viewed as a membership function of a fuzzy set, it is usually a continuous function. For example

$$m(dis(\alpha, c|\omega), d) = \begin{cases} (d - dis(\alpha, c|\omega))/d, & dis(\alpha, c|\omega) < d \\ 0, & dis(\alpha, c|\omega) \geqslant d \end{cases} \tag{2}$$

where the distance function can be a set to set distance function, e.g. Hausdorff function.

For the point to point distance function case, we define the r-cut set of $sp(\alpha, c|dis, d, \omega)$ as.

$sp^r(\alpha, c|dis, d, \omega)sp(\alpha, c|dis, d, \omega) > r$ (strong r-cut set) or $\overline{sp^r}(\alpha, c|dis, d, \omega) = sp(\alpha, c|dis, d, \omega) \geqslant r$ (r-cut set).

$sp^r(x, c|dis, d, \omega)$ defines an open convex region and denoted as a granule, and $\overline{sp^r}(x, c|dis, d, \omega)$ defines a closed convex region.

Definition 2 (*leveled perception granular system based on tolerance relation Gsys*). A granular system based on tolerance relations of distance function (granular system for short) is a set of granules, every granule is a 2-tuple $\{g, S_F\}$, here g is a convex region which is described by tolerance relations of distance function, and S_F is a set of fuzzy logical functions (denoted as "adjoint functions") which are computed from the convex region g, the outputs of all fuzzy logical functions in S_F are denoted as "an adjoint vector" of this granule. The granules of Gsys have the following attributions.

(1) Multi-scale leveled structure: The metric space X (e.g. a finite connected volume region in the n-dimensional real space R^n) is the only level 0 granule, the level 0 granule is denoted as $G(coeG^0) = \{X, S_F\}$, where $coeG^0$ is a coefficient set, $coeG^0$ is usually empty, and the function S_F is a set of fuzzy logical functions (denoted as adjoint functions). The convex region of a level 1 granule $G(coeG^1) = \{g^1, S_{F1}\}$ is defined by the conjunction of finite number r-cut sets $sp^r\left(\alpha, c_1^1|dis^1, d_1^1, \omega\right), sp^r\left(\alpha, c_2^1|dis^1, d_2^1, \omega\right), \ldots, sp^r\left(\alpha, c_{k_1}^1|dis^1, d_{k_1}^1, \omega\right)$, where $coeG^1$ is its coefficient set to define the convex region g^1, so g^1 can also be written as $g^1(coeG^1)$ and $coeG^1 = \left\{c_1^1, c_2^1, \ldots, c_{k_1}^1 \in X, d_1^1, d_2^1, \ldots, d_{k_1}^1, dis^1\right\}$. The first level of our granular system is denoted as $C^1(X) = \{G(coeG^1)\}$. If the l level granules $G(coeG^l) = \{g^l, S_{Fl}\}$ have been defined, the $l+1$ level granules can be defined as $G(coeG^{l+1})$, which can be defined as the convex region created by the intersection of g^l and finite number strong r-cut sets: $sp^r\left(\alpha, c_1^{l+1}|dis^{l+1}, d_1^{l+1}, \omega\right)$, $sp^r\left(\alpha, c_2^{l+1}|dis^{l+1}, d_2^{l+1}, \omega\right), \cdots, sp^r\left(\alpha, c_{k_{l+1}}^{l+1}|dis^{l+1}, d_{k_{l+1}}^{l+1}, \omega\right)$. A level $l+1$ granule's coefficient is $coeG^{l+1} = \{c_1^{l+1}, c_2^{l+1}, \ldots, c_{k_{l+1}}^{l+1} \in X, d_1^{l+1}, d_2^{l+1}, \ldots d_{k_{l+1}}^{l+1}, dis^{l+1}\}$, where k_{l+1} is denoted as the number of simple logical formulas in same level. The $centerofG(coeG^{l+1})$ is $c_i^{l+1} \in G(coeG^l)$, and its radius is $d_i^{l+1} \leqslant d_i^l, \lim l \to \infty d_i^l = 0$. The $G(coeG^1)$ is called as the "the father granule of $G(coeG^{l+1})$ ".

(2) Granular computing (GrC): A granular computing is described by a set of fuzzy logical formula upon above multi scale leveled structure of convex regions.
The purpose of a granular computing (GrC) is to transfer feature information and classy points in an input space X, so at least one level of a granular computing (GrC) outputs a fuzzy label for points in an input space X. If there are totally m classes, a fuzzy label L is a m dimensional fuzzy vector $L = \{l_1, l_2, \ldots, l_m\}, and \sum_{i=1,\ldots,m} l_i = 1$.
Castro (1995) proved that Fuzzy logic controllers using fuzzy rules are universal approximations, later, Li and Philip Chen (2000) shows a proof of the equality between a forward neural circuit (or circuit) and a fuzzy logical inference. So it is not difficulty to prove that any continuous functions $F : R^n \longrightarrow [0, 1]^n$ can be simulated by such kind nested layered granular computing with arbitrary small errors.
A level $l+1$ adjoint function F_{l+1} receives its input from the outputs of level $l+p, p > 1$ adjoint functions, i.e. if a leveled granular system Gsys has k levels, GrC is taken from level k to 1, so the level of a GrC is upside down with the level of the Gsys. The 1st level GrC takes place in the smallest kth level granules' convex regions of the Gsys. Two kinds layered computing can be taken place over a granular system. In the first kind layered computing,the adjoint feature vectors of larger scale level n granules are computed based on the

adjoint feature vectors of smaller scale $n+1$ level granules,such kind layered computing has strictly nested structure, (Fig. 1(a)), and is denoted as "nested layered computing". In the second kind layered computing, adjoint feature vectors of level n granules can be computed based on all adjoint feature vectors of smaller granules, which have level greater than n (Fig. 1(b)), such kind layered computing is denoted as "unnested layered computing". Nested layered computing is a special case of unnested layered computing.

(3) Radiuses of convex regions: A granular system can have countable infinite or finite levels. The radiuses of granules'convex regions decrease and tend to zero when the level goes to infinite.

(4) Centers' grid: The centers of granules will distribute in a so called center grid, we call the set of all centers of granules of level $l+1$ on the granule $G(coeG^l)$ as the center grid of level l granule $G(coeG^l)$ denoted as $Gc^{l+1}(G(coeG^l))$. We denote the set of all centers of level $l+1$ granules over X as $Gc^{l+1}(X)$ and all centers of level $l+1$ granules over a level $k<l$ granule $G(coeG^k)$ as $Gc^{l+1}(G(coeG^k))$.
The center grid is usually discrete, but it can also be a continuous set e.g. the whole metric space X.

(5) Shape of granules: the shapes of granules is defined by their distance functions $dis()$. If $dis()$ can be an abstract distance function, then a granule's convex region can be an arbitrary convex region. Every level uses same distance function, so the granules in the same level have the same shape, but for different levels, granules'region may have different shapes.

(6) The cover over a granule: In order to create a cover over $G(coeG^l)$, the elements in the centers' grid Gc^{l+1} $(G(coeG^l))$ should be tight enough. Such kind cover can be formerly defined as:

$$C^{l+1}(G(coeG^l)) = \left\{ \bigwedge_{1\leqslant i\leqslant k_{l+1}} sp^r(x, c_i^{l+1}|dis^{l+1}, d_i^{l+1}, \omega) \wedge g(coeG^l) | c_i^{l+1} \in Gc^{l+1}(G(coeG^l)), sp^r(x, c_i^{l+1}|dis_{l+1}, d_i^{l+1}, \omega) \wedge g(coeG^l) \neq \phi \right\}$$

All level $l+1$ granules create a cover of the whole space X, and denoted as the level $l+1$ cover or the level $l+1$ layer of X, $C^{l+1}(X) = \bigcup_{G(coeG^l)\in C^l(X)} \{C^{l+1}(G(coeG^l))\}$.

Radial Basis neural network (Haykin, 2008), which can be used to simulate continuous functions, is an example of two layers granule system.

Definition 3 (*Hyper-granules and mini-granules*). All level $n+1$ granules $G(coeG^{n+1})$, which are contained in a level n granule $G(coeG^n)$, denoted as "mini-granules", and the level n granule $G(coeG^n)$ is denoted as a "hyper-granule".

After the theory of fuzzy logic was conceived by Zadeh (1965), many fuzzy logical systems have been presented, for example, the Zadeh system, the probability system, the algebraic system, and Bounded operator system, etc. According to universal approximation theorem (Haykin, 1994), in this paper, the extended Bounded operator is selected, which is denoted as "q-value Weighted Bounded operator". It is not difficult to prove that q-value weighted fuzzy logical formulas can precisely simulate any continuous functions $F : R^n \longrightarrow [0,1]^n$ with arbitrary small error, or vice versa, i.e. every GrC can be completed by a set of fuzzy logical functions of q-value weighted bounded operator with arbitrary small error.

Definition 4 (*Bounded Operator* $F(\oplus_f, \otimes_f)$). Bounded product: $x \otimes_f y = \max(0, x+y-1)$, and Bounded sum: $x \oplus_f y = \min(1, x+y)$, where $0 \leqslant x, y \leqslant 1$.

In order to simulate GrC, it is necessary to extend the Bounded Operator to Weighted Bounded Operator. The fuzzy formulas defined by q-value weighted bounded operators is denoted as q-value weighted fuzzy logical functions.

Definition 5 (*q-value Weighted Bounded operator* $F(\oplus_f, \otimes_f)$). q-value Weighted Bounded product:

$$\begin{aligned} p_1 \otimes_f p_2 &= F_{\otimes_f}(p_1, p_2, w_1, w_2) \\ &= \max(0, w_1 p_1 + w_2 p_2 - (w_1 + w_2 - 1)q) \end{aligned} \tag{3}$$

q-value Weighted Bounded sum:

$$p_1 \oplus_f p_2 = F_{\oplus_f}(p_1, q_2, w_1, w_2) = \min(q, w_1 p_1 + w_2 p_2) \tag{4}$$

where $0 \leqslant p_1, p_2 \leqslant q$.

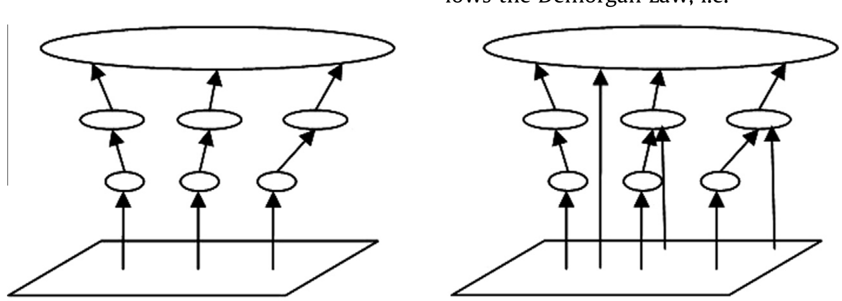
For association and distribution rules, we define:

$(p_1 \Delta_f p_2) \Theta_f p_3 = F_{\Theta_f}(F_{\Delta_f}(p_1, p_2, w_1, w_2), p_3, 1, w_3)$ and $p_1 \Delta_f (p_2 \Theta_f p_3) = F_{\Delta_f}(p_1, F_{\Theta_f}(p_2, p_3, w_2, w_3), w_1, 1)$, Here $\Delta_f, \Theta_f = \otimes_f$ or $\oplus_f$. We can prove that $\oplus_f$ and $\otimes_f$ follow the associative condition (see Appendix C) and

$$x_1 \oplus_f x_2 \oplus_f x_3 \ldots \oplus_f x_n = \min\left(q, \sum_{1\leqslant i\leqslant n} w_i x_i\right) \tag{5}$$

$$x_1 \otimes_f x_2 \otimes_f x_3 \ldots \otimes_f x_n = \max\left(0, \sum_{1\leqslant i\leqslant n} w_i x_i - \left(\sum_{1\leqslant i\leqslant n} w_i - 1\right) q\right) \tag{6}$$

For more above q -value weighted bounded operator $F(\oplus_f, \otimes_f)$ follows the Demorgan Law, i.e.



(a) nested layered computing (b) unnested layered computing

Fig. 1. Two kind layered computing over granular system.

$$
\begin{aligned}
N(x_1 \oplus_f x_2 \oplus_f x_3 \ldots \oplus_f x_n) &= q - \min\left(q, \sum_{1 \leqslant i \leqslant n} w_i x_i\right) \\
&= \max\left(0, q - \sum_{1 \leqslant i \leqslant n} w_i x_i\right) \\
&= \max\left(0, \sum_{1 \leqslant i \leqslant n} w_i (q - x_i) - (\sum_{1 \leqslant i \leqslant n} w_i - 1) q\right) \\
&= N(x_1) \otimes_f N(x_2) \otimes_f N(x_3) \cdots \otimes_f N(x_n).
\end{aligned} \tag{7}
$$

But for the q-value weighted bounded operator $F(\oplus_f, \otimes_f)$, the distribution condition is usually not hold, and the boundary condition is hold only all weights equal to 1, for $p_1 \otimes_f q = F_{\otimes_f}(p_1, q, w_1, w_2) = \max(0, w_1 p_1 + (1 - w_1) q)$ and $p_1 \oplus_f q = F_{\oplus_f}(p_1, q, w_1, w_2) = \min(q, w_1 p_1 + w_2 q)$. In this paper, we show that the task of haze-free can be completed by a common GrC based on fuzzy logical formulas of bounded fuzzy operator.

3. Hybrid designing of leveled perception granular system based on fuzzy logic and PSVM

Owing to the limitation of the scope, in this paper only nested layered GrC is discussed. A nested layered GrC is defined by the input and output relation of a granular computing on a granular system. There are three kinds relations between nearby layers (layers k and $k+1$) of a nested GrC: (1) binary logic; (2) fuzzy logic; (3) alogical relation.

Because fuzzy logic and binary logic are all created by the sigmoid function, so back propagation method can be used to modify weights of all layers. In order to speed up the learning process, for a layered GrC, we combine logical designing with PSVM (Fung & Mangasarian, 2001), such kind novel approach is called as "Logical support vector machine (LPSVM)". For nested layered GrC, parameters in the binary logical layers can be directly designated according to the binary relation; for the fuzzy logical layers, parameters can also be set according to these layers'functions, but a suitable small adjustment by back propagation is necessary, this is similar to the deep learning proposed by Geoffrey Hinton such that a many-layered neural network could be effectively pre-trained one layer at a time, treating each layer in turn as an unsupervised restricted Boltzmann machine,then using supervised backpropagation for fine-tuning. For the non logical (alogical) layer, parameters should be learned based on samples according to the input and output relation function $f_i(x_1, x_2, x_3, \ldots, x_n)$, we can use Back Propagation method or PSVM, to learn weights for $f_i(x_1, x_2, x_3, \ldots, x_n)$.

The designing strategy of LPSVM:

- Step 1: Except for the alogical layer's weights, designing the layers' weights according to the logical (binary or fuzzy) relations, for fuzzy logical relations, a suitable modification of weights maybe be necessary according to the task of this layer;
- Step 2: Alogical layers' weights are computed for the input layer to the last output layer. For an alogical layer i, if X is the input train set, computing the inner layers' output from the 1st layer to the $(i-1)$th layer based on X;
- Step 3: Using PSVM to compute the ith layer's weights W_i according to (8);
- Step 4: Repeat the Step 2 to Step 4, until the output error is small enough.
- Step 5: using back propagating approach to modify all layers weights. The weight vector W_l of

$$W_i = \mathbb{X}' \mathbb{D} \mathbb{U} \tag{8}$$

Where The weight vector W_i of the node, $\mathbb{U}$ is computed by (9) and $\mathbb{X}$ and $\mathbb{D}$ are the problem data, i.e. $\mathbb{X} = [X_1, \ldots, X_n]$, and diagonal matrix $\mathbb{D} = \begin{bmatrix} y_1 & 0 & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & y_n \end{bmatrix}$, (X_i, y_i) is a training sample with X_i feature vector and target y_i .

$$\mathbb{U} = \left(\frac{\mathbb{I}}{v} + \mathbb{D}(\mathbb{X}\mathbb{X}' + EE')\mathbb{D}\right)^{-1} E \tag{9}$$

Where v is a positive parameter selected for guarantee of a small magnitude $\|W_i\|$, $\mathbb{I}$ is the identity matrix, and E is a vector with all elements are 1.

4. Granular system for visual task

The columnar organization of our brain's primary visual cortex strongly supports the granular system defined aforementioned. Many functions of the primary visual cortex are still unknown, but the columnar organization is well understood. The lateral geniculate nucleus (LGN) transfers information from eyes to brain stem and primary visual cortex (V1) (Mountcastle, 1997). Columnar organization of V1 plays an important role in the processing of visual information.

Local similarity of information processing gives rise to Columnar organization has a granular structure. V1 is composed of a grid of $(1 \times 1mm^2)$ neural area of hypercolumns (hc) in our brain's primary visual cortex. Every hypercolumn contains a set of minicolumns (mc), which have same focus. Each hypercolumn analyzes information from one small region (described by a distance function) of the retina. Adjacent hypercolumns analyze information from adjacent areas of the retina, so the structure of a columnar organization can be described by a set of fuzzy logical formulas similar to a granular system. Hypercolumns (or supercolumns), minicolumns (mc) can be viewed as granules. Similar to the primary visual cortex,in our granular system, there are two kind granules:hyper-granule and mini-granules in some levels of our granular system. A hyper-granule contain a bundle of mini-granules.

Definition 6 (*Perception Granular system of Columnar Organization (COGsys)*). A perception columnar organization is a special perception granular system, in which, there is at least one hyper-granule $G(coeG^{n+1})$ such that all mini-granules included in it have same convex region, but different adjoint functions.

In this paper, in order to simulating visual cortex, a granular system of columnar organization (COGsys) is designed for the haze-free task.In our Hybrid designing approach (LPSVM), we firstly design Leveled Granular Systems with the help of fuzzy logic, and then we use PSVM to accomplish the learning for some concrete visual tasks.

4.1. The theory of image matting

According to Levin, Lischinski, and Weiss (2008), image matting refers to the problem of softly extracting the foreground object from an input image and a trimap image. "Tripmap" means three kinds of regions, white denotes definite foreground region, black denotes definite background region and gray denotes undefined region. Formally, image matting methods take I as an input, which is assumed to be a composite of a foreground image F_o and a background B in a linear form and can be written as $I = \alpha F_o + (1 - \alpha) B$. For the haze-free task, the fuzzy label of haze or non-haze is described by the parameter α . And the task of image matting tries to find a function $F_o = f_{F_o}(I)$. Closed form solution assumes that α is a linear function of the input image I in a small window

$w : \alpha_i \approx aI_i + b, \forall i \in w$. Then to solve a spare linear system to get the alpha matte. Our GrC approach gets rid of the linear assumption between α and I. Instead, we try to introduce nonlinear relation between α and I:

$$\alpha_w = F(W_{I_w}) \tag{10}$$

here W_{I_w} is the image block included in the small window w, and α_w is its center pixel's fuzzy label. We take color or texture in local window as our input feature, and the trimap image as the target. After training, the neural fuzzy logical network will generate the result of alpha matte. In the application of alpha matting, our method can remove the haze using dark channel prior as the trimap.

4.2. Leveled perception granular system for haze-free task

In this section, we try to design a Perception Granular system of Columnar Organization (COGsys) for the haze-free task, here only nested layered GrC is needed. The recognition of our Leveled Granular System (see Fig. 2) is started with the recognition orientation or simple structure of local patterns, then the trimap image is computed based on these local patterns. Eq. (11), which has a high ability to simulate fuzzy logic operator (see the detail in the appendix) is used to design GrC. The weight w_i in Eq. (11) can be viewed as connections among granules. A nested layered GrC is defined by the input and output relation of a granular computing on a granular system. just as above mentioned, there are three ways to design weights of a layered GrC:according to the binary or fuzzy logical relation about this layered GrC and according to the input and output relation function $f_i(x_1, x_2, x_3, \ldots, x_n)$ from training samples.

$$\begin{aligned} U_{l+1,i} &= \sum_k w_{l+1,i,k} \cdot I_{l+1,k,i} \\ O_{l+1,i} &= sigm(U_{l+1,i}, T_{l+1,i}, \lambda) \end{aligned} \tag{11}$$

where $sigm()$ is a sigmoid function Eq. (12), and $O_{l+1,i}$ is the output of a level $l+1$ granule.

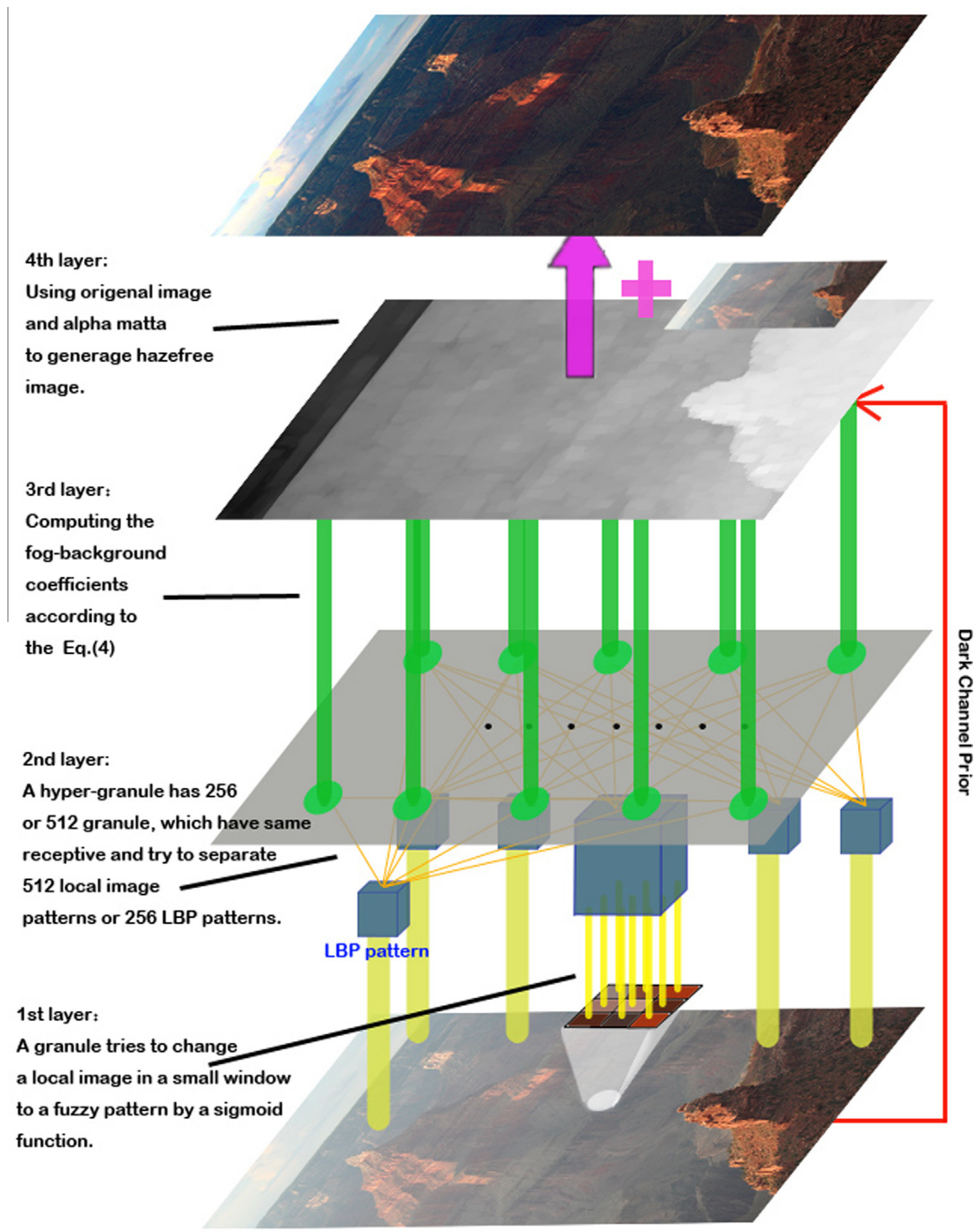


Fig. 2. A 4 layers' structure of a granular system for haze-background separation.

$$sigm(x,t,\lambda) = 1/\{1+exp\{-\lambda\cdot(x-t)\}\} \quad (12)$$

The Theorem 1 discussed in the Appendix A guarantee that above defined granule computing can simulate a boolean function with arbitrary small error. As the designing of Gsys contains two parts: (1) convex regions, (2) adjoint fuzzy logical functions for GrC. The following Gsys for haze-free task is a very simple convex regions can be described by distance function.

The input space is just an image, which is a 5-dimensional space $X = \{(x,y,r,g,b)\}$, here every example (x,y,r,g,b) represents a pixel of this image, (x,y) is the pixel's location and (r,g,b) is pixel's color value. the nested granular system is build on the image. A granular system is built upon images, with fuzzy logical formula $sp(\alpha,c|dis,d,\omega)$ here $\omega = (1,1,0,0,0)$, and $d = 0$ for level 1 GrC, and $d > 1$ for higher level *GrC*s. All levels' centers are located on the whole image plane, so every centers grid is just the image plane and granules are overlapped.

In the following pages, we focus on the designing of adjoint fuzzy logical functions for GrC.

If there are k levels in our Gsys, the kth level receives the input image I, and the first level granule outputs the result haze free image. The relation between input and output of a level-l granule is described by Eq. (11). The weights among granules can be designed by LPSVM, the weights of 1st and 2nd layers are designed by fuzzy logic, and the weights of the 3rd layer are designed by PSVM to learn the trimap image.

For the sake of simplicity, in the following Gsys, we use the order of GrC level which is upside down with the granular system level, and one layer may contain two GrC levels.

The GrC of COGsys is formally defined as bellow:

(1) The 1st layer – fuzzy logical layer Every hyper-granule (Fig. 3) in the 1st layer tries to change a 3×3 pixels' image block $I_{b3\times3}$ into a binary 3×3 pixels'texture pattern. The input image is normalized. A hyper-granule $HG = (g, S_F)$ in the 1st layer contains 3×3 mini-granules to focus a 3×3 small window, every mini-granule focuses only one pixel, so the convex region of a hyper-granule is described by $dis(x,c) \le 0$. A hyper-granule completes the task of image processing. There are three kinds fuzzy logical functions in a hyper-granule's S_F :

(1) In a local image pattern recognition way (LIPW): the 1st processing directly transforms every pixel's value to a fuzzy logical one by a sigmoid function.

$$F_1(\{I_b\}_{3\times3}) = \{sigm(p_{i,j})\}_{3\times3} \quad (13)$$

here $p_{i,j}, i,j = 1,2,3$ is the RGB pixel value in a small 3×3 window;

(2) In a local Binary Pattern operator simulating way (LBPW). The 2nd processing is also completed by a sigmoid function; the difference is that every boundary pixel's value is fuzzy exclusive OR $\oplus$ with the center pixel's value before sending it to a sigmoid function,

$$F_2(\{I_b\}_{3\times3}) = \{f(p_{i,j})\}_{3\times3} \quad (14)$$

Here $f(p_{i,j}) = sigm(p_{i,j}\oplus p_{2,2})$ when $i,j \ne 2$, and $f(p_{2,2}) = 0$. F_2 is similar to a Local Binary Pattern operator (LBP) mentioned in Ojala, Pietikäinen, and Harwood (1996) as a mean of summarizing local gray-level structure. The operator takes a local neighborhood around each pixel, thresholds the pixels of the neighborhood at the value of the central pixel and uses the resulting binary-valued image patch as a local image descriptor. It was originally defined for neighborhoods, giving 8 bit codes based on the 8 pixels around the central one. Such processing emphasizes the contrast of texture, and our experiments support this fact.

(3) Hybrid LIPW and LBPW (LBIPW). The adjoint function $F_3(\cdot)$ in LBIPW is same as $F_2(\cdot)$ in LBPW, except that $f(p_{2,2}) = p_{2,2}$ in $F_3(\cdot)$, while $f(p_{2,2}) = 0$ in $F_2(\cdot)$.

Every granule in a 1st layer's granule has only one input weight w_{ij} in Fig. 3, which equals 1; when $\lambda \to +\infty$, the coefficient λ in Eq. (11) changes the outputs from fuzzy values to binary numbers.

(2) The 2nd layer–binary logical layer Every 2nd layer mini-granules try to recognize a definite shape (see Fig. 4), so they share the same convex region with a 1st layer hyper-granule, which focuses on the same small 3×3 window in an image, and can be described by $dis(x,c) < 2$. If there are total q local small patterns, a hyper-granule in the 2nd layer contains q (in our system $q = 256$ or 512) mini-granules of the 2nd layer, which have same receptive field, but with a different adjoint fuzzy logical function, which tries to recognize a definite shape from the output of a 1st layer hyper-granule. For example, the "∩" shape in Fig. 4 can be described by a adjoint fuzzy logical formula (Eq. (15)). The "and" operator for 9 inputs in Eq. (15)can be created by a granule mc (see. Fig. 4). In Eq. (15), every pixel P_{ij} has two states m_{ij} and $\overline{m_{ij}}$. Suppose the unified gray value (or RGB value) of P_{ij} is g_{ij}, and an image module needs a high value g_{ij} at the place of m_{ij} and a low value at $\overline{m_{ij}}$. So the input for the granule mc at m_{ij} is $I_{ij} = g_{ij}$, and at $\overline{m_{ij}}$ is $I_{ij} = -(1.0-g_{ij})$. A not gate mc' is needed for $I_{ij} = -(1.0-g_{ij})$, here $g_{ij}, i,j = 1,2,3$ is the output of a 1_{st} layer hyper-granule.

$$\mathrm{P} = m_{11}\wedge m_{12}\wedge m_{13}\wedge m_{21}\wedge m_{23}\wedge m_{31}\wedge m_{33}\wedge \overline{m}_{22}\wedge \overline{m}_{32} \quad (15)$$

$$w_{ij} = \begin{cases} 1, & \text{if the } j\text{th bit of a binary pattern} = 1 \\ -1, & \text{if the } j\text{th bit of a binary pattern} = 0 \end{cases} \quad (16)$$

where for LIPW and LBIPW, $j = 1,2,3,\ldots,9$; for LBPW, the center 1st-layer granule is useless, so $j = 1,2,3,\ldots 8$. There are three kinds hyper-granules in the 2nd-layer, which receive three different outputs of a 1st-layer's hyper-granule, so a hyper-granule in the 2nd-layer may work in one of following three ways:

1. In the local image pattern recognition way (LIPW): every 2nd layer hyper-granule contains 512 2nd-layer's mini-granules, and inputs of these 2nd-layer's mini-granules come from a 1st-layer's hyper-granule which works in LIPW way. Every 2nd-layer's hyper-granule tries to classify the image block in this window into 512 binary texture patterns (BTP), e.g. eight important BTPs are shown in Fig. 5. The pixel value is "1" for

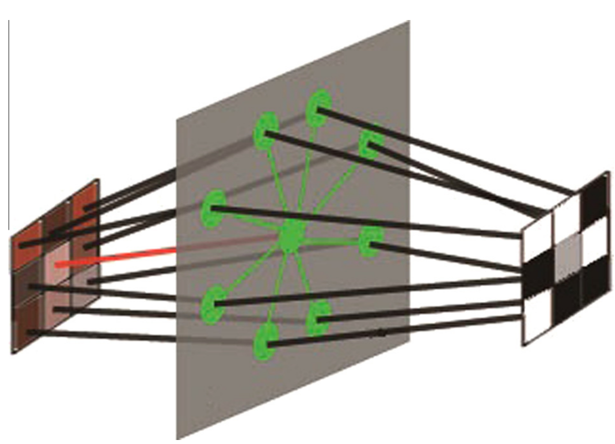


Fig. 3. Every 1st layer granule tries to change a local image into a binary texture pattern. For a hyper granule is defined by a distance function $dis(x,c) < 3$, which is just a 3×3 small window, a hyper-granule in the 1st layer contains 9 granules, and every granule focuses only one pixel.

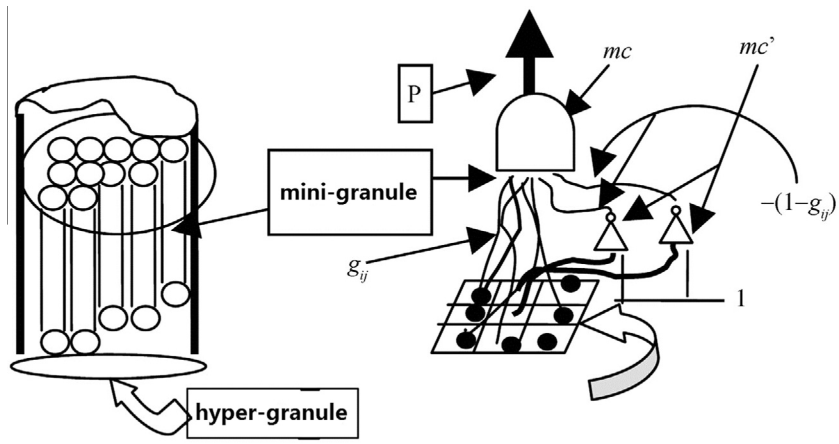


Fig. 4. A hyper-granule in the 2nd layer contains q granules which have same receptive field and try to recognize q definite small shapes. A '*and*' granule is needed for every 2nd layer granule.

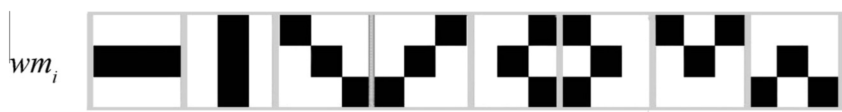


Fig. 5. Every the 2nd layer's granule contains 256 or 512 granule which corresponds to 256 or 512 modules in above picture.

white and "0" for black. In this mode, 3×3 granules of the 1st layer output a 3×3 vector, i.e., a 3×3 fuzzy logical pattern of a BTP, which is computed by a sigmoid function.

2. In the local Binary Pattern operator simulating way (LBPW), a 2nd-layer's hyper-granule contains 256 2nd-layer's mini-granules which receive input from the output of a 1st-layer's hyper-granule, which works in the way of LBPW.
3. In the hybrid LIPW and LBPW (LBIPW) way, a 2nd-layer's hyper-granule contains 512 2nd-layer's mini-granules which receive input from the output of a 1st-layer's hyper-granule, which has 9-dimensions.

In our system, a Gsys is built for every color channel R,G or B, so a hyper-granule in the 2nd layer has a 512×3 dimensions output or 256×3 dimensions output.

As a binary logical layer, in order to recognize a binary pattern, an 'and' granule with index i is needed (see Fig. 4) for every 2nd-layer granule, and the weights of this 'and' granule to the 1st-layer granules are set as Eq. (16), the corresponding parameters in Eq. (11) are set as the threshold $T_i = 5.1$, and $\lambda = 0.9$.

4.2.1. The 3rd layer – alogical layer

The convex region of this layer can also be described by $dis(x, c) < 2$. The output of a hyper-granule in the 2nd layer, which has 3×256 or 3×512 dimensions, is transformed to the 3rd-layer granules to compute the similarity parameter or fuzzy value α_i in Eq. (10), the weights of this layer is computed by psvm, the target is provided by so called dark channel prior which is computed by the approach mentioned in He, Sun, and Tang (2011). As all α_i are optimised on the whole image, in this layer,the whole image is the only convex region. As the small windows focused by hyper-granules in the 2nd-layer are overlapped, the focuses of 3rd-layer's granules are also overlapped.

4.2.2. The 4th layer – fuzzy logical layer

In this layer, a granule tries to remove the haze from original image. A granule in the 4th layer computes a pixel of a haze free image according to fuzzy logical equation Eq. (17)

$$I_i(x) = \min\{q, \alpha_i(x) \cdot J_i(x) + (1 - \alpha_i(x)) \cdot A_i\} = J_i(x) \oplus_f A_i \quad (17)$$

where J_i is the haze free image, I_i is the original image, A_i is the global atmospheric light which can be estimated from dark channel prior, α_i is the alpha matte generated by 3rd layer, and $\oplus_f$ is the q-value Weighted Bounded sum with weights $w_1 = \alpha_i(x), w_2 = 1 - \alpha_i(x)$, here q is max gray or RGB value of a pixel, and $\alpha_i(x)$ and $(1 - \alpha_i(x))$ are weights. Although we can use back propagation approach to compute pixels' value $J_i(x)$ given the haze image pixel value $I_i(x)$ based on Eq. (17), for the sake of simplicity, we directly use the Eq. (18) mentioned by He et al. (2011) to compute the haze free image. As every $\alpha_i(x)$ is computed upon the whole image, the pixel of haze-free image is also computed upon whole image, so the whole image is also the convex region of this layer.

$$J_i(x) = \frac{I_i(x) - A_i}{max(\alpha_i(x), \alpha_0)} + A_i \quad (18)$$

where J_i is the haze free image, I_i is the original image, A_i is the global atmospheric light which can be estimated from dark channel prior, α_i is the alpha matte generated by 3rd layer, and α_0 is a threshold, a typical value is 1.

4.3. Experiments result

The haze-free experiment result

(1) The haze-free and texture information entropy
Texture information can give out a rough measure about the effect of haze-freeing, we use the entropy of the texture histogram to measure the effect of deleting haze from images. The entropy of the histogram is described in Eq. (19).
Haze makes the texture of an image unclear, so theoretically speaking, haze removing will increase the entropy of the texture histogram.

$$Entropy : H = -\sum_{i=0}^{G-1} p(i) log_2[p(i)] \quad (19)$$

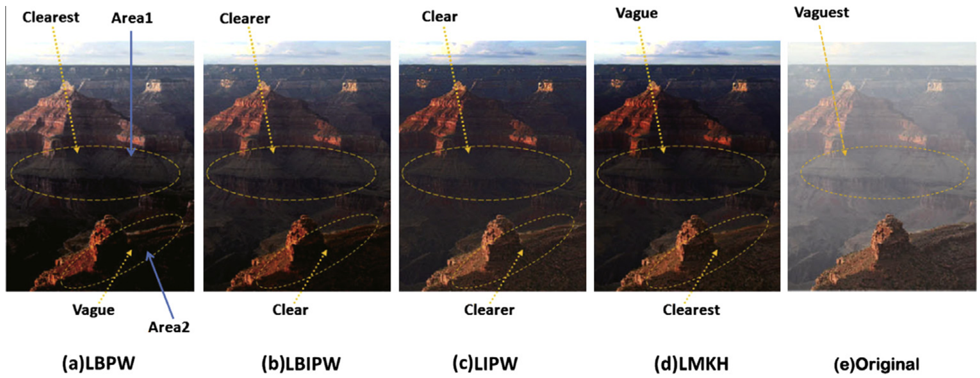


Fig. 6. The processing result of granular system for visual haze-free task.

The $p(i)$ denote the rate of each pattern in histogram. In general $p(i)$ define in Eq. (20). Here patterns we use are the LBPs in Eq. (14), where $Sigm(i_n - i_C) = 1$, if $i_n > i_C + 10$ else $Sigm(i_n - i_C) = 0$.

$$p(i) = h(i)/(NM), \quad i = 0, 1, \ldots, G - 1 \tag{20}$$

In Fig. 6(a), (b), (c), (d) and (e) are the results of LBPW, LBIPW, LIPW, the linear mode (LMKH) by He et al. (2011), and the original image respectively. From Fig. 6, we can see that the texture structure in the waist of a mountain becomes vaguer from LBPW, LBIPW, LIPW to LMKH. For the sake of the 2nd kind processing in the 1st-layer's granules pays much more attention to the contrast, LBPW has the highest ability to remove the haze, LBPW and LIPW are complementary approaches, LBIPW, which is the cooperation of them, has a similar ability as the linear approach proposed by He et al. (2011). According to the results showed in the Table 1, which are about texture information entropy of the image, we can see that the texture information entropy is increased after haze-free processing, so our approaches have higher ability to increase the texture information entropy than the linear approach proposed by He et al. (2011).

Table 1
The texture information entropy of the image blocks (Area1: the waist of a mountain; Area2: right bottom corner) in the Fig. 6.

Area	LBPW	LBIPW	LIPW	LMKH	Original
Area1	5.4852	5.2906	5.1593	4.8323	1.0893
Area2	6.1091	10.3280	10.2999	9.1759	8.3718

Theoretically speaking, LBPW is a pure texture processing, so LBPW has a highest value, LIPW is much more weaker than LBPW, LBIPW is the hybrid of LBPW and LIPW, so it has a average ability. The texture information entropy of the Area1 correctly reflects this fact. But for the Area2, as it already has a clearest texture structure in the original image, the deleting of haze may cause overdone. The texture information is over emphasized by LBPW in the Aera2, so it has a lowest texture information entropy and almost becomes a dark area. This fact means that overtreatment is more easier to appear in a non linear processing than a linear one in the haze-free task.

(2) The effect about the degree of fuzzyness

Just as the Theorem 1 mentioned above, the parameter λ in Eq. (10) can control the fuzzyness of a granule, when the parameter λ in Eq. (10) tends to infinite, a granule behaves from a fuzzy logical formula to a binary logical formula. This experiment is about the relation among the precision (*rmse*) of PSVM learning and λ parameters in the first and second layer. LBPW is a pure texture processing and pays much more attention to the contrast of an image's nearby pixels, a set of large λ is necessary for a low *rmse*, which corresponds to binary logic; but LBIPW and LIPW aphe pear to prefer fuzzy logic for a set of small λ when *rmse* is small. A possible explanation for this fact is that LBP proposed by Ojala et al. (1996) is binary, not fuzzy, and has a sound classification ability for image understanding under binary pattern, but LBIPW and LIPW are not binary, they have fuzzy information at least for the center pixel of a 3×3 small window (Fig. 7).

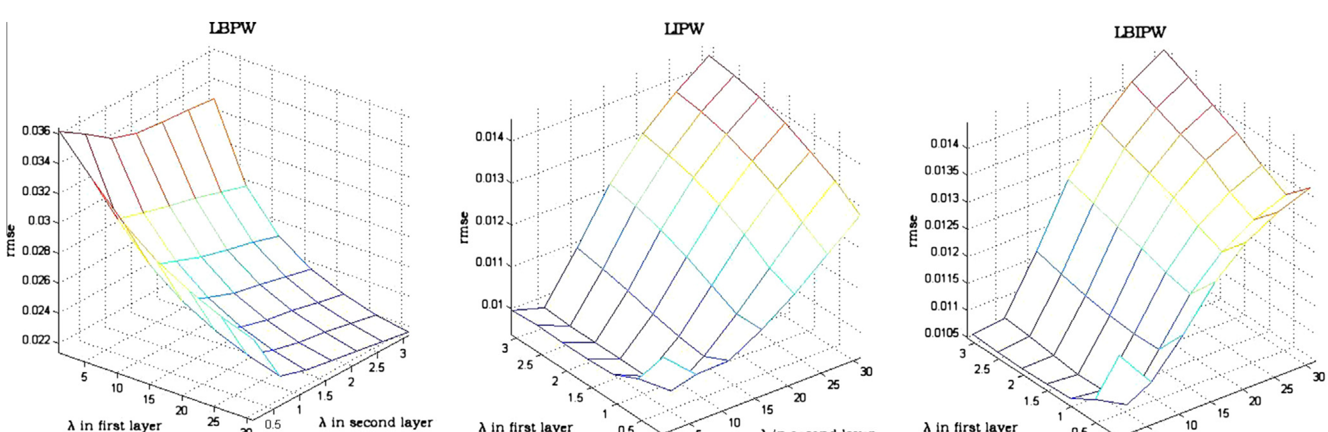


Fig. 7. The relation among the precision (*rmse*) of PSVM learning and λ parameters in the first and second layer, the parameter λ in Eq. (10) can control the fuzzyness of a granule.

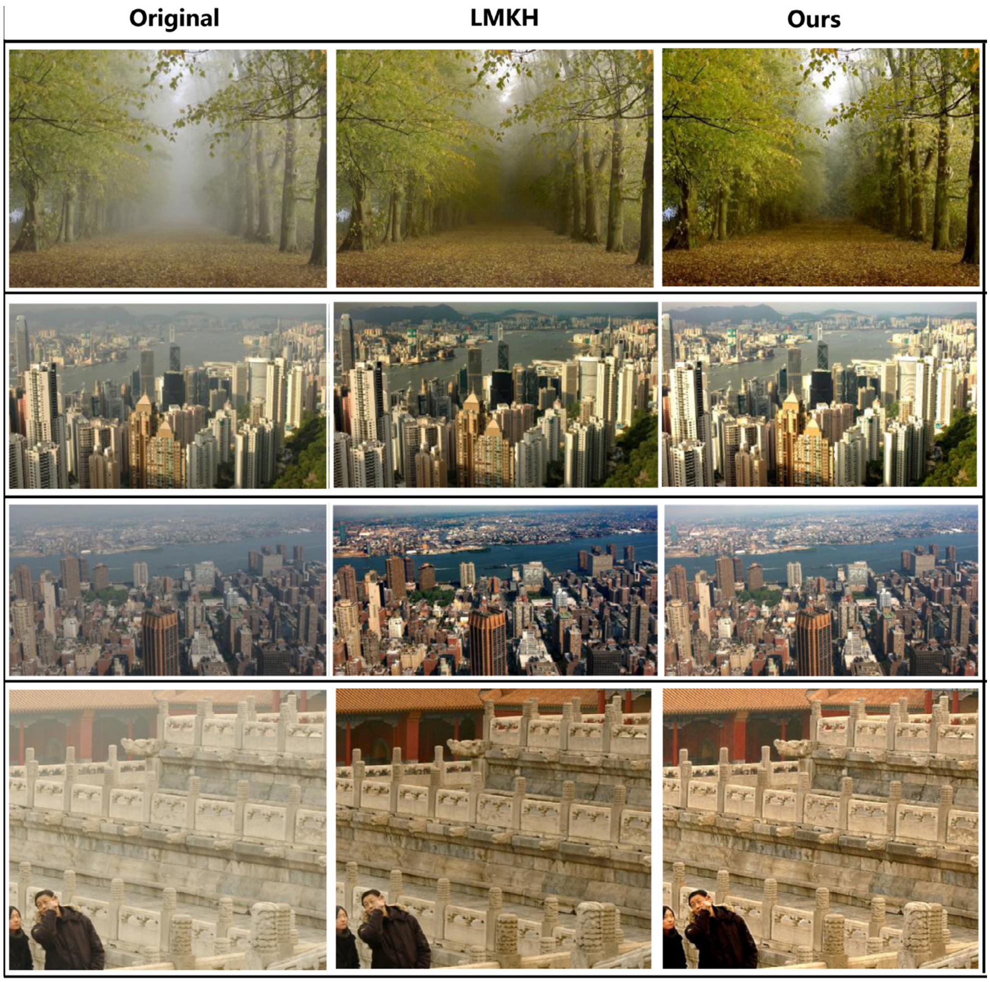


Fig. 8. More result of granular system for visual haze-free task.

(3) The comparison between our approach and LMKH
To illustrate the effect of our approach in haze-free task, we apply it on the other images and compare with LMKH (Fig. 8). Half of the result is better than the LMKH, the rest is as good as the LMKH by manual evaluation.

5. Discussion

In this paper, we give out a concrete example to show that the theory of GrC can help us to design the brain-like computer. The experimental results show that LPSVM is a promising approach for designing of a granular system similar to a columnar organization for image haze-removing task. The concept of granular computing is proposed by Bargiela and Pedrycz (2006). Just as he said: a granule is a clump of objects (points) drawn together by indistinguishability, similarity, proximity or functionality. The necessary of granular computing to study the information transformation in the pattern recognition lies in indistinguishability, similarity, proximity or functionality of sensed information Due to the local similarity in the information processing of pattern recognition, multi-scale information processing is a common phenomenon in pattern recognition. In actual fact, the GrC based on the leveled granular system aforementioned can simulate all multi-scale information processing with arbitrary small error. This fact is very important for the hot approach-deep learning. In this paper, we use a novel designing approach (LPSVM) to design a granular system similar to the structure of columnar organization of visual cortex, We demonstrate that fuzzy logic and machine learning can be hybrid and cooperated easily to design a granular system.

This approach not only give out a novel concrete realization of abstract models for granular computing mentioned Lin (2012), but also gives a new focus for deep learning. For more,the corresponding of GrC can simulate multi-scale information processing for the task of haze-free of images, and our experiments show that

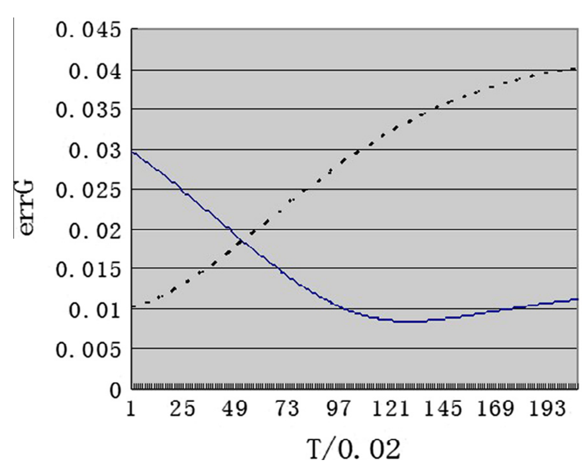


Fig. 9. Simulating fuzzy logical and-or by changing thresholds of Eq. (11). The X-axis is the threshold value divided by 0.02, the Y-axis is errG. The real line is errAnd between $I_1 \otimes_f I_2$ and $\overline{V}_i$, and the dot line is the errOr between $I_1 \oplus_f I_2$ and $\overline{V}_i$.

our approach has some approvement for the task of haze-free comparing to the approach proposed by the linear approach proposed by He et al. (2011).

For further directions, although our LPSVM gives out a concrete example for designing a granular system for haze free task, many details of LPSVM should be studied in the task of pattern recognition under the framework of deep learning, especially for the layered feature abstraction in the task of pattern recognition. For more, we will extend the investigation by looking at other nested layered computing for more complex tasks. However, since layered computing has no feedback, which is important for many visual tasks in dynamical situations, we also plan to extend our layered granular computing to a more general one which allows for both feedback and dynamical regulation for the task of computer vision.

Acknowledgments

This work is partially supported by the National Program on Key Basic Research Project (973 Program) (No. 2013CB329502), the National Natural Science Foundation of China (Nos. 61072085, 61035003, 61202212, 60933004), the National High-tech R and D Program of China (863 Program) (No. 2012AA011003), the National Science and Technology Support Program (2012BA107B02) and the China Information Technology Security Evaluation Center (CNITSEC-KY-2012-006/1).

Appendix A. Sigmoid function and Binary Logic

Theorem 1. *Suppose in Eq. (11), every* $w_{lik} = \beta_k T_i, \beta_k > 0, T_i > 0, 1 \leqslant k \leqslant K$, *for more,* $C = \{S_i | i = 1, \dots, L\}$ *is a class of index sets, and every index set* S_i *is a subset of* $\{1, 2, 3, \dots, K\}$, *then we have:*

(1) *If* $f(x_1, x_2, \dots, x_k) = \underset{l=1,\dots,L}{\vee} (\underset{j_l \in S_l}{\wedge} x_{j_l})$ *is a disjunctive normal form (DNF) formula, and the class* $C = \{S_i | i = 1, \dots, L\}$ *is the class which has the following two characters: (1). for every* $S_i, S_j \in C, S_i \cap S_j \neq S_k \in C$ *for all k and* $i \neq j$ *(this condition assures that* $f(x_1, x_2, \dots, x_k)$ *has a simplest form); (2). every* S_i *has the character* $\sum_{j \in S_i} \beta_j > 1$, *where* $S_i \in C$, *and any index sets* $S' \notin C$ *have character* $\sum_{j \in S'} \beta_j < 1$, *or if* $\sum_{j \in S'} \beta_j > 1$, *there must be an index set* $S_i \in C$ *such that* $S' \cap S_i = S_i$ *(this condition assures C is the largest), then the output described by Eq. (11) can simulate the DNF formula* $f(x_1, x_2, \dots, x_k) = \underset{l=1,\dots,L}{\vee} (\underset{i \in S_l}{\wedge} x_{j_l})$ *with arbitrary small error, where* $x_i = z_i$, *if the corresponding input* $I_i = z_i$, *or* $x_i = \bar{z}_i$ *if* $I_i = 1 - z_i$.

(2) *If a neural cell described by Eq. (11) can simulate the Boolean formula* $f(x_1, x_2, \dots, x_k)$ *with arbitrary small error, and* $(\underset{i \in S_l}{\wedge} x_i)$ *is an item in the disjunctive normal form of* $f(x_1, x_2, \dots, x_k)$, *i.e.* $f(x_1, x_2, \dots, x_k) = 1$ *at* $x_j = 1$ *for all* $j \in S_l$ *and* $x_j = 0$ *for all* $j \notin S_l$, *then* $\sum_{i \in S_l} \beta_i > 1$.

(3) *If a couple of index sets* S_{l_1} *and* S_{l_2} *can be found in the formula* $f(x_1, x_2, \dots, x_k) = \underset{l=1,\dots,k,}{\vee} (\underset{t \in S_l}{\wedge} x_t)$, *such that* $(\underset{t_1 \in S_{l_1}}{\wedge} x_{t_1}) \wedge (\underset{t_2 \in S_{l_2}}{\wedge} x_{t_2}) = z_i \wedge \bar{z}_i = false$, *then the output described by Eq. (11) can't simulate the formula* $f(x_1, x_2, \dots, x_k)$.

Proof.

(1) If $I_t = 1$, for all $t \in S_l$, and $I_t = 0$, for all $t \notin S_l$, because $\sum_{i \in S_l} \beta_i > 1$, then for the index set S_l is a subset of $\{1, 2, 3, \dots, K\}$, we have

$$\overline{V}_i = 1/[\exp(-\lambda(\overline{U}_i - T_i) + 1]$$
$$= 1/[\exp(-\lambda(\sum_{1 \leqslant k \leqslant K} w_{ik} I_k - T_i)) + 1]$$
$$= 1/[\exp(-\lambda(\sum_{i \in S_l} \beta_i - 1)T_i)) + 1],$$

so $\lim_{\lambda \to +\infty} \overline{V}_i = 1 = f(x_1, x_2, \dots, x_k)$. If $I_t = 1, \forall t \in S'; I_t = 0, \forall t \notin S'$ and $S' \notin C$, then according to the condition of this theorem: if $\sum_{i \in S'} \beta_i < 1$, $\lim_{\lambda \to +\infty} \overline{V}_i = 0 = f(x_1, x_2, \dots, x_k)$; if $\sum_{i \in S'} \beta_i > 1$, then there is an index set $S_i \in C$ such that $S' \cap S_i = S_i$, then $\lim_{\lambda \to +\infty} \overline{V}_i = 1 = f(x_1, x_2, \dots, x_k)$. So when $\lambda \to \infty$, the error between output described by Eq. (11) and $f(x_1, x_2, \dots, x_k)$ trends to 0.

(2) If the output described by Eq. (11) can simulate the Boolean formula $f(x_1, x_2, \dots, x_k)$ which is not a constant with arbitrary small error, and for a definite binary input $x_1, x_2, \dots, x_k$, then the arbitrary small error is achieved when λ trends to infinite and $(\overline{U}_i - T_i) = \sum_{k \in S_l} w_{ik} I_k - T_i \neq 0$ where S_l is the set of the labels and $I_i = 1$, for all $i \in S_l$, and $I_i = 0$, for all $i \notin S_l$.
The theorem's condition supposes that every $w_{ik} = \beta_k T_i, \beta_k > 0, T_i > 0, 1 \leqslant k \leqslant K$, and $x_1, x_2, \dots, x_k$ are binary number 0 or 1, so if $f(x_1, x_2, \dots, x_k)$ is not a constant, when $f(x_1, x_2, \dots, x_k) = 0$, there must be $\lim_{\lambda \to +\infty} V_i = 0$; and when $f(x_1, x_2, \dots, x_k) = 1$, it is necessary for $\lim_{\lambda \to +\infty} V_i = 1$. $\lim_{\lambda \to +\infty} V_i = 0$ needs that -$\lambda(\sum_{i \in S_l} \beta_i T_i - T_i)$ trends to minus infinite and $\lim_{\lambda \to +\infty} V_i = 1$ needs that -$\lambda(\sum_{i \in S_l} \beta_i T_i - T_i)$ trends to plus infinite. So if $f(x_1, x_2, \dots, x_k) = 1$ at $x_j = 1$ for all $j \in S_l$ and $x_j = 0$ for all $j \notin S_l$, in order to guarantee $\lim_{\lambda \to +\infty} err_{f(x_1, x_2, \dots, x_k)}(w_{i,1}, w_{i,2}, \dots, w_{i,k}, T_i) = 0, \sum_{i \in S_l} \beta_i > 1$ must be hold, here $err_{f(x_1, x_2, \dots, x_k)}(w_{i,1}, w_{i,2}, \dots, w_{i,k}, T_i)$ is the error between output described by Eq. (11) and $f(x_1, x_2, \dots, x_k)$.

(3) The third part of the theorem is based on the simple fact that for a single neuron V_i is monotone on every input I_i which can be z_i or $1 - z_i$. □

Appendix B. Sigmoid function and fuzzy logic

For more above granular computing can approximately simulate Bounded operator. Bounded operator $F(\oplus_f, \otimes_f)$ Bounded product $p \otimes_f q = \max(0, p + q - 1)$, Bounded sum $p \oplus_f q = \min(1, p + q)$. Based on Eq. (11), the membrane potential's fixed point under input I_k is $\overline{U}_i = \sum_k w_{ik} I_k$ and the output at the fixed point is $\overline{V}_i = 1/(\exp(-\overline{U}_i + T_i) + 1)$.

If there are only two inputs $I_1, I_2 (I_1, I_2 \in [0, 1])$ in Eq. (11), we set $w_1 = 1.0$ and $w_2 = 1.0$, then $\overline{U}_i = I_1 + I_2$.

Now we try to prove that the Bounded operator $F(\oplus_f, \otimes_f)$ is the best fuzzy operator to simulate neural cells described by (3) and the threshold Ti can change the neural cell from the bounded operator $\oplus_f$ to $\otimes_f$ by analyzing the output at the fixed point $\overline{V}_i = 1/(\exp(-\overline{U}_i + T_i) + 1)$. If $C > 0$ is a constant and $\overline{U}_i = I_1 + I_2 \geqslant C$, then $1/(\exp(-C + T_i) + 1) \leqslant \overline{V}_i < 1$. When $\overline{U}_i = I_1 + I_2 \to +\infty \overline{V}_i \to 1$, so in this case, if C is large enough, $\overline{V}_i \approx 1$. If -$C \leqslant \overline{U}_i = I_1 + I_2 \leqslant C$, then $1/(\exp(C + T_i) + 1) \leqslant \overline{V}_i \leqslant 1/(\exp(-C + T_i) + 1)$, according to equation (a). We can select a T_i, that makes $|T_i + \sum_{j=2}^{\infty} (-\overline{U}_i + T_i)^j / j! - \sum_{k=2}^{\infty} (-1)^k \exp(-k(\overline{U}_i - T_i))|$ small enough, then $\overline{V}_i \approx I_1 + I_2$.

$$\begin{aligned}\overline{V}_i &= 1/(\exp(-\overline{U}_i + T_i) + 1)\\ &= 1 - \exp(-\overline{U}_i + T_i) + \sum_{k=2}^{\infty}(-1)^k \exp(-k(\overline{U}_i - T_i))\\ &= \overline{U}_i - T_i - \sum_{j=2}^{\infty}(-\overline{U}_i + T_i)^j/j! + \sum_{k=2}^{\infty}(-1)^k \exp(-k(\overline{U}_i\_T_i))\\ &= \overline{U}_i - T_i - \sum_{j=2}^{\infty}(-\overline{U}_i + T_i)^j/j! + \sum_{k=2}^{\infty}(-1)^k \exp(-k(\overline{U}_i\_T_i)).(a)\end{aligned}$$

So in this case, $\overline{V}_i \approx I_1 \oplus_f I_2 = \min(1, I_1 + I_2)$.

Similarly, if $\overline{U}_i = I_1 + I_2 \to -\infty$ $\overline{V}_i \to 0$. So when C is large enough and $\overline{U}_i = I_1 + I_2 \leqslant -C < 0$, then $\overline{V}_i \approx 0$. When $-C \leqslant \overline{U}_i = I_1 + I_2 \leqslant C$, if we select a suitable T_i which makes $T_i + \sum_{j=2}^{\infty}(-\overline{U}_i + T_i)^j/j! - \sum_{k=2}^{\infty}(-1)^k \exp(-k(\overline{U}_i - T_i)) \approx 1$, then $\overline{V}_i \approx I_1 \otimes_f I_2 = \max(0, I_1 + I_2 - 1)$.

Based on above analysis, the Bounded operator fuzzy system is suitable for *GrC* described by Eq. (11) when $a_i = 1.0, w_1 = 1.0$ and $w_2 = 1.0$. For arbitrary positive a_i, w_1 and w_2, we can use corresponding q-value weighted universal fuzzy logical function based on Bounded operator to simulate such kind neural cells. If a weight w is negative, a N-norm operator $N(x) = 1 - x$ should be used.

Experiments done by scanning the whole region of (I_1, I_2) in $[0,1]^2$ to find the suitable coefficients for $\oplus_f$ and $\otimes_f$ show that above analysis is sound. We denote the input in Eq. (11) as $\vec{x} = (I_1, I_2)$. The "errOr" for $\oplus_f$ and "errAnd" for $\otimes_f$ are shown in Fig. 9 as the solid line and the dotted line respectively. In Fig. 9, the threshold T_i is scanned from 0 to 4.1 with step size 0.01. The best T_i in Eq. (4) for $\otimes_f$ is 2.54 and the best T_i in Eq. (4) for $\oplus_f$ is 0, when $a = 1.0, w_1 = 1.0$ and $w_2 = 1.0$. In this case the "errOr" and "errAnd" is less than 0.01. Our experiments show that suitable T_i can be found. So in most cases, the bounded operator $F(\oplus_f, \otimes_f)$ mentioned above is the suitable fuzzy logical framework for the neuron defined by Eq. (3). If the weight $0 < w_1$ and $0 < w_2$, we should use a q-value weighted bounded operator $F(\oplus_f, \otimes_f)$ to represent above neuron.

Appendix C. Associative condition and Demorgan law of *q*-weighted bounded operator

It is easily to see $\oplus_f$ follows the associative condition and $x_1 \oplus_f x_2 \oplus_f x_3 \ldots \oplus_f x_n = \min(q, \sum_{1 \leqslant i \leqslant n} w_i x_i)$.

For $\otimes_f$, we can prove the associative condition is hold also. The proof is listed as below:

If $w_1 p_1 + w_2 p_2 - (w_1 + w_2 - 1)q \geqslant 0$, we have:

$$\begin{aligned}(p_1 \otimes_f p_2) \otimes_f p_3 &= F_{\otimes_f}(F_{\otimes_f}(p_1, p_2, w_1, w_2), p_3, 1, w_3)\\ &= F_{\otimes_f}(w_1 p_1 + w_2 p_2 - (w_1 + w_2 - 1)q, p_3, 1, w_3)\\ &= \max(0, w_1 p_1 + w_2 p_2 - (w_1 + w_2 - 1)q + w_3 p_3\\ &\quad - (1 + w_3 - 1)q)\\ &= \max(0, w_1 p_1 + w_2 p_2 + w_3 p_3 - (w_1 + w_2 + w_3 - 1)q);\end{aligned}$$

if $w_1 p_1 + w_2 p_2 - (w_1 + w_2 - 1)q < 0$, we have

$$\begin{aligned}(p_1 \otimes_f p_2) \otimes_f p_3 &= F_{\otimes_f}(F_{\otimes_f}(p_1, p_2, w_1, w_2), p_3, 1, w_3)\\ &= F_{\otimes_f}(0, p_3, 1, w_3) = \max(0, 0 + w_3 p_3 - (1 + w_3 - 1)q)\\ &= \max(0, w_3 p_3 - w_3 q) \overset{for 0 \leqslant p_3 \leqslant q}{=} 0\\ &= \max(0, w_1 p_1 + w_2 p_2 + w_3 p_3 - (w_1 + w_2 + w_3 - 1)q);\end{aligned}$$

so $(p_1 \otimes_f p_2) \otimes_f p_3 = p_1 \otimes_f (p_2 \otimes_f p_3) = \max(0, w_1 p_1 + w_2 p_2 + w_3 p_3 - (w_1 + w_2 + w_3 - 1)q)$.

By inductive approach, we can prove that $\otimes_f$ also follows the associative condition and $x_1 \otimes_f x_2 \otimes_f x_3 \ldots \otimes_f x_n = \max(0, \sum_{1 \leqslant i \leqslant n} w_i x_i - (\sum_{1 \leqslant i \leqslant n} w_i - 1)q)$.

For more if we define $N(p) = q - p$ (usually, a negative weight w_i corresponds a N-norm), above weighted bounded operator $F(\oplus_f, \otimes_f)$ follows the Demorgan Law, i.e.

$$\begin{aligned}N(x_1 \oplus_f x_2 \oplus_f x_3 \ldots \oplus_f x_n) &= q - \min\left(q, \sum_{1 \leqslant i \leqslant n} w_i x_i\right)\\ &= \max\left(0, q - \sum_{1 \leqslant i \leqslant n} w_i x_i\right)\\ &= \max\left(0, \sum_{1 \leqslant i \leqslant n} w_i (q - x_i) - (\sum_{1 \leqslant i \leqslant n} w_i - 1)q\right)\\ &= N(x_1) \otimes_f N(x_2) \otimes_f N(x_3) \ldots \otimes_f N(x_n).\end{aligned}$$

References

Andrzej, B., & Pedrycz, W. (2006). The roots of granular computing. In *GrC* (pp. 806–809).

Castro, J. L. (1995). Fuzzy logic controllers are universal approximators. *IEEE Transactions on Systems, Man and Cybernetics, 25*(4), 629–635.

Fung, G., & Mangasarian, O. L. (2001). Proximal support vector machine classifiers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 77–86). ACM.

Haykin, S. (1994). *Neural networks: A comprehensive foundation*. Prentice Hall PTR.

Haykin, S. (2008). *neural networks: A comprehensive foundation*. Englewood cliffs: Prentive-Hall.

He, K., Sun, J., & Tang, X. (2011). Single image haze removal using dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 33*(12), 2341–2353.

Levin, A., Lischinski, D., & Weiss, Y. (2008). A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 30*(2), 228–242.

Lin, T. Y. (1998). Granular computing on binary relations I: Data mining and neighborhood systems. *Rough Sets in Knowledge Discovery, 1*, 107–121.

Lin, T. Y. (1999). Granular computing: Fuzzy logic and rough sets. *Computing with words in information/intelligent systems* (Vol. 1, pp. 183–200). Springer.

Lin, T. Y. (2007). *Neighborhood systems: A qualitative theory for fuzzy and rough sets*. Berkeley: University of California. 94720.

Lin, T. Y. (2012). Granular computing: Practices, theories, and future directions. In *Computational complexity* (pp. 1404–1420). Springer.

Li, H.-X., & Philip Chen, C. L. (2000). The equivalence between fuzzy logic systems and feedforward neural networks. *IEEE Transactions on Neural Networks, 11*(2), 356–365.

Liu, H., Xiong, S., & Wu, C.-a. (2012). Hyperspherical granular computing classification algorithm based on fuzzy lattices. *Mathematical and Computer Modelling*.

Mountcastle, V. B. (1997). The columnar organization of the neocortex. *Brain, 120*(4), 701–722.

Ojala, Timo, Pietikäinen, Matti, & Harwood, David (1996). A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition, 29*(1), 51–59.

Pedrycz, Adam, Hirota, Kaoru, Pedrycz, Witold, & Dong, Fangyan (2012). Granular representation and granular computing with fuzzy sets. *Fuzzy Sets and Systems, 203*, 17–32.

Yao, Y. Y. (1998). Relational interpretations of neighborhood operators and rough set approximation operators. *Information Sciences, 111*(1), 239–259.

Yao, Y. Y. (1999). Granular computing using neighborhood systems. In *Advances in soft computing* (pp. 539–553). Springer.

Yao, Y. Y. (2000). Granular computing: Basic issues and possible solutions. In *Proceedings of the 5th joint conference on information sciences* (Vol. 1, pp. 186–189) Citeseer.

Yao, Y. Y. (2001). On modeling data mining with granular computing. In *25th Annual international computer software and applications conference, 2001. COMPSAC 2001* (pp. 638–643). IEEE.

Yao, Y. Y. (2001). Information granulation and rough set approximation. *International Journal of Intelligent Systems, 16*(1), 87–104.

Yao, Y. (2006). Granular computing for data mining. In *Defense and security symposium* (pp. 624105). International Society for Optics and Photonics.

Yao, Y., & Deng, X. (2013). A granular computing paradigm for concept learning. In *Emerging paradigms in machine learning* (pp. 307–326). Springer.

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control, 8*(3), 338–353.

Zadeh, L. A. (1997). Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets and Systems, 90*(2), 111–127.

Zhang, L., & Zhang, B. (2003). Theory of fuzzy quotient space (methods of fuzzy granular computing). *Journal of Software, 14*(4), 770–776.

Zhang, L., & Zhang, B. (2004a). The quotient space theory of problem solving. *Fundamenta Informaticae, 59*(2), 287–298.

Zhang, L., & Zhang, B. (2004b). The quotient space theory of problem solving. *Fundamenta Informaticae, 59*(2), 287–298.

Zhang, L., & Zhang, B. (2005). Quotient space model based hierarchical machine learning. *International conference on neural networks and brain, 2005. ICNN&B'05* (Vol. 1). IEEE, pp. xiv–xiv.