Knowledge-Based Systems 258 (2022) 109941

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Fuzzy granular deep convolutional network with residual structures

Linjie He, Yumin Chen *, Keshou Wu

College of Computer Science and Technology, Xiamen University of Technology, Xiamen, 361024, China

ARTICLE INFO

Article history:
Received 15 March 2022
Received in revised form 4 September 2022
Accepted 25 September 2022
Available online 3 October 2022

Keywords:
Fuzzy sets
Granular computing
Fuzzy granular convolution
Fuzzy networks
Residual structures

ABSTRACT

In recent years, the deep neural network technology has developed rapidly and has been effective in processing and analyzing images, videos, sounds and many other aspects. However, traditional neural networks only learn the samples themselves and ignore the differences between samples. When the network depth is too deep, network degradation phenomenon may occur. To solve these drawbacks, we propose the fuzzy granular deep convolutional network with residual structures. Firstly, we define the concept, operation and correlation measures of fuzzy granules, and construct the granulation reference system by random sampling for fuzzy granulation. Furthermore, modules of granular neuron, granular activation function, granular convolution, and granular residual are defined, and the fuzzy granular deep convolutional network with residual structures is built. The loss functions and learning algorithm are designed for the granular neural network, and the fuzzy granular neural network is successfully trained. The fuzzy granular neural network owns the characteristics of multi-granularity, multi-angle and structured, which has good generalization performance. Finally, experiments are conducted on the Cifar100 and Tiny Imagenet datasets. The experimental results show the effectiveness of the residual granular neural network. We further found that the granular neural network could alleviate the degradation problem. Moreover granular neural network with fewer hidden layers can achieve the function of a neural network with more hidden layers.

1. Introduction

The concept of fuzzy sets and fuzzy computation were introduced by the American scientist Zadeh [1], where fuzzy sets represent the relationship between elements and sets by means of an affiliation function. Fuzzy set theory is an extension of classical set theory. In 1979, Zadeh first introduced the concept of granulation and granular computing. Pedrycz, a Canadian academician, proposed a variety of granular classification [2] and granular clustering algorithms [3]. In 1985 Hoboss [4] published an article on the title 'Granularity', which uses different granularities to describe real phenomena and solves complex problems. In 1997 Zadeh [5] stated that fuzzy logic and human reasoning logic are similar to human reasoning logic. Fuzzy information granulation is based on human reasoning approach to granulate information. In addition to fuzzy information granulation, there are many other granulation methods, such as rough sets granulation and shadowed sets granulation. Pederycz summarized and discussed these granulation methods under different conditions [6]. Lin successfully applied granular computing to knowledge discovery [7] and data mining [8]. Yao constructed neighborhood granulation by defining neighborhood relations [9,10]. In 2002, Miao proposed a knowledge-based granulation method [11], which can effectively deal with the attribute approximation problem [12]. Yager [13] pointed out that the idea of granular computing is important to promote the development of intelligent systems. Qian et al. proposed multi-granularity fusion learning [14] to learn decisions under different granularity patterns. In 2014, Qian [15] et al. designed a parallelized attribute approximation algorithm to improve the efficiency of granular computing operations on the attribute approximation problem. Sanchez et al. [16] proposed a granulation approach for Interval Type-2 Fuzzy information granules based on uncertain information theory. The new approach to modular neural network optimization based on a multi-objective hierarchical genetic algorithm has been proposed by Patricia Melin [17]. And Melin applied it to face detection, ear recognition and other pattern recognition tasks. Chen defined the structure of granules from the perspective of sets and vectors and further investigated the uncertainty and distance metrics of granules [18]. In 2020, modular granular neural networks [19] based on particle swarm optimization variants and fuzzy dynamic parameter adaptation were proposed by Daniela Sanchez et al. and applied to human recognition tasks. Li [20] integrated the boosting algorithm with the granular KNN algorithm to further improve the performance of the KNN algorithm. Granular computing is an arithmetic model defined from the perspective of human cognition, which is highly similar to human logic, cognition, and memory, and has been successfully applied in several fields [21–27].

* Corresponding author.
E-mail address: cym0620@163.com (Y. Chen).

https://doi.org/10.1016/j.knosys.2022.109941
0950-7051/

In 1943, Mcculloch and Pitts first proposed the concept of Artificial Neural Networks by analyzing neural network features that mimic those of living beings . In 1958, Rosenbaltt proposed the perceptron model [28], which implemented a single neuron. The proposal of the perceptron model brought the development of neural networks to its first climax. The second culmination of neural networks was accompanied by connectionism in the 1980s. By using the backpropagation algorithm, Rumelhart made it possible to successfully train neural networks with one or two hidden layers [29]. Hochreiter et al. proposed the LSTM structure to enable the application of neural networks to model time series data [30]. LeCun proposed the LeNet-5 network, which, along with its later generated variants, defined the modern convolutional neural network basic structure [31]. In 2006, Hioton proposed the deep belief network [32], which solved the problem that deep neural networks are difficult to train by greedy layer-by-layer training. The proposal of deep belief networks reignited the wave of neural networks. In 2012, the AlexNet [33] network won the ImageNet image recognition competition with a far second-place finish, which drew the attention of both academia and industry. Kai-Ming He et al. proposed the residual network [34], which somehow solved the problem of neural network too deep leading to gradient disappearance and network degradation. Once the residual network was proposed, it received wide attention from related researchers. Residual networks have been applied to several fields [35–39], and many variants have appeared [40,41]. Nevertheless, traditional neural networks only learn from the current samples. There are still some connections between the characteristics of multiple separate samples in the data set, which are overlooked by traditional neural networks. When the network depth is too deep, network degradation phenomenon may occur. To address these weaknesses, we design a fuzzy granular depth convolutional network with residual structures. The multi-angle learning mode of the granular depth convolutional network makes the granular neural network have better generalization performance and can achieve more hidden layer neural network functions with fewer granular hidden layers. The granulation method adopted in this paper belongs to fuzzy granulation, but it is different from general fuzzy granulation method. The fuzzy granulation of this algorithm is based on randomly selected reference samples. The granulation mode is part granulation rather than global granulation. In order to construct granular neural networks, some granular proposed operations are also different from the traditional fuzzy information granular operations. Based on the structural nature of granules, granular neurons are naturally parallelizable. Moreover, random reference granulation is used to save granulation time and space.

The contributions to this article are as follows. First we define a feature similarity-based granulation approach. Then we also define various granular operations and granularity metrics. After that granular neurons and granular activation functions are proposed. In the same way, the granular residual structure and granular convolution are defined. Further, we present the granular residual network and its optimization algorithm. Finally, the granular residual network is proved to be effective in the experiment.

Section 1 of this paper briefly introduces the background of granular computing and neural networks. In Section 2, we describe the definition of granule, granule operations and granule metrics. Then, Section 3 designs the structures of granular neurons, granular activation functions, granular convolution, and granular residual blocks to build the fuzzy granular deep convolutional network with residual structures. In addition, Section 4 shows the effectiveness of the proposed algorithm in several data sets. The last section concludes the whole paper.

2. Definition, operation and metric of fuzzy granules

2.1. Fuzzy granulation

Fuzzy sets is an effective tool for processing uncertainty information. For the information system $U = (X, C, D)$, $X = \{x_1, x_2, \dots, x_n\}$ is the sample set, $C = \{c_1, c_2, \dots, c_m\}$ is the feature set corresponding to the sample, and D is the decision set corresponding to the sample. Some samples $P = \{p_1, p_2, \dots, p_k\} \subseteq X$ are randomly selected from X as the reference sample set. For the given sample $x \in X$, where a single feature $c \in C$, $v(x, c) \in [0, 1]$ denotes the value of the sample x normalized over the feature c.

Definition 1. The information system $U = (X, C, D)$ with samples $x \in X$, $p \in P$ and a single feature $c \in C$. The similarity of x and p on feature c is:

$$S_c(x, p) = 1 - |v(x, c) - v(p, c)|. \tag{1}$$

Definition 2. For the information system $U = (X, C, D)$, we fuzzy granulate the sample $x \in X$ and the reference sample set $P = \{p_1, p_2, \dots, p_k\}$ on any feature $c \in C$ The sample x and the reference sample set P are granulated on the feature c to form a fuzzy granule. The fuzzy granule of x on c is defined as:

$$g_c(x) = \{g_c(x)_j\}_{j=1}^{k} = \{r_j\}_{j=1}^{k} = \{r_1, r_2, \dots, r_k\}, \tag{2}$$

$$where\ r_j = S_c(x, p_j) = 1 - |v(x, c) - v(p_j, c)|. \tag{3}$$

$g_c(x)$ is a collection. r_j is an element of the collection. And r_j is the similarity between the sample x and the reference sample p_j on the feature c. It is easy to know from Definition 1 that $r_j = s_c\left(x, p_j\right) \in [0, 1]$. We define $g_c\left(x\right)$ as the fuzzy granule, $g_c\left(x\right)_j$ as the jth granule kernel of the fuzzy granule $g_c\left(x\right)$. The granule is composed of granule kernels. If $\{r_j\}_{j=1}^{k} = \{0, 0, 0, \dots, 0\}$, we declare $\{r_j\}_{j=1}^{k}$ as 0-fuzzy granule. And if $\{r_j\}_{j=1}^{k} = \{1, 1, 1, \dots, 1\}$, we declare $\{r_j\}_{j=1}^{k}$ as 1-fuzzy granule.

Definition 3. For the information system $U = (X, C, D)$ with any $x \subseteq X$ and a reference sample set P, any feature subset $B = \{b_1, b_2, \dots, b_m\} \subseteq X$, the fuzzy granular vector of x on the feature subset B is:

$$G\left(x\right) = \left(g_{b1}\left(x\right), g_{b2}\left(x\right), \dots, g_{bm}\left(x\right)\right)^T. \tag{4}$$

$g_{bm}\left(x\right)$ is the fuzzy granular of x on the feature b_m, abbreviated as $G\left(x\right) = \left(g_1\left(x\right), g_2\left(x\right), \dots, g_m\left(x\right)\right)^T$.

The fuzzy granular vector is composed of fuzzy granules, and the fuzzy granule consists of granule kernels. So the fuzzy granular vector can also be represented by the granule kernel matrix.

$$G\left(x\right) = \begin{bmatrix} g_1(x)_1 & g_1(x)_2 & \cdots & g_1(x)_k \\ g_2(x)_1 & g_2(x)_2 & \cdots & g_2(x)_k \\ \cdots & \cdots & \cdots & \cdots \\ g_m(x)_1 & g_m(x)_2 & \cdots & g_m(x)_k \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1k} \\ r_{21} & r_{22} & \cdots & r_{2k} \\ \cdots & \cdots & \cdots & \cdots \\ r_{m1} & r_{m2} & \cdots & r_{mk} \end{bmatrix} \tag{5}$$

The fuzzy granular vector $G\left(x\right)$ is a collection of fuzzy granules, while the elements of the conventional vector are real numbers.

Example 1. The information system $U = (X, C, D)$ is shown in Table 1, where $X = \{x_1, x_2, x_3, x_4\}$, $C = \{a, b, c\}$, $D = \{dog, cat\}$.

Table 1
An example of the information system.

U	a	b	c	dog	cat
x_1	0.3	0.6	0.6	1	0
x_2	0.5	0.4	0.5	1	0
x_3	0.1	0.2	0.9	0	1
x_4	0.4	0.7	0.1	0	1

Let the reference set be $P = \{x_1, x_3\}$. The fuzzy granulation of the sample set $X = \{x_1, x_2, x_3, x_4\}$ on feature a is the results of:

$$g_a(x_1) = \{1, 0.8\}, g_a(x_2) = \{0.8, 0.6\}, g_a(x_3) = \{0.8, 1\}, g_a(x_4) = \{0.9, 0.7\}. \tag{6}$$

The fuzzy granulation of the sample set $X = \{x_1, x_2, x_3, x_4\}$ on feature b is the results of:

$$g_b(x_1) = \{1, 0.6\}, g_b(x_2) = \{0.8, 0.8\}, g_b(x_3) = \{0.6, 1\}, g_b(x_4) = \{0.9, 0.5\}. \tag{7}$$

The fuzzy granulation of the sample set $X = \{x_1, x_2, x_3, x_4\}$ on feature c is the results of:

$$g_c(x_1) = \{1, 0.7\}, g_c(x_2) = \{0.9, 0.6\}, g_c(x_3) = \{0.7, 1\}, g_c(x_4) = \{0.5, 0.2\}. \tag{8}$$

The granular vector of the sample $X = (x_1, x_2, x_3, x_4)$ are:

$$G_{\{a,b,c\}}(x_1) = (g_a(x_1), g_b(x_1), g_c(x_1))^T = (\{1, 0.8\}, \{1, 0.6\}, \{1, 0.7\})^T, \tag{9}$$

$$G_{\{a,b,c\}}(x_2) = (g_a(x_2), g_b(x_2), g_c(x_2))^T = (\{0.8, 0.6\}, \{0.8, 0.8\}, \{0.9, 0.6\})^T, \tag{10}$$

$$G_{\{a,b,c\}}(x_3) = (g_a(x_3), g_b(x_3), g_c(x_3))^T = (\{0.8, 1\}, \{0.8, 1\}, \{0.7, 1\})^T, \tag{11}$$

$$G_{\{a,b,c\}}(x_4) = (g_a(x_4), g_b(x_4), g_c(x_4))^T = (\{0.9, 0.7\}, \{0.9, 0.5\}, \{0.5, 0.2\})^T. \tag{12}$$

2.2. Fuzzy granular operations and metrics

This subsection gives the operations and metrics of granules.

Definition 4. Let $g_a(x), g_b(x)$ be the two fuzzy granules of sample x on features a and b respectively, then the addition, subtraction, multiplication and division operations of fuzzy granules are as follows:

$$g_a(x) + g_b(x) = \{g_a(x)_j + g_b(x)_j\}_{j=1}^k = \{g_a(x)_1 + g_b(x)_1, g_a(x)_2 + g_b(x)_2, \ldots, g_a(x)_k + g_b(x)_k\}, \tag{13}$$

$$g_a(x) - g_b(x) = \{g_a(x)_j - g_b(x)_j\}_{j=1}^k = \{g_a(x)_1 - g_b(x)_1, g_a(x)_2 - g_b(x)_2, \ldots, g_a(x)_k - g_b(x)_k\}, \tag{14}$$

$$g_a(x) * g_b(x) = \{g_a(x)_j * g_b(x)_j\}_{j=1}^k = \{g_a(x)_1 * g_b(x)_1, g_a(x)_2 * g_b(x)_2, \ldots, g_a(x)_k * g_b(x)_k\}, \tag{15}$$

$$g_a(x) / g_b(x) = \{g_a(x)_j / g_b(x)_j\}_{j=1}^k = \{g_a(x)_1 / g_b(x)_1, g_a(x)_2 / g_b(x)_2, \ldots, g_a(x)_k / g_b(x)_k\}. \tag{16}$$

Example 2. The information system $U = (X, C, D)$ is shown in Table 1. From Example 1, it is easy to know that:

$$g_a(x_2) = \{0.8, 0.6\}. \tag{17}$$

$$g_b(x_2) = \{0.8, 0.8\}. \tag{18}$$

According to Definition 4:

$$g_a(x_1) + g_b(x_1) = \{0.8 + 0.8, 0.6 + 0.8\} = \{1.6, 1.4\}. \tag{19}$$

$$g_a(x_1) - g_b(x_1) = \{0.8 - 0.8, 0.6 - 0.8\} = \{0, -0.2\}. \tag{20}$$

$$g_a(x_1) * g_b(x_1) = \{0.8 * 0.8, 0.6 * 0.8\} = \{0.64, 0.48\}. \tag{21}$$

$$g_a(x_1) / g_b(x_1) = \{0.8/0.8, 0.6/0.8\} = \{1, 0.75\}. \tag{22}$$

Definition 5. Let $g_a(x), g_a(y)$ be the two fuzzy granules of sample x on feature a, then the addition, subtraction, multiplication and division operations of fuzzy granules are as follows:

$$g_a(x) + g_a(y) = \{g_a(x)_j + g_a(y)_j\}_{j=1}^k = \{g_a(x)_1 + g_a(y)_1, g_a(x)_2 + g_a(y)_2, \ldots, g_a(x)_k + g_a(y)_k\}, \tag{23}$$

$$g_a(x) - g_a(y) = \{g_a(x)_j - g_a(y)_j\}_{j=1}^k = \{g_a(x)_j - g_a(y)_j, g_a(x)_j - g_a(y)_j, \ldots, g_a(x)_j - g_a(y)_j\}, \tag{24}$$

$$g_a(x) * g_a(y) = \{g_a(x)_j * g_a(y)_j\}_{j=1}^k = \{g_a(x)_1 * g_a(y)_1, g_a(x)_2 * g_a(y)_2, \ldots, g_a(x)_k * g_a(y)_k\}, \tag{25}$$

$$g_a(x) / g_a(y) = \{g_a(x)_j / g_a(y)_j\}_{j=1}^k = \{g_a(x)_1 / g_a(y)_1, g_a(x)_2 / g_a(y)_2, \ldots, g_a(x)_k / g_a(y)_k\}. \tag{26}$$

Example 3. The information system $U = (X, C, D)$ is shown in Table 1. From Example 1, it is easy to know that:

$$g_a(x_1) = \{1, 0.8\}. \tag{27}$$

$$g_a(x_2) = \{0.8, 0.6\}. \tag{28}$$

According to Definition 5:

$$g_a(x_1) + g_a(x_2) = \{1 + 0.8, 0.8 + 0.6\} = \{1.8, 1.4\}. \tag{29}$$

$$g_a(x_1) - g_a(x_2) = \{1 - 0.8, 0.8 - 0.6\} = \{0.2, 0.2\}. \tag{30}$$

$$g_a(x_1) * g_a(x_2) = \{1 * 0.8, 0.8 * 0.6\} = \{0.8, 0.48\}. \tag{31}$$

$$g_a(x_1) / g_a(x_2) = \{1/0.8, 0.8/0.6\} = \{1.25, 1.3334\}. \tag{32}$$

The result of two fuzzy granules fourth operations is also a fuzzy granule. The four fundamental operations in Definition 4 are operations on fuzzy granules of the same sample on different features. The operations in Definition 5 are operations on fuzzy granules of different samples on the same feature.

Definition 6. Let the fuzzy granular vectors be $G(x) = (g_1(x), g_2(x), \ldots, g_m(x))^T$ and $G(y) = (g_1(y), g_2(y), \ldots, g_m(y))^T$, then the dot product of two fuzzy granular vectors is defined as:

$$G(x) \cdot G(y) = g_1(x) * g_1(y) + g_2(x) * g_2(y) + \cdots + g_m(x) * g_m(y). \tag{33}$$

Example 4. Assuming that:

$$G(x) = (g_1(x), g_2(x), g_3(x))^T, G(y) = (g_1(y), g_2(y), g_3(y))^T. \tag{34}$$

$$g_1(x) = \{1, 1, 1\}, g_2(x) = \{1, 2, 3\}, g_3(x) = \{0, 1, 0\}. \tag{35}$$

$$g_1(y) = \{2, 2, 2\}, g_2(y) = \{0, 4, 2\}, g_3(y) = \{2, 4, 6\}. \tag{36}$$

As can be seen from Definition 6:

$$G(x)\cdot G(y)=\{1*2,\,1*2,\,1*2\}+\{1*0,\,2*4,\,3*2\}+\{0*2,\,1*4,\,0*6\} \tag{37}$$

$$G(x)\cdot G(y)=\{2,2,2\}+\{0,8,6\}+\{0,4,0\} \tag{38}$$

$$G(x)\cdot G(y)=\{2,12,8\} \tag{39}$$

The dot product of two fuzzy granular vectors results in one fuzzy granule. Let the fuzzy granular vector be $G(x)=(g_1(x),g_2(x),g_3(x),\ldots,g_m(x),1)$, and the weight fuzzy granular vector be $W=(w_1,w_2,w_3,\ldots,w_m,b)$, whose dot product is

$$W\bullet G(x)=w_1*g_1(x)+w_2*g_2(x)+\cdots+w_m*g_m(x)+1*b. \tag{40}$$

Definition 7. Let the fuzzy granule be $g(x)=\{r_j\}_{j=1}^{k}$, then the fuzzy granular function is defined as:

$$f(g(x))=f\left(\{r_j\}_{j=1}^{k}\right)=\{f(r_1),\,f(r_2),\ldots,f(r_k)\}. \tag{41}$$

Definition 8. Let the fuzzy granule be $g(x)=\{r_j\}_{j=1}^{k}$, its quality is:

$$Quality(g(x))=\sum_{j=1}^{k} r_j. \tag{42}$$

Definition 9. Let the m-dimensional fuzzy granular vector be $G=(g_1,g_2,\ldots,g_m)$, then the fuzzy granular vector norm granular are defined as:

(a). The 1-norm of a fuzzy granular vector:

$$\|G\|_1=\sum_{i=1}^{m}|g_i|. \tag{43}$$

(b). The 2-norm of a fuzzy granular vector:

$$\|G\|_2=\sqrt{\sum_{i=1}^{m} g_i*g_i}=\sqrt{G\cdot G}. \tag{44}$$

(c). The p-norm of a fuzzy granular vector:

$$\|G\|_p=\left(\sum_{i=1}^{m} {g_i}^p\right)^{\frac{1}{p}}. \tag{45}$$

The results of the norm operations of fuzzy granular vectors are fuzzy granules. The norm operation provides a way to transform from fuzzy granular vectors to fuzzy granules.

3. Fuzzy granular residual network

3.1. Granular activation functions and granular neurons

The activation function is important for neural network training to learn complex nonlinear functions, and it provides an important nonlinear factor for neural networks. To enable the granular neural network to also fit various complex nonlinear functions, we introduce several nonlinear granular activation functions for the granular neural network. Both the input and output of the granular activation functions are fuzzy granules.

In traditional neural networks, the common activation functions are sigmoid function, tanh function, ReLU function, Leaky Relu function, etc. We define the granular sigmoid function, the granular tanh function, the granular ReLU function, and the granular Leaky Relu function, respectively.

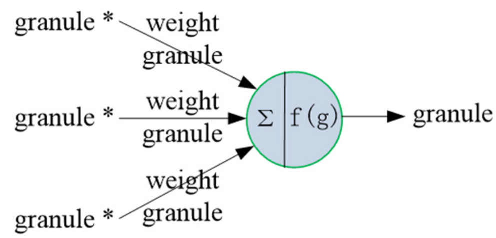

Fig. 1. Granular neurons.

Definition 10. Let the fuzzy granule be $g(x)=\{r_j\}_{j=1}^{k}$. Its granular sigmoid function, granular tanh function, granular ReLU function, and granular Leaky Relu function are as follows:

$$f_1(g(x))=\left\{\frac{1}{1+e^{-r_j}}\right\}_{j=1}^{k}. \tag{46}$$

$$f_2(g(x))=\left\{\frac{e^{r_j}-e^{-r_j}}{e^{r_j}+e^{-r_j}}\right\}_{j=1}^{k}. \tag{47}$$

$$f_3(g(x))=\left\{\max(0,r_j)\right\}_{j=1}^{k}. \tag{48}$$

$$f_4(g(x))=\{\begin{cases}\alpha r_j, & x<0\\ r_j, & x\geq 0\end{cases}\}_{j=1}^{k}. \tag{49}$$

The derivatives of the granular activation functions are respectively:

$$f_1{}'(g(x))=\left\{\frac{d}{dr_j}f_1(r_j)\right\}_{j=1}^{k}=\left\{\frac{1}{1+e^{-r_j}}\left(1-\frac{1}{1+e^{-r_j}}\right)\right\}_{j=1}^{k}. \tag{50}$$

$$f_2{}'(g(x))=\left\{\frac{d}{dr_j}f_2(r_j)\right\}_{j=1}^{k}=\left\{1-\left(\frac{e^{r_j}-e^{-r_j}}{e^{r_j}+e^{-r_j}}\right)^2\right\}_{j=1}^{k}. \tag{51}$$

$$f_3{}'(g(x))=\left\{\frac{d}{dr_j}f_3(r_j)\right\}_{j=1}^{k}=\{\begin{cases}0, & \alpha r_j<0\\ 1, & \alpha r_j=0\\ undefined, & \alpha r_j=0\end{cases}\}_{j=1}^{k}. \tag{52}$$

$$f_4{}'(g(x))=\left\{\frac{d}{dr_j}f_4(r_j)\right\}_{j=1}^{k}=\{\begin{cases}\alpha, & \alpha r_j<0\\ 1, & \alpha r_j=0\\ undefined, & \alpha r_j=0\end{cases}\}_{j=1}^{k}. \tag{53}$$

The granular neuron is analogous to the traditional neurons as shown in Fig. 1. The input of the granular neuron is a fuzzy granular vector and the output is fuzzy granule. The fuzzy granular vector and the weight granular vector are dotted to obtain a fuzzy granule. Then the fuzzy granule is put into the granule activation function to get the output fuzzy granule.

3.2. Granular convolution

Definition 11. Let the fuzzy granular vector be $G=(g_1(x),g_2(x),\ldots,g_m(x))$, the kernel granular vector be $K=(k_1,k_2,\ldots,k_n,)$, and the granular vector convolution be defined as:

$$S(G,K)=(G\bullet K)[m]=\sum_{n} G[m+n]\cdot K[n]. \tag{54}$$

When we train a granular convolutional network, G and K perform a convolution operation to output a granular vector S as shown in Fig. 2. When the error is back-propagated, an error granular vector $E=\frac{\partial L}{\partial S}$ is generated, where L is the granular loss function. In order to correct the weights of the kernel granular vector K, we need to perform a derivative operation on its weights.

$$\frac{\partial L}{\partial K}=\sum_{n} E\cdot G[m+n]. \tag{55}$$

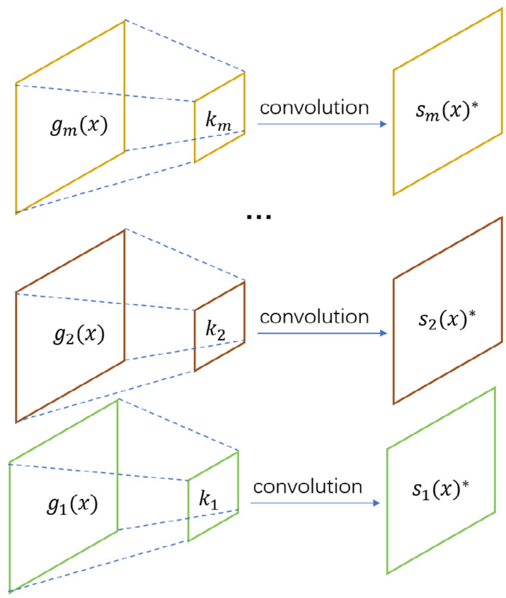

Fig. 2. Granular convolution operation.

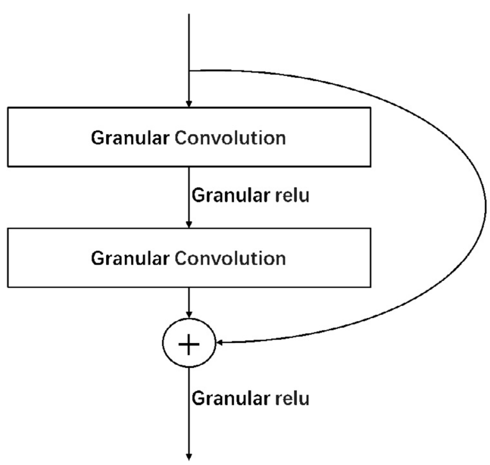

Fig. 3. Granular residual structure.

3.3. Granular residual structure

In order to solve the degradation problem when the granular neural network is overly deep, the granular residual block is proposed. For the ordinary granular neural network, the input fuzzy granular vector is $G(x)$, the feature granular vector after the granular neural network learning is $D(G(x))$. We define the granular residual as $F(G(x)) = D(G(x)) - G(x)$, then the feature granular vector learned by the granular residual block is $D(G(x)) + G(x)$. The granular residual block is shown in Fig. 3.

3.4. Granular neural network loss functions

In deep learning, designing the loss function or cost function is a very important aspect. The function of the loss function is to evaluate the difference between the predicted value of the model and the true value. For a general classification network, a smaller loss of the neural network means that the model classifies better.

In a granular neural network, the input is a feature fuzzy granular vector and the output is a decision fuzzy granular vector. We extend the onehot encoding of the sample labels to the labeled granular vector. The decision fuzzy granular vector can be compared with the label fuzzy granular vector after the softmax function, which constitutes the granular loss function.

Definition 12. Let the decision fuzzy granular vector $G = \{g_1, g_2, g_3, \ldots, g_l\}$, where $g_i = \{r_j\}_{j=1}^{k}$ denotes the output fuzzy granules. The label fuzzy granular vector $D = \{d_1, d_2, d_3, \ldots, d_l\}$, where $d_i = \{s_j\}_{j=1}^{k}$ denotes the label fuzzy granules. The granular loss function is defined as follows.

(a). Absolute value loss function:

$$L(G, D) = \frac{1}{l}\|G - D\|_1 = \frac{1}{l}\sum_{i=1}^{l}|g_i - d_i|. \tag{56}$$

The absolute value loss function metric of two granular vectors is the 1-norm granular mean of the difference between the two granular vectors. The result is one granule. The difference between the output granular vector and the labeled granular vector can be measured. This difference granule can be back propagated to correct the weighted granule in the granule classifier.

(b). Squared loss function:

$$L(G, D) = \frac{1}{l}\|G - D\|_2^2 = \frac{1}{l}\sum_{i=1}^{l}(g_i - d_i)^2. \tag{57}$$

The squared loss function measure of two granular vectors is the 2-norm granular squared mean of the difference between the two granular vectors, resulting in one granule.

(c). Cross-entropy loss function:

$$L(G, D) = -log(P) \cdot D = \sum_{i=1}^{l} log(p_i) * d_i, \tag{58}$$

where $P = (p_1, p_2, \ldots, p_l)$ is the probability granular vector after softmax. p_i is the probability granule denoted as: $p_i = \frac{e^{g_i}}{\sum_{i=1}^{l} e^{g_i}}$

3.5. Fuzzy granular deep convolutional network with residual structures

The granular residual network is designed with reference to the structure of residual network. The residual blocks are added to the granular neural network through the shortcut mechanism. The structure of our network is similar to the residual network. However the inputs of the granular residual network are fuzzy granules, and the outputs are also fuzzy granules. The structure diagram of the granular residual network is Fig. 4.

After finishing the construction of the granular neural network structure, we start to train the granular neural network in the following. The training algorithm of the granular neural network is shown in Algorithm 1.

After completing the network training, we apply the residual granular neural network to the classification task. The granular classification algorithm is shown below.

4. Experimental analysis

In order to verify the classification performance of the fuzzy granular residual network, we choose Cifar100 [42] and Tiny Imagenet [43] datasets for comparison experiments. Cifar100 consists of 60,000 three-channel color images of size 32*32, with 50,000 as the training set and 10,000 as the test set. The classification number of Cifar100 dataset is 100 classes. The Tiny Imagenet is

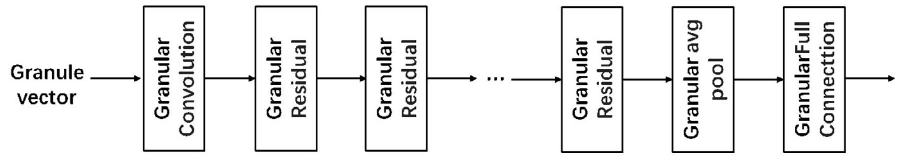

Fig. 4. Granular residual network structure diagram.

Algorithm 1 Granular residual network learning algorithm

Input: The training set is $U = (X, Y)$, where x_i is an m-dimensional feature vector, $x_i \in X \subseteq R^m$, y_i is an l-dimensional label vector, which means it contains l categories, $y_i \in Y \subseteq R^l$, $i = 1, 2, \ldots, n$.
Output: Granular weight matrix W and granular threshold b.

(1)Randomly selected some samples as the reference sample set $S = P_1, P_2, \ldots, P_k$.
(2)The training set p is fuzzy granulated in the reference set P as $GT = \{(G(x_1), G(y_1)), (G(x_2), G(y_2)), \ldots, (G(x_n), G(y_n))\}$, where the feature fuzzy granular vector is $G(x_i) = \{g_1(x_i), g_2(x_i), \ldots, g_m(x_i)\}$, the label fuzzy granular vector is $G(y_i) = \{g_1(y_i), g_2(y_i), \ldots, g_l(y_i)\}$.
(3)Construct a multilayer granular neural network and randomly initialize the granular weights in the network.
(4)For $x \in X$, input the characteristic fuzzy granular vector $G(x) = \{g_1(x), g_2(x), \ldots, g_m(x)\}$ into the granular neural network for forward calculation; after the hidden layer calculation and softmax, the output granular vector is $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_l\}$.
(5)Calculate the loss function $L(G(y), \alpha)$ of the output fuzzy granular vector and the label fuzzy granular vector , called a lossy fuzzy granule.
(6)The lossy fuzzy granule is propagated back to the granular weight vector in the modified granular neural network according to the gradient direction.
(7)Go to (4) until the loss function converges or the maximum number of iterations is reached.
(8)Output granular neural network weight matrix W and granular threshold b, and reference sample set $S = \{P_1, P_2, \ldots, P_k\}$.

Algorithm 2 Granular residual network classification algorithm

Input: The reference set $S = P_1, P_2, \ldots, P_k$, granular neural network weight matrix W and granular threshold b, test samples x.
Output: The class of sample x.

(1)The test sample x is granularized in the reference set S as $G(x) = \{g_1(x), g_2(x), \ldots, g_m(x)\}$.
(2)Input the granular vector $G(x)$ into the granular neural network for forward computation; after the hidden layer computation and softmax, output l fuzzy granules α_i.
(3)Calculate the size of the fuzzy granule α_i by the formula $\alpha_i = \sum_{j=1}^{k} r_j$, where r_j is the granule kernel of α_i.
(4)Calculate the probability of each class $\sigma_i = \frac{e^{\alpha_i}}{\sum_{t=1}^{l} e^{\alpha_t}}$.
(5)Determine the test sample as the class with the highest probability.

selected from the Imagenet dataset. It contains 200 categories, each category contains 500 training images and 50 test images, total 100000 training images and 10000 test images. the resolution of the images in Tiny Imagenet is 64*64, and the images are all three-channel color images. The description of the datasets is shown in Table 2.

Table 2
Description of the datasets.

Dataset	Instance	Size	Class
Cifar100	60 000	32*32	100
Tiny Imagenet	110 000	64*64	200

The purpose of this article is not to achieve in best results on the datasets. Rather, it is to compare the differences between granular neural network and conventional neural networks with similar network structures, the same number of network layers, and the same network hyperparameter settings. In the experimental part, we did not do any data augmentation on the datasets, and the learning rate was set by default and not manually adjusted during the training process. The experimental code is available at this link (https://github.com/huanggguagua555/Fuzzy-Granular-Residual-Network).

4.1. Cifar100

We first normalized the sample set data. Then k samples were randomly selected from the sample set as the reference set and the sample set was granularized. Because the resolution of the images in the Cifar100 dataset is $32 * 32$, the plane size of all convolution layers is set to $3 * 3$ to train the network effectively.

We selected Resnet18 and Resnet34 to compare with our proposed fuzzy granular neural network. The fuzzy granular neural networks are denoted as GResnet18 and GResnet34, respectively. Resnet18 has a similar structure and the same number of layers as GResnet18, and Resnet34 has a similar structure and the same number of layers as GResnet34. To ensure as fair a comparison as possible, the optimizer for all networks was Aadm with the learning rate of 0.001, beta1 = 0.9, beta2 = 0.999, and epsilon = 1e−8. Mini-batch was set to 128 for all networks.

To show the training process of granular neural network, we exhibit loss curves and accuracy curves. In the course of training, the training set loss and test set loss of Resnet34 are illustrated in Fig. 5. Fig. 6 shows the accuracy of the training set and the accuracy of the training set for Resnet34. It can be seen from Fig. 5 and Fig. 6 that the granular neural network can converge well without over-fitting.

Due to the different number of reference samples selected for fuzzy granulation of the data, the performance of the granular neural network performs differently. We selected 3 to 15 reference samples for GResnet18 network to fuzzy granulate the training data. For the GResnet18 network, the testing set correct rates corresponding to different numbers of reference granules are shown in Fig. 7. When the number of reference granules is less than 11, the correct classification rate gradually increases to about 60%. When the number of reference granules reached 13, the classification rate oscillates around 60.4% as the number of reference granules increases. However, when the number of reference granules reached 15, the correct classification rate dropped to 60.34% (see Fig. 7).

We also selected 3 to 15 reference samples for GResnet34 network for fuzzy granulation, and the correct classification rates of the test set corresponding to different numbers of reference

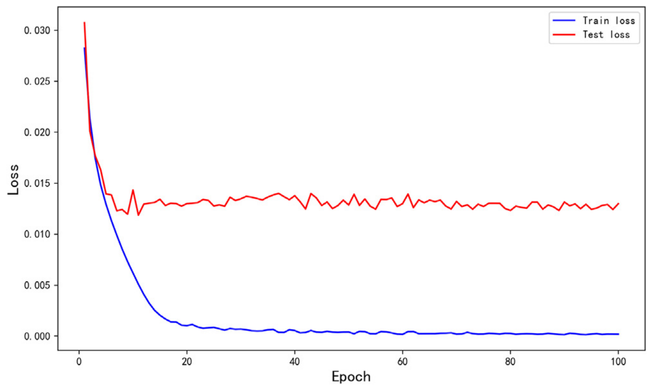

Fig. 5. The train loss and test loss of GResnet34 on Cifar100.

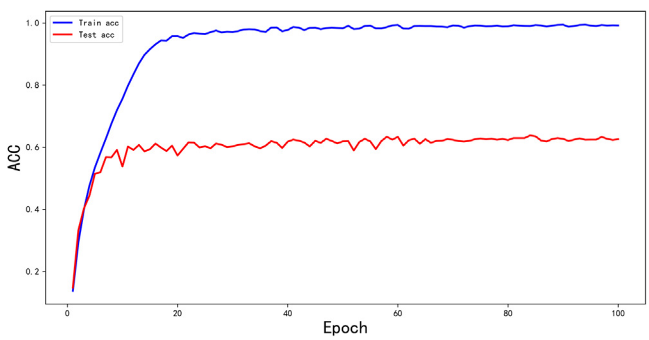

Fig. 6. The train accuracy and test accuracy of GResnet34 on Cifar100.

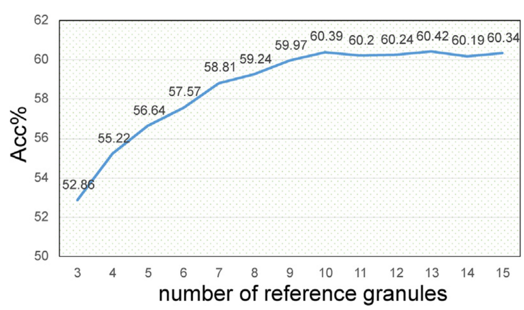

Fig. 7. Impacts of reference granules number k on accuracy in GResnet18.

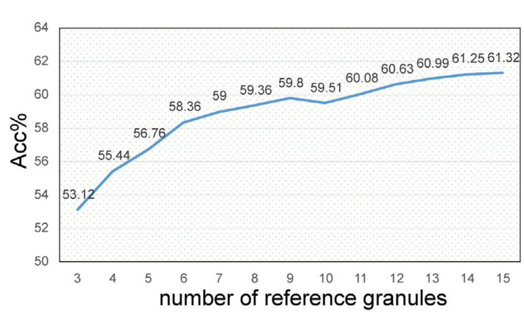

Fig. 8. Impacts of reference granules number k on accuracy in GResnet34.

granules are shown in Fig. 8. As the number of reference samples granulated increases, the test set correct rate increases in general. When the number of reference granules reaches 12 to 14, the correct rate is stable. In summary, the test set correct rate is basically positively correlated with the number of reference samples granulated, and the test set correct rate will gradually converge

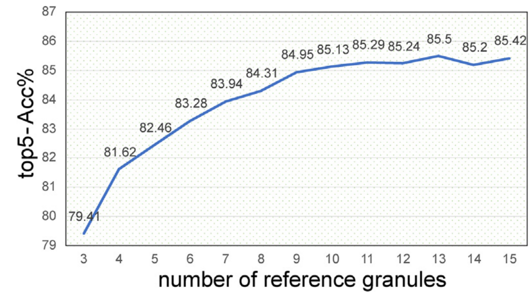

Fig. 9. Impacts of reference granules number k on top5-accuracy in GResnet18.

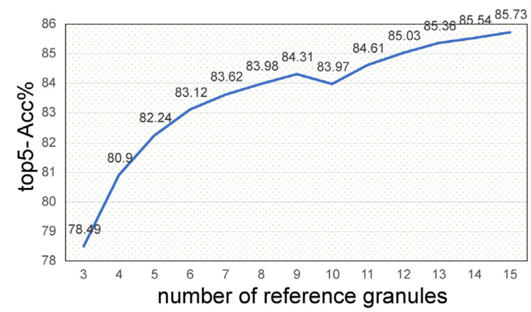

Fig. 10. Impacts of reference granules number k on top5-accuracy in GResnet34.

Table 3
Effects of random granules in Cifar100.

Networks	Difference value	t = 1	t = 2	t = 3	t = 4	t = 5
GResnet18	Accuracy difference	0.31%	1.03%	0.95%	1.79%	1.49%
GResnet18	Top5 accuracy difference	0.12%	0.93%	0.87%	1.37%	1.5%
GResnet34	Accuracy difference	0.93%	1.37%	1.79%	1.38%	1.15%
GResnet34	Top5 accuracy difference	0.14%	0.1%	0.7%	1.11%	0.15%

when the number of reference samples granulated increases to a large enough size.

We also give the effects of the number of reference granules on the top-5 accuracy. And the results are shown in Figs. 9 and 10. The top-5 accuracy also increases as k increases.

Some experiments are performed to illustrate the effect of random reference granules. We set the k value to 10. First we freeze all random granules. After that, we change the random granules one by one. The t denotes the number of random granules replaced. Table 3 shows the difference between the accuracy of the granular neural network after replacing the reference granules and the original granular neural network. For GResnet18, the difference in accuracy between the original network and the network replacing the reference granules is maximum of 1.79% when t = 4. The largest top5 accuracy margin is 1.37% for GResnet18. According to the experimental results in Table 3, random granulation has a limited effect on the granular neural network.

Table 4 describes some training settings and test set accuracies for GResnet34, GResnet18, Resnet34 and Resnet18 networks on Cifar100. All models were subjected to five replicate experiments, then the average accuracy data were obtained. The loss function of all networks is cross entropy, the optimizer of all networks is Adam, and the learning rate is 0.001. It can be seen from the table that when the network hidden layers are the same, GResnet18 correct rate is higher than Resnet18 and GResnet34 correct rate is higher than Resnet34. GResnet18 network with only 18 hidden layers is also higher than Resnet34 with 34 hidden layers in the test set correct rate. And the top5 accuracy

Table 4
Classification performances (*mean* ± *std*%) and settings of GResnet and Resnet in Cifar100.

Network	GResnet34	GResnet18	Resnet34	Resnet18
Number of hidden layers	34	18	34	18
Loss function	Cross entropy	Cross entropy	Cross entropy	Cross entropy
Optimizer	Adam	Adam	Adam	Adam
Learning Rate	0.001	0.001	0.001	0.001
Accuracy	**61.36** ± **0.63%**	60.28 ± 0.77%	54.54 ± 0.85%	54.34 ± 0.76%
Top5-accuracy	**85.57** ± **0.75%**	84.92 ± 0.64%	80.64 ± 0.72%	80.46 ± 0.72%
Top10-accuracy	**91.51** ± **0.79%**	91.01 ± 0.81%	88.36 ± 0.4%	88.33 ± 0.53%

Table 5
Classification performances (*mean* ± *std*%) and settings of granular residual network variant and residual network variants in Cifar100.

Network	GIResnet34	IResnet34	ReXnet	Resnext	Densenet	SeResnet
Loss function	Cross entropy	Cross entropy	Cross entropy	Cross entropy	Cross entropy	Cross entropy
Optimizer	Adam	Adam	Adam	Adam	Adam	Adam
Learning Rate	0.001	0.001	0.001	0.001	0.001	0.001
Accuracy	**64.51** ± **0.69%**	59.8 ± 0.64%	63.11 ± 0.52%	58.01 ± 1.15%	61.95 ± 1.28%	57.72 ± 1.05%
Top5-accuracy	**88.23** ± **0.74%**	84.65 ± 0.65%	87.65 ± 0.72%	83.35 ± 0.69%	86.41 ± 0.82%	83.81 ± 0.7%
Top10-accuracy	**93.31** ± **0.75%**	91.24 ± 0.48%	92.39 ± 0.75%	90.34 ± 0.59%	92.38 ± 0.66%	90.9 ± 0.47%

Table 6
Effects of random granules in Tiny.

Network	Difference value	t = 1	t = 2	t = 3	t = 4	t = 5
GResnet34	Accuracy difference	0.13%	0.61%	0.45%	0.87%	0.9%
GResnet34	Top5 accuracy difference	0.62%	1.34%	1.09%	1.51%	1.5%
GResnet50	Accuracy difference	0.04%	0.26%	0.66%	0.09%	0.28%
GResnet50	Top5 accuracy difference	0.06%	0.16%	0.49%	0.45%	0.19%

and top10 accuracy of GResnet18 are also greater than those of Resnet34 and Resnet18. There is little or no performance improvement in Resnet34 compared to Resnet18. GResnet34 has a noticeable performance improvement over GResnet18. Granular residual networks have better ability to restrain network degradation.

To further show the effectiveness of the granular neural network, we apply the granular neural network to the variant of the residual neural network. We propose the granular IResnet with reference to the network structure of IResnet [44]. And we compare the granular IResnet with the residual network variants granular IResnet [44], ReXnet [45], and Resnext [40], Densenet [46], SEnet [47] for the experiment. The variant of the granular residual network has better classification performance than the variants of the residual network with the same hyperparameters from Table 5.

4.2. Tiny imagenet

First we normalized the Tiny Imagene dataset. We also selected k samples as the reference set and granularized the sample set. We selected Resnet34 and Resnet50 for comparison with our proposed fuzzy granular neural networks, which are denoted as GResnet34 and GResnet50, respectively. The convolutional kernel size and all network optimizer settings are the same as in the Cifar100 dataset experiments.

Figs. 11 and 12 show the loss and accuracy curves for GResnet50. GResnet50 learns well on both training and test sets. Over-learning and under-learning do not appear.

For the GResnet34 network, the testing set correct rates corresponding to different numbers of reference granules are shown in Fig. 13. As reference granules number k increases, the correct rate gradually increases. When k increases to about 13 the correct rate stabilizes and the correct rate is around 48.5%. As shown in Fig. 14, the effect of k on the correct rates of GResnet50 is approximately the same as that of k for GResnet34. The accuracy is gradually converged to 50.7% as k increases.

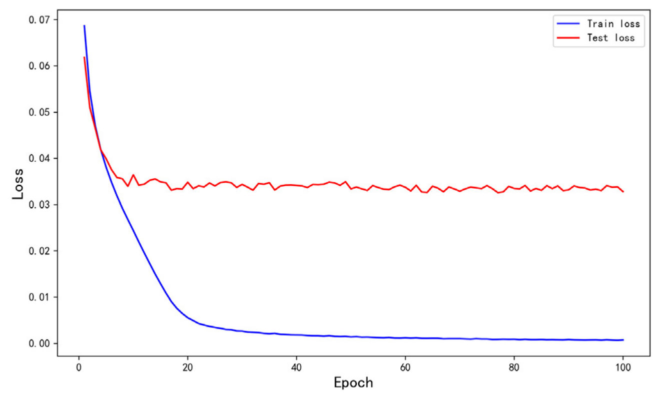

Fig. 11. The train loss and test loss of GResnet50 on Tiny Imagenet.

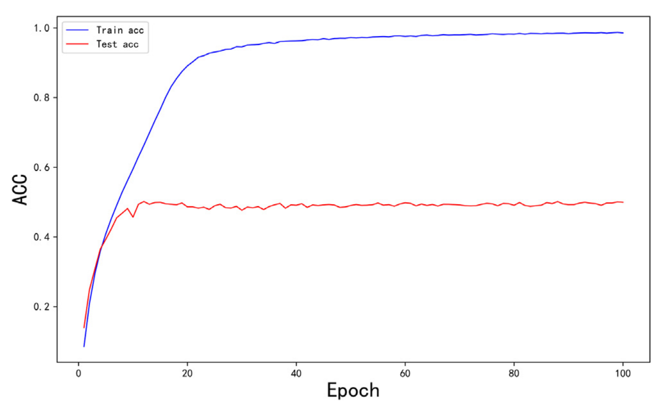

Fig. 12. The train accuracy and test accuracy of GResnet50 on Tiny Imagenet.

The effects of the number of reference granules on the top-5 accuracy are shown in Figs. 15 and 16. In GResnet34, the top-5 accuracy grows with increasing k. As k increases to 10, the top-5 accuracy tends to plateau in GResnet34. However, the top-5 accuracy keeps growing with k in GResnet50.

The experiment demonstrates to a certain extent that the performance of the granular neural network improves as the reference set increases and eventually reaches convergence. In terms of human cognition, we recognize our own position by constantly learning from others in comparison. It shows that granular neural

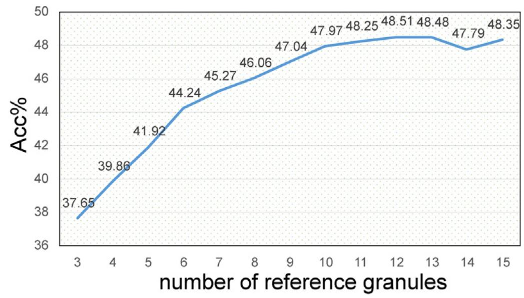

Fig. 13. Impacts of reference granules number k on accuracy in GResnet34.

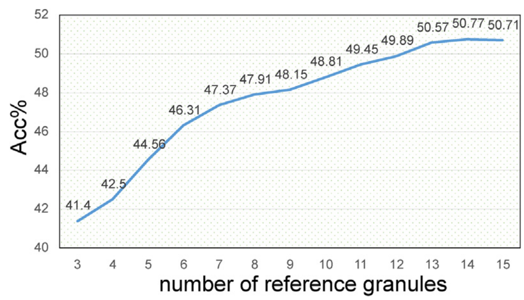

Fig. 14. Impacts of reference granules number k on accuracy in GResnet50.

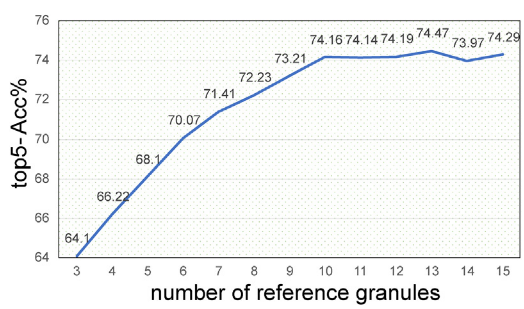

Fig. 15. Impacts of reference granules number k on top5-accuracy in GResnet34.

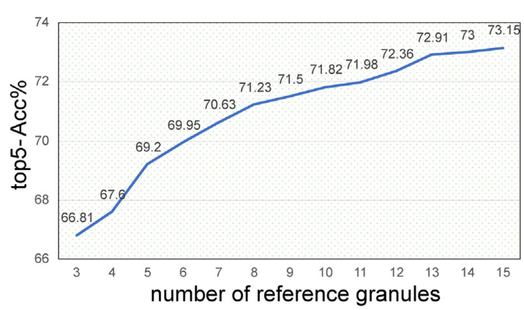

Fig. 16. Impacts of reference granules number k on top5-accuracy in GResnet50.

networks can also continuously strengthen themselves through incremental learning.

Similarly, Table 6 illustrates the effect of different random granules on GResnet34 and GResnet50 on the tiny dataset when $k = 10$. Both the accuracy tolerance and the top5 accuracy tolerance of GResnet50 are maximized when t = 2. For GResnet50, the maximum Top5 accuracy difference is less than 1%. Moreover, the maximum accuracy difference is only 0.66%.

Table 7 shows some training settings and test set accuracies for GResnet50, GResnet34, Resnet50 and Resnet34 networks on Tiny Imagenet. All models were run with 5 replicates. And averaged accuracy were obtained in Tables 7, 8. The granular residual neural network classification performance is substantially higher than that of the residual network with similar structures under the same number of layers. Even with fewer layers of granular residual neural network, its classification accuracy is higher in the Tiny Imagenet dataset. Although the accuracy of GResnet50 is greater than that of GResnet34, the top5 accuracy and top10 accuracy of GResnet34 are greater than those of GResnet50.

The granular neural network was likewise modified based on the residual network variant of IResnet50. We compare the modified granular neural network GIResnet50 with some variants of the residual network IResnet50 [44], ReXnet [45], Resnext [36], Densenet [46], SEnet [47] in Tiny Imagenet. The networks parameters settings and results are shown in Table 8. The proposed GIResnet50 has better classification performance both in terms of accuracy, top5 accuracy, top10 accuracy under the same hyperparameters.

Overall, the proposed granular neural network has better performance in both residual network and residual network variants under the same hyperparameters. Moreover, different reference granules can provide different perspectives, while traditional neural networks only learn information about the current sample. So the performance of the proposed granular neural network is continuously improving as the number of reference granules k increases.

5. Conclusions and discussions

In this paper, the fuzzy granular deep convolutional network with residual structures is proposed. The problem of excessive global granulation time complexity and space complexity is avoided by randomly selecting the reference system. Since granules are well-structured, it makes granular neurons naturally parallel. This algorithm constructs the granular neural network from the perspective of feature similarities of multiple samples, which is fundamentally different from traditional neural networks that focus only on the current sample. Neural network is a multi-granularity computational model, and granular neural network is a multi-granularity and multi-angle computational model. This multi-angle learning mode allows granular neural networks to have better generalization performance. Furthermore, granular neural network which has less hidden layers can achieve the functions of the ordinary neural network with more hidden layers. We also introduce the residual structure of neural network to deepen the granular neural network, which makes the granular neural network have better representation ability. We finally verify the feasibility and effectiveness of the residual granular neural network through experiments.

In the future, granular neural networks will be applied to more areas than just classification tasks. There are many directions for the improvement of granular neural networks. For example, replacing granulation with neighborhood granulation will bring different knowledge to granular neural networks. It is also an interesting work to combine granular neural networks with attention mechanisms. Neural network pruning techniques may also reduce the complexity of granular neural networks. And we will further build more complex and deeper granular networks too. Granular computing and deep learning would be better combined to solve a wide variety of complex problems. In terms of human cognition, we recognize our own position by constantly learning from others in comparison.

Table 7
Classification performances ($mean\% \pm std\%$) and settings of GResnet and Resnet in Tiny Imagenet.

Network	GResnet50	GResnet34	Resnet50	Resnet34
Number of hidden layers	50	34	50	34
Loss function	Cross entropy	Cross entropy	Cross entropy	Cross entropy
Optimizer	Adam	Adam	Adam	Adam
Learning Rate	0.001	0.001	0.001	0.001
Accuracy	**50.69** ± **0.81%**	48.57 ± 0.79%	48.13 ± 0.95%	43.47 ± 0.49%
Top5-accuracy	72.78 ± 0.66%	**75.14** ± **0.56%**	73.07 ± 0.46%	69.2 ± 0.48%
Top10-accuracy	81.1 ± 0.87%	**82.96** ± **0.5%**	81.35 ± 0.28%	78.23 ± 0.45%

Table 8
Classification performances ($mean\% \pm std\%$) and settings of Granular residual network variant and residual network variants in Tiny Imagenet.

Network	GIResnet50	IResnet50	ReXnet	Resnext	Densenet	SeResnet
Loss function	Cross entropy	Cross entropy	Cross entropy	Cross entropy	Cross entropy	Cross entropy
Optimizer	Adam	Adam	Adam	Adam	Adam	Adam
Learning Rate	0.001	0.001	0.001	0.001	0.001	0.001
Accuracy	**54.95** ± **0.53%**	53.99 ± 0.56%	48.12 ± 0.52%	50.23 ± 1.15%	50.13 ± 0.83%	49.14 ± 0.42%
Top5-accuracy	**78.81** ± **0.79%**	77.43 ± 0.23%	73.85 ± 0.74%	74.12 ± 0.87%	74.15 ± 0.62%	73.53 ± 0.18%
Top10-accuracy	**86.51** ± **0.55%**	84.93 ± 0.37%	81.39 ± 0.65%	82.25 ± 0.91%	82.29 ± 0.65%	81.79 ± 0.46%

CRediT authorship contribution statement

Linjie He: Conceptualization, Methodology, Validation, Formal analysis, Writing – original draft, Software, Project administration, Writing – review & editing. **Yumin Chen:** Methodology, Software, Visualization, Project administration, Writing – review & editing, Funding acquisition. **Keshou Wu:** Investigation, Supervision, Project administration, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61976183) and the Central Special Project of Local Development of Science and Technology (No. 2022L3029).

References

[1] L.A. Zadeh, Fuzzy Sets, Inf. Control 8 (3) (1965) 338–353.
[2] S.B. Roh, W. Pedrycz, T.C. Ahn, A design of granular fuzzy classifier, Expert Syst. Appl. 41 (15) (2014) 6786–6795.
[3] W. Pedrycz, H. Izakian, Cluster-centric fuzzy modeling, IEEE Trans. Fuzzy Syst. 22 (6) (2014) 1585–1597.
[4] J.R. Hobbs, Granularity, in: Proc. of the IJCAI (1985) 432–435.
[5] L.A. Zadeh, Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, Fuzzy Sets and Systems 90 (2) (1997) 111–127.
[6] W. Pedrycz, W. Homenda, Building the fundamentals of granular computing: A principle of justifiable granularity, Appl. Soft Comput. 13 (10) (2013) 4209–4218.
[7] T.Y. Lin, Data mining: Granular computing approach, Lecture Notes in Comput. Sci. 1574 (1999) 24–33.
[8] T.Y. Lin, L.A. Zadeh, Special issue on granular computing and data mining, Int. J. Intell. Syst. 19 (7) (2004) 565–566.
[9] Y.Y. Yao, Relational interpretations of neighborhood operators and rough set approximation operators, Inf. Sci 111 (1998) 239–259.
[10] Y.Y. Yao, Information granulation and rough set approximation, Int. J. Intell. Syst. 16 (1) (2001) 87–104.
[11] M. Duoqian, F. Shidong, The calculation of knowledge granulation and its application, Syst. Eng.-Theory Pract. 22 (1) (2002) 48–56.
[12] D. Miao, N. Zhang, X. Yue, Knowledge reduction in interval-valued information systems, in: 8th IEEE International Conference on Cognitive Informatics, 2009, pp. 320–327.
[13] R.R. Yager, Some learning paradigms for granular computing, in: IEEE International Conference on Granular Computing, 2006.
[14] Y.H. Qian, J.Y. Liang, C.Y. Dang, Incomplete multigranulation rough set, IEEE Trans. Syst. Man Cybern. A 40 (2010) 420–431.
[15] J. Qian, D.Q. Miao, Z.H. Zhang, et al., Parallel attribute reduction algorithms using MapReduce, Inf. Sci. 279 (2014) 671–690.
[16] M.A. Sanchez, O. Castillo, J.R. Castro, Information granule formation via the concept of uncertainty-based information with interval type-2 fuzzy sets representation and Takagi–Sugeno–Kang consequents optimized with Cuckoo search, Appl. Soft Comput. 27 (2015) 602–609.
[17] P. Melin, D. Sánchez, Multi-objective optimization for modular granular neural networks applied to pattern recognition, Inform. Sci. 460–461 (2018) 594–610.
[18] Y.M. Chen, N. Qin, W. Li, et al., Granule structures, distances and measures in neighborhood systems, Knowl.-Based Syst. 165 (2019) 268–281.
[19] D. Sánchez, P. Melin, O. Castillo, Comparison of particle swarm optimization variants with fuzzy dynamic parameter adaptation for modular granular neural networks for human recognition, J. Intell. Fuzzy Syst. 38 (3) (2020) 3229–3252.
[20] W. Li, Y.M. Chen, Y.P. Song, Boosted K-nearest neighbor classifiers based on fuzzy granules, Knowl.-Based Syst. 195 (2020) 105–606.
[21] G.P. Lin, J.Y. Liang, J.J. Li, A fuzzy multigranulation decision-theoretic approach to multi-source fuzzy information systems, Knowl.-Based Syst. 91 (2016) 102–113.
[22] T. He, W. Xu, Z. Lu, et al., Adaptive fuzzy logic energy management strategy based on reasonable SOC reference curve for online control of plug-in hybrid electric city bus, IEEE Trans. Intell. Transp. Syst. 19 (5) (2018) 1607–1617.
[23] Y.M. Chen, Q.X. Zhu, H.R. Xu, Finding rough set reducts with fish swarm algorithm, Knowl.-Based Syst. 81 (2015) 22–29.
[24] L.B. Cosme, W.M. Caminhas, M.F. Dangelo, R.M. Palhares, A novel fault-prognostic approach based on interacting multiple model filters and fuzzy systems, IEEE Trans. Ind. Electron 66 (1) (2018) 519–528.
[25] T. Lei, X. Jia, Y. Zhang, S. Liu, H. Meng, A.K. Nandi, Superpixel-based fast fuzzy C-means clustering for color image segmentation, IEEE Trans. Fuzzy Syst. 27 (9) (2019) 1753–1766.
[26] O. Castillo, et al., A generalized type-2 fuzzy granular approach with applications to aerospace, Inform. Sci. 354 (2016) 165–177.
[27] J.M. Mendel, P.P. Bonissone, Critical thinking about explainable AI (XAI) for rule-based fuzzy systems, IEEE Trans. Fuzzy Syst. 29 (12) (2021) 3579–3593.
[28] F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization in the brain, Psychol. Rev. 65 (6) (1958) 386.
[29] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back propagating errors, Nature 323 (1986) 533–536.
[30] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[31] Y. Lecun, L. Bottou, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[32] G.E. Hinton, S. Osindero, Y.W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2006) 1527–1554.
[33] Krizhevsky. Alex, Ilya. Sutskever, Geoffrey E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2012) 1097–1105.
[34] He. Kaiming, et al., Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.
[35] S. Wu, S. Zhong, Y. Liu, Deep residual learning for image steganalysis, Multimedia Tools Appl. 77 (9) (2018) 10437–10453.

[36] Tan. Runjie, et al., Color image demosaicking via deep residual learning, in: IEEE Int. Conf. Multimedia and Expo (ICME), vol. 2(4), 2017.
[37] Yixin Du, Li Xin, Recursive deep residual learning for single image dehazing, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018.
[38] C. Liu, X. Liu, D.W.K. Ng, J. Yuan, Deep residual learning for channel estimation in intelligent reflecting surface-assisted multi-user communications, IEEE Trans. Wirel. Commun. 21 (2) (2022) 898–912.
[39] H. Qiu, Q. Zheng, G. Memmi, J. Lu, B. Thuraisingham, Deep residual learning-based enhanced JPEG compression in the internet of things, IEEE Trans. Industr. Inform. 17 (3) (2021) 2124–2133.
[40] Xie. Saining, et al., Aggregated residual transformations for deep neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017.
[41] K. He, X. Zhang, S. Ren, S. Jian, Identity mappings in deep residual networks, in: European Conference on Computer Vision, Springer, Cham, 2016.
[42] Alex Krizhevsky, Geoffrey Hinton, et al., Learning multiple layers of features from tiny images, 2009.
[43] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al., Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015).
[44] I.C. Duta, et al., Improved Residual Networks for Image and Video Recognition, in: Proceedings of International Conference on Pattern Recognition, 2021.
[45] D. Han, S.Yun, B. Heo, Y.J. Yoo, Rethinking Channel Dimensions for Efficient Model Design, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2021.
[46] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely Connected Convolutional Networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017.
[47] J. Hu, L. Shen, S. Albanie, G. Sun, E. Wu, Squeeze-and-excitation networks, IEEE Trans. Pattern Anal. Mach. Intell. 42 (8) (2020) 2011–2023.