LETTER
A Development of Cascade Granular Neural Networks

Keun-Chang KWAK[†a)], *Member*

SUMMARY   This paper studies the design of Cascade Granular Neural Networks (CGNN) for human-centric systems. In contrast to typical rule-based systems encountered in fuzzy modeling, the proposed method consists of two-phase development for CGNN. First, we construct a Granular Neural Network (GNN) which could be treated as a preliminary design. Next, all modeling discrepancies are compensated by a second GNN with a collection of rules that become attached to the regions of the input space where the error is localized. These granular networks are constructed by building a collection of user-centric information granules through Context-based Fuzzy c-Means (CFCM) clustering. Finally, the experimental results on two examples reveal that the proposed approach shows good performance in comparison with the previous works.
key words: cascade granular neural network, human-centric system, information granules, context-based fuzzy c-means clustering

1.   Introduction

During the past few years, a considerable number of studies have been conducted on fuzzy logic and Granular Neural Network (GNN), together with a rapid growth in the variety of applications [1], [9]–[11]. In spite of this profound diversity existing in the area, all fuzzy models share a unified design viewpoint and rely on a single coherent methodological platform. On the other hand, GNN introduced in this paper uses Linguistic Model (LM) of complex systems consisting of information granules [2]. The purpose of this network is to reveal associations between fuzzy sets defined in input and output spaces. Furthermore, this network is designed by using fuzzy information granulation realized via Context-based Fuzzy c-Means (CFCM) clustering [3]–[7]. This clustering technique builds information granules in the form of fuzzy sets and develops clusters by preserving the homogeneity of the clustered patterns associated with the input and output space. In this paper, we develop the concept of Cascade Granular Neural Networks (CGNN) for human-centric systems as knowledge-based models. After adopting a construct of granular network as a preliminary network, all modeling discrepancies are compensated by a second granular network that becomes attached to the regions of the input space where the error is localized. The experimental results reveal that the proposed method yields a better performance in comparison with conventional GN, Radial Basis Function Networks (RBFN) based on CFCM clustering for Boston

housing data and automobile MPG (miles per gallon) prediction [5], [7].

2.   Granular Neural Networks (GNN)

For the design of the GNN, we consider the contexts to be described by triangular membership functions being distributed in the output space with the 1/2 overlap occurring between two successive fuzzy sets. The automatic generation of linguistic contexts is obtained by the output data density and probabilistic distribution [8]. We denote those fuzzy sets by W_1, W_2, ..., W_p. Let us recall that each context generates a number of induced clusters whose activation levels are afterwards summed up as shown in Fig. 1. The output type of the network denoted by $\xi_1, \xi_2, \ldots, \xi_p$ is granular. More specifically, assuming the triangular form of the contexts, triangular fuzzy number Y is expressed as

$$E = W_1 \otimes \xi_1 \oplus W_2 \otimes \xi_2 \oplus \ldots W_n \otimes \xi_n \tag{1}$$

We denote the algebraic operations by $\otimes$ and $\oplus$ to emphasize that the underlying computing operates on a collection of fuzzy numbers. As such, E is completely characterized by its three parameters that are the lower, modal value, and upper bounds as follows
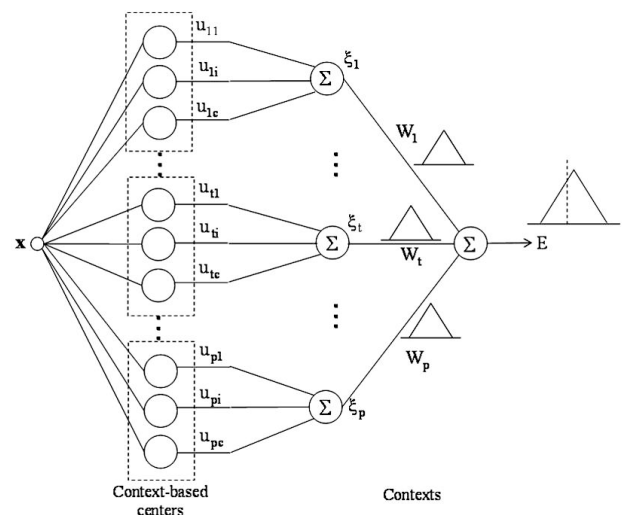

Fig. 1   Architecture of granular neural network.

$$e_{k-} = \sum_{t=1}^{p} W_{p-}\xi_p$$

$$e_k = \sum_{t=1}^{p} W_p\xi_p$$

$$e_{k+} = \sum_{t=1}^{p} W_{p+}\xi_p \tag{2}$$

For the k-th data point, x_k, we use the explicit notation $E(x_k) = <e_{k-}, e_k, e_{k+}>$ which helps emphasize the input-output relationship. The computations realized by the output neuron in Fig. 1 depend on the underlying formalism of granular computing realized there. The emergence of the network structure suggests that we should be able to eliminate possible systematic error and this could be easily accomplished by augmenting the summation node at the output layer by a numeric bias term w_0. This shifts the fuzzy set of output by the bias value. The bias term is computed in a straightforward manner so that it eliminates a potential systematic error

$$w_0 = \frac{1}{N} \sum_{k=1}^{N} (e_k - \tilde{e}_k) \tag{3}$$

where $\tilde{e}_k$ denotes a modal value of E produced for given input x_k.

The optimization completed by the CFCM clustering is realized iteratively by updating the partition matrix and the prototypes. The update of the partition matrix is completed as follows [3]

$$u_{tik} = \frac{w_{tk}}{\sum_{j=1}^{c} \left(\frac{\|x_k - v_i\|}{\|x_k - v_j\|} \right)^{\frac{2}{m-1}}} \tag{4}$$

where $i = 1, 2, \ldots, c, k = 1, 2, \ldots, N$

The u_{tik} represents the element of the partition matrix induced by the i-th cluster and k-th data in the t-th context. Here w_{tk} denotes a membership value of the k-th data to the t-th context. The cluster centers v_i are calculated in the form

$$v_i = \frac{\sum_{k=1}^{N} u_{tik}^m x_k}{\sum_{k=1}^{N} u_{tik}^m} \tag{5}$$

where the fuzzification factor "m" is generally taken as 2.0.

3. Cascade Granular Neural Networks (CGNN)

In this section, we cover the fundamental concept of the construction of the GNN with cascade structure. There are two essential phases: First, we construct a preliminary GNN which could be treated as a preliminary construction. Next, all modeling discrepancies are compensated by a second GNN with a collection of rules that become attached to the regions of the input space where the error is localized. The proposed clustering supporting the design of information granules is completed in the space of the input data associated with the error of the preliminary GNN while the
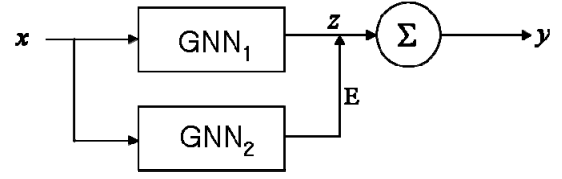

Fig. 2 The block diagram of cascade granular neural network.

clusters is guided by a collection of some pre-defined fuzzy sets defined in the space of error. By taking into account the contexts, the clustering in the input space is focused by some predefined fuzzy sets of contexts. This helps reveal the relationships between the regions of the input space and the associated error and naturally leads to the formation of some web of connection between the information granules defined in the error space and constructed in the input space.

The experimental data under discussion are the pairs of the n-dimensional input-output data sets. They come in the following form $\{x(k), \text{target}(k)\}$ $k = 1, 2, \ldots, N$, where $x(k) \in R^n$ and $\text{target}(k) \in R$. The enhancement of the network at which the granular part comes into the play is based on the transformed data $\{x(k), e(k)\}$ where the residual part manifests through the expression $e(k) = \text{target} - z(k)$ which denotes the error of the preliminary GNN. In the sequel, those data pairs are used to develop a second GNN as the cascade rule-based part of the network. Here this rule-based augmentation of the second network associates input data set with the error produced by the preliminary GNN in the form of the rules "if input then error". The rules and the information granules are constructed by means of CFCM clustering. Figure 2 shows the system modeling by combining preliminary granular network (GNN_1) and second granular network (GNN_2).

4. Experimental Results

First, we shall use the Boston Housing data as an example. This data set concerns prices of real estate in the Boston area. The MEDV (median value of the price of the house) depends on 13 continuous attributes and 1 binary attribute. The data set consists of 506 examples. The training and testing data set are randomly selected by 60 %–40 % split in the normalized space between 0 and 1, respectively. The experiment is performed by 20 iterations.

The training data set is used for model construction, while the testing set is used for model validation. Thus, the resultant model is not biased toward the training data set and it is likely to have a better generalization capacity to new data. Figure 3 shows the linguistic contexts produced by probabilistic distribution of localized error. Here, the error between the actual output and network output in Fig. 3 is obtained after performing preliminary granular network on training data set. This linguistic context is generated by histogram, probability density function, and conditional density function in alphabetical order. For further details on the automatic generation of linguistic contexts,
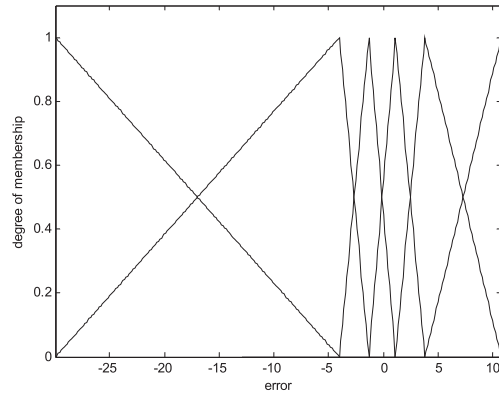
Fig. 3 Linguistic contexts obtained by localized error.

see [8]. As shown in Fig. 3, the linguistic contexts are produced by output error density when p=6. Figure 4 displays the output of the CGNN and actual output for training and testing data, respectively. As shown in Fig. 4, it is obvious that the proposed CGNN has good prediction performance. Table 1 lists the RMSE (root mean square error) results regarding approximation and generalization capability, respectively. Here Train_RMSE and Test_RMSE represent RMSE for training and testing data, respectively.

In the design of GNN, we used six contexts and clusters (p=c=6) in each context for CFCM clustering. Although the GNN has a structured knowledge representation in the form of fuzzy if-then rules, it lacked the adaptability to deal with nonlinear model as listed in Table 1. Moreover, we constructed the RBFN based on six contexts and clusters in the same manner. Here learning rate is 0.0001 and the number of epoch is 1000. As listed in Table 1, the experimental results revealed that the proposed networks yielded a better performance in comparison with the GNN and RBFN-CFCM.

Next, we shall use the well-known automobile MPG (miles per gallon) data as a nonlinear regression example. In this example, 6 input variables consist of cylinder number, displacement, horsepower, weight, acceleration, and model year. The output variable to be predicted in terms of the preceding six input variables is the automobile's fuel consumption in MPG. The data set consists of 392 examples of different car makes after removing instances with missing values. In a similar fashion, the training and testing data set are randomly selected by 60 %–40 % split. The experiment is performed by 20 iterations. Figure 5 and Table 2 represent the experimental results regarding approximation and generalization capability obtained by training and testing data, respectively. As listed in Table 2, the experimental results revealed that the proposed approach yielded a better performance in comparison with the conventional granular network and RBFN based on context-based fuzzy clustering.

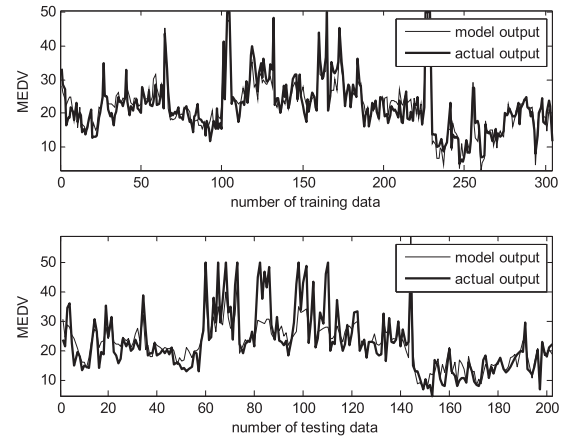**Fig. 4** Predication performance of CGNN.

Table 1 Performance comparison of the proposed method.

	Train_RMSE	Test_RMSE
GNN [4]	4.96	6.11
RBFN-CFCM [6]	5.52	6.91
The proposed method	3.38	4.84
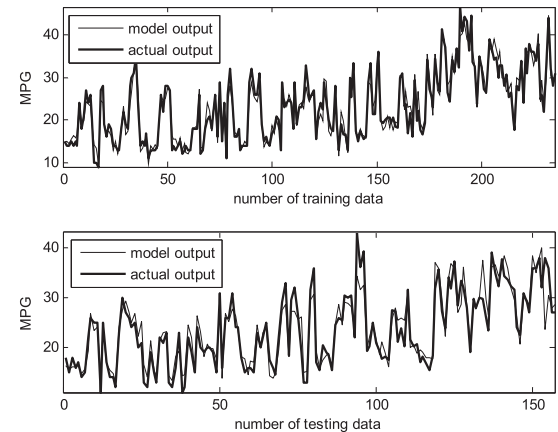
Fig. 5 Predication performance of CGNN.

Table 2 Performance comparison of the proposed method.

	Train_RMSE	Test_RMSE
GNN [4]	2.80	3.32
RBFN-CFCM [6]	3.21	3.51
The proposed method	2.10	3.19

5. Conclusions

We have proposed the cascade design methodology of granular neural network. This methodology is quite different from the one commonly pursued in the realm of fuzzy models with the predominant concept of rule-based architectures. One could view this design methodology as a way of augmenting a preliminary granular neural network by a

collection of local rule-based models. The experimental results on two examples revealed that the proposed CGNN showed a better performance and effectiveness in comparison with the previous literatures. We can recognize that the well-defined semantics of the information granules used in system modeling is essential when designing a user-centric system.

Acknowledgments

References

[1] J.S.R. Jang, C.T. Sun, and E. Mizutani, Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, Prentice Hall, 1997.

[2] W. Pedrycz, A. Skowron, and V. Kreinovich, Handbook of Granular Computing, John Wiley & Sons, 2008.

[3] W. Pedrycz, "Conditional fuzzy c-means," Pattern Recognit. Lett., vol.17, pp.625–632, 1996.

[4] W. Pedrycz and A.V. Vasilakos, "Linguistic models and linguistic modeling," IEEE Trans. Syst. Man Cybern. C, Appl. Rev., vol.29, no.6, pp.745–757, 1999.

[5] W. Pedrycz and K.C. Kwak, "Linguistic models as framework of user-centric system modeling," IEEE Trans. Syst. Man Cybern. A, Syst. Humans, vol.36, no.4, pp.727–745, 2006.

[6] W. Pedrycz, "Conditional fuzzy clustering in the design of radial basis function neural networks," IEEE Trans. Neural Netw., vol.9, no.4, pp.745–757, 1999.

[7] W. Pedrycz and K.C. Kwak, "The development of incremental models," IEEE Trans. Fuzzy Syst., vol.15, no.3, pp.507–518, 2007.

[8] K.C. Kwak and D.H. Kim, "TSK-based linguistic fuzzy model with uncertain model output," IEICE Trans. Inf. & Syst., vol.E89-D, no.12, pp.2919–2923, Dec. 2006.

[9] A.V. Nandedkar and P.K. Biswas, "A granular reflex fuzzy min-max neural network for classification," IEEE Trans. Neural Netw., vol.20, no.7, pp.1117–1134. 2009.

[10] H.S. Park, W. Pedrycz, and S.K. Oh, "Granular neural networks and their development through context-based clustering and adjustable dimensionality of receptive fields," IEEE Trans. Neural Netw., vol.20, no.10, pp.1604–1616, 2009.

[11] Y.Q. Zhang, B. Jin, and Y. Tang, "Granular neural networks with evolutionary interval learning," IEEE Trans. Fuzzy Syst., vol.16, no.2, pp.309–319, 2008.