Analysis and Comparison of the Structure and Performance of Local Neural Networks

Shuang Cong*
Department of Automation
University of Science and Technology of China
Hefei, China

*Corresponding Author: scong@mail.ustc.edu.cn

Kezhi Li
Institute of Health informatio
University College London
222 Euston Rd, London, NW12DA, UK
Ken.li@ucl.ac.uk

***Abstract*—The paper synthesizes the local neural networks. Network structures and their activation functions of three local networks CMAC, B-spline, RBF that are often used to approach functions are analyzed and compared in detail. The network structure of ART-2 is also discussed. Based on the fuzzy system of these local networks, the paper depicts their fuzzy structures and performances. The study and analysis in the paper are useful to instruct to select and design the local neural networks.**

***Keywords*—local neural network; CMAC, B-spline; ART-2; fuzzy-neural network.**

I. INTRODUCTION

Since the eighties of the 20^{th} century, great progress has been made in artificial neural network (ANN) theory. Neural networks have been used for the non-linear modeling and design of controller, pattern classification and recognition, associative memory, computing optimization and so on. From general structure classification, ANN can be divided into feed-forward networks and feedback networks. BP network is the typical representative of the former and Hopfield network is that of the later. However, from the connection mode classification, artificial neural networks can also be divided into the global neural networks and local neural networks. In global neural networks, every output of networks is influenced by all connection weights. BP network that is used widely in control field is the typical one of this network. Though being the merit of global approximation to function, the BP network has relatively slow learning speed. However, if only part or even one weight is influenced by each of inputs/outputs, such networks are called 'local neural networks (LNN)'. Owning to their learning method, these networks have the faster learning speed, which is important to real-time control. Therefore, these networks are extensively used in this control field. LNN are in a very broad category. Generally speaking, they can be divided into two classes according to their fundamental functions. The one is mainly including in CMAC, B-spline and RBF networks. Approximation of function and systematic modeling are their main functions. And ART-2 and Adaptive Competitive Network is belonging to another class. They are always used for pattern classification and recognition.

The purpose of this paper is to analyze and compare the structure and performance of local neural networks. Firstly, three neural networks that are used to approximate function such as CMAC, B-spline and RBF are analyzed and compared. Then the ART-2 neural network that is used to pattern recognition is analyzed. At last, discussion about the relationship between local neural network and fuzzy network system are forwarded. The generalization and difference about local neural networks are also presented in this paper.

The paper is organized as follows. Local neural network for approximation of function is studied in detail in Sec. II, which includes the similarities among CMAC, B-spline and RBF neural networks; the differences among CMAC, B-spline and RBF neural networks. Local network with function of automatic classification is in Sec. III. Relation between local neural network and fuzzy neural system is in Sec. IV. Finally, Sec. V is the conclusion.

II. LOCAL NEURAL NETWORK FOR APPROXIMATION OF FUNCTION

A. Similarities among CMAC, B-spline and RBF Neural Networks

CMAC, B-spline and RBF have generally similar structure [1] . The kind of network with three layers is shown in Fig. 1, in which q groups of r-dimension input-vectors $\overline{X}_l = [x_1, x_2, x_3, ..., x_r]$ (l = 1, 2, …, q) are inputted into certain primary function in the hidden layer which has m nodes. Then the corresponding outputs of each node in the hidden layer are multiplied with each weight. Finally the s-dimension output of neural network is obtained at the output layer.

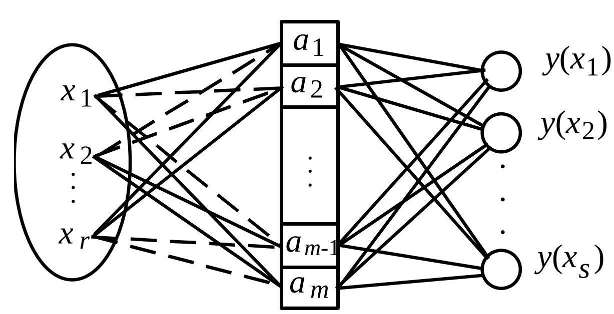


Fig. 1 Basic structure of local neural networks

The output of this local neural network is

$$y_k = \sum_{j=1}^{m} w_{kj}\alpha_j(x_i)\ ,\ k = 1, \ldots, s;\ i = 1, \ldots, r \qquad (1)$$

where y_k is the output of k-th node, $\alpha_j(x)$ is the certain primary function. w_{kj} is the connection weight between j-th neuron of hidden layer and i-th neuron of output layer.

In Fig. 1, each neuron of output layer is influenced by all neurons of hidden layer. However, by choosing certain primary function, most of output values of nodes can become zero. Thus, as to random input, the output can be gotten by weight summation of outputs of little non-zero nodes. The network actually is locally connected.

B. Differences Among CMAC, B-spline and RBF Neural Networks

1) Squire Primary Function of CMAC

CMAC (Cerebellar Model Articulation Controller) is put forward by J. Albus in 1975 [1]. This network simulates the connection structure of human cerebellum and is based on the table inquiry output/input technology. Its primary structure is shown in Fig. 1 and composed of input layer, hidden layer and output layer. The linear mapping is between hidden layer and output layer. And variable parameter w_{kj} (k = 1, 2, …, s ; j = 1, 2, …, m) is connection weight. Other mapping is between output layer and hidden layer of CMAC, which is similar with look-up structure and defined when the network begins to work. Its function is to map input vectors in input layer to hidden layer according to their distance of each other. Hidden layer of CMAC is composed of a group of quantized perceptions. Each of the vectors in input layer is corresponded to C perceptions in the hidden layer. Number C is important to CMAC and called perceptive field. Each input vector can only affect C perceptions and make them output 1 and other zero. Thus, the corresponding output of each input is only affected by such C perceptions. So we can see that the CMAC adopts a kind of simple square primary function that is show in following equation.

$$\alpha_j(x_i) = \begin{cases} 1, & j \in \phi \\ 0, & j \notin \phi \end{cases} \qquad (2)$$

where $\alpha_j(x_i)$ presents the primary function that is corresponded to j-th perception. And ϕ presents the set of C perceptions that are corresponded to i-th input vector x. When being trained, the network can only adjust local connection weight of output layer. Compared with other local connection neural networks, the CMAC is faster in studying and quite suitable to adaptive modeling and control. on the other hand, using simple squire primary function, the CMAC can only approach smooth function. Thus, its precision of approaching is not good enough. Through adding the value of C can improve the precision, but storage space has to be increased accordingly. This is the limitation of CMAC network. However, for its simpleness, the CMAC is very suitable for real-time application.

2) B-spline Neural Network

B spline is the abbreviation of basic spline. Different from CMAC network, B-spline primary function has the smallest local support among all of the spline functions. B-spline neural network can fit the given input/output values by using piecewise polynomial interpolation. Single variable B-spline function consists of a group of B-spline primary functions, and multivariable B-spline function is composed of several groups of B-spline primary functions. B-spline primary function is satisfied with following recurrence relation [2]:

$$B_{i,0}(x) = \begin{cases} 1, x_i < x \le x_{i+1} \\ 0,\ other \end{cases} \qquad (3)$$

$$B_{i,k}(x) = \frac{x - x_i}{x_{i+k} - x_i} \cdot B_{i,k-1}(x) + \frac{x_{i+k+1} - x}{x_{i+k+1} - x_{i+1}} \cdot B_{i+1,k-1}(x) \qquad k = 1, 2, \ldots m \qquad (4)$$

where, x_i is sequence of nodes, k is order of B-spline.

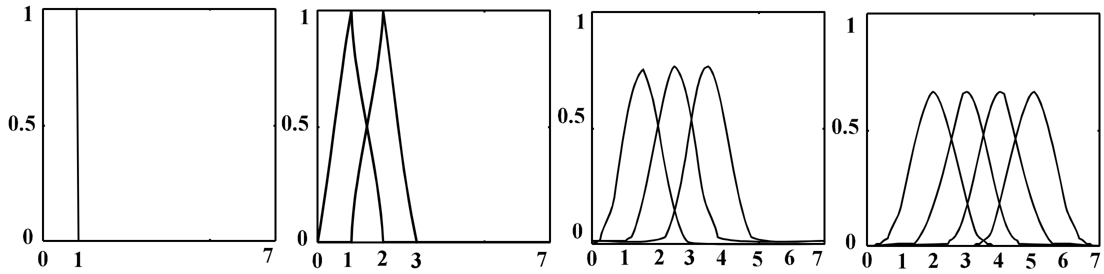
From (4) one can see that one k order B-spline function is obtained by fitting the polynomial curve of the lower order k-1, and the B-spline function is non-negative and locally supported, i.e.

$$B_{i,k}(x) \begin{cases} > 0, & x \in [x_i, x_{i+m}) \\ = 0, & other \end{cases} \qquad (5)$$

and they form a single segmentation, that is:

$$\sum_{i-1}^{N} B_{i,k}(x) = 1, \quad x \in [x_m, x_{N+1}) \qquad (6)$$

The single variate B-spline basis function curves at polynomial order m of 0, 1, 2 and 3 are shown in Fig. 2.



(a) $m = 0$ (b) $m = 1$ (c) $m = 2$ (d) $m = 3$

Fig. 2 Graph of B-spline basis function when order *m* is 0, 1, 2 and 3.

It can be seen from Fig. 2 that the order of B-spline is always one order higher than the order of polynomial curve used to represent this order of B-spline. For example, the first-order B-spline is represented by a straight line of order $m = 0$; The second-order B-spline is represented by a slash of order $m = 1$. For multivariable B-splines, the dimensional input vector can be assumed that $x = [x_1 \quad x_2 \quad \cdots x_n]^T$, and a univariate B-spline basis function is defined on each input axis as $B_{i,m_i}^{k_i}(x_i), k_i = 1,2,\cdots,m_i, i = 1,2,\cdots n$, then the multivariable B-spline basis function $M_k(x)$ is composed of the product (tensor product) of *n* univariate basis functions:

$$M_k(x) = B_{1,m_1}^{k_1}(x_1) B_{2,m_2}^{k_2}(x_2) \cdots B_{n,m_n}^{k_n}(x_n) = \prod_{i=1}^{n} B_{i,m_i}^{k_i}(x_i) \tag{7}$$

where the summer of $M_k(x)$ is $p = m_1 m_2 \cdots m_n$, so $k = 1,2,\cdots p$.

In the formula above, B-spline primary function in B-spline neural network is more precise than that of CMAC. Also, it has been proved that B-spline neural network can approach to any given nonlinear function unlimitedly by using polynomial interpolation theory. Thus B-spline network is a kind of universal approaching tool. However, since B-spline only uses polynomial interpolation to approach given input/output points, the curve for approximation is not smooth. Thus, the generalization ability is not gotten improved. The application of B-spline neural network is limited for this weakness.

3) *RBF Neural Network*

The full name of RBF is radial based function neural network. This network use Gaussian primary function as the mapping between output layer and hidden layer. Gaussian primary function is show as following [4]:

$$a_j(x) = e^{-\frac{\|X - c_j^2\|}{\delta_j^2}} \tag{8}$$

where c_j is the center point of *j*-th primary function. δ_j is called extension constants and indicates the dimension corresponding to the distance between *X* and c_j.

What should be pointed out is that the weight algorithm of RBF neural network is functioned layer-by-layer. And the weight of hidden layer is equivalent to the input vector. Thus, though it seems to be global connective in structure, the RBF neural network is actually local neural network. As to a given input, only one neuron of network is active. Outputs of other neurons can be neglected. Thus, RBF neural network is a kind of local neural networks.

Compared with the B-spline function, Gaussian primary function is smoother. And its derived number of any orders is existent. Thus, RBF neural network not only has excellent approximation ability but also has ability of generalization. At the same time, Gaussian primary function is relatively simple in format and suitable to be used in analyzing the multivariable system with good analyticity. Thus among the all nonlinear control algorithms based on neural network, the adaptive algorithm based on RBF neural network is the most precise and is suitable to analyze. What's more, RBF neural network can identify nonlinear system relatively easily without complex calculation. However, compared with the CMAC [5] and B-spline neural networks, RBF needs adjust more connection weights when trained. Therefore, the speed of RBF network is relatively slower for its amount of calculation.

III. Local Network with Function of Automatic Classification

In the procession of training sample data, it has been noticed that the network will forget part or all of old data no matter to be trained with supervise or not. When supervised trained, the network will gradually format stable memory after sample data are inputted thousands upon thousands times again and again. When trained without supervised, the old data will gradually be updated and forgotten by new ones. Even in certain condition, all vectors recorded will fluctuate and become no any mean. Thus, the fact of forgetting old data when studying new one is a big problem existed in the networks. It is hoped that the network can learn new knowledge without any negative effect to original memory. However, it's very difficult to realize this goal when lots of vectors are inputted. Generally speaking, the compromise has to be made for network to accept more new knowledge and affect little memory.

Adaptive Resonance Theory (ART) network can solve above-mentioned problem [6]. The network and algorithm can adapt the new input patterns and void to destroy the learned patterns according to its own requirement. By adding the memory capacity with the increase of the sorts of samples, the network can learn the new sample without destroying the old knowledge. ART neural network is put forward by Grossberg and Carpenter in Adaptive System Center of Department of Mathematics at Boston University. The earliest ART-1 (see Fig. 3) only handled binary input patterns. Then appeared the ART-2 (see Fig. 4) model that is based on and improved ART-1 model. Different with ART-1, ART-2 [7] network can classify any kind of patterns. Adaptive Competitive Network has the similar function of classify patterns to ART neural network. However as to former, the number of sorts has to be set in advance. ART-2 neural network can classify given patterns automatically without supervised by different given restrict constant. Thus ART neural network have obvious merits not only in the theory but also in application.

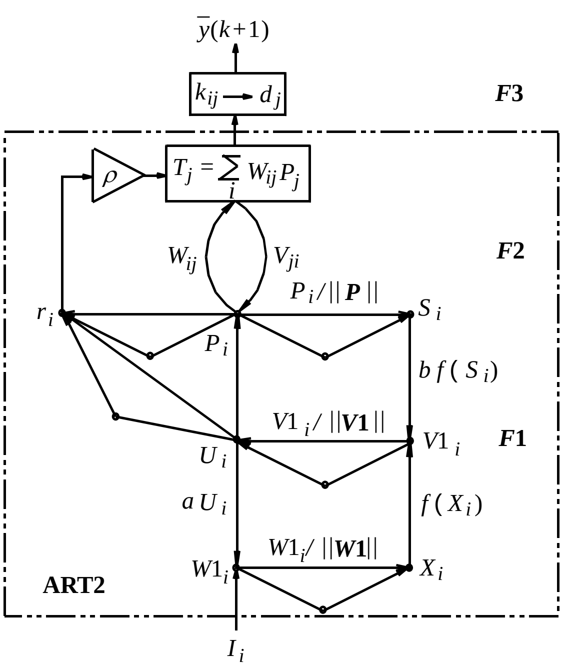


Fig. 3 One type of ART-1 contracture

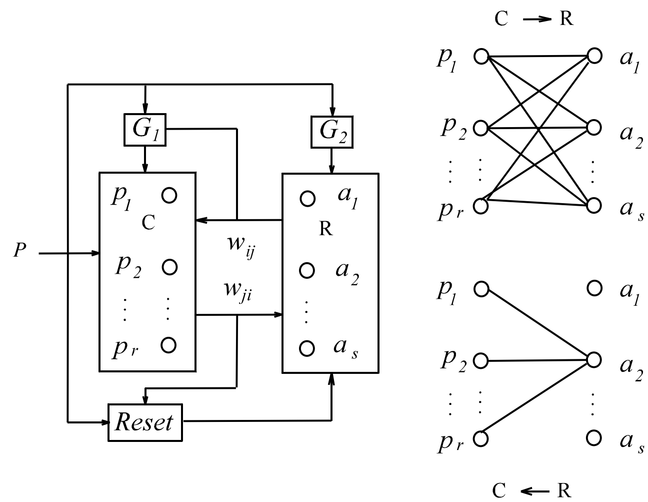


Fig. 4 One type of ART-2 contracture

Besides this, ART-2 network can also be used in function approximation. However, its algorithm and performance is different with other local neural networks mentioned above. An additional output layer has to be adding in ART-2. In the algorithm, local connection network maps the given pattern into several neurons through primary function. However, ART-2 neural network classify the patterns definitely. According to the winner-take-all principle. The input pattern only resonates with one neuron that is the most similar with the input pattern. And only the connection weight of this neuron is modified. In the aspect of network training, an important parameter of network is reset constant. The more is the reset value, the stricter is made the classification. When ART-2 is used to approach functions, the guard value is often set to be close to 1. Otherwise, the approach precision will be not good enough. At the same time, the number of neuron in hidden layer will increase with the number of samples. This is the weakness of the ART-2 used to function approximation.

IV. Relation Between Local Neural Network and Fuzzy Neural System

The fuzzy neural system is a kind of neural network combined with fuzzy logic. Generally, it can be divided into four layers. The typical two-input fuzzy logic neural system are shown in Fig. 5.

The first layer is the input layer which accepts and transmits vector *X*. The second layer can get membership grade of the sub-vector as to each verbal variable through the membership grade function. Each neuron in third layer represents one fuzzy ruler, which can be used to map the first component of the fuzzy ruler and get the relevance weight about each ruler. The fourth layer realizes the normalization and clear calculation and gets the relational expression between input and output as:

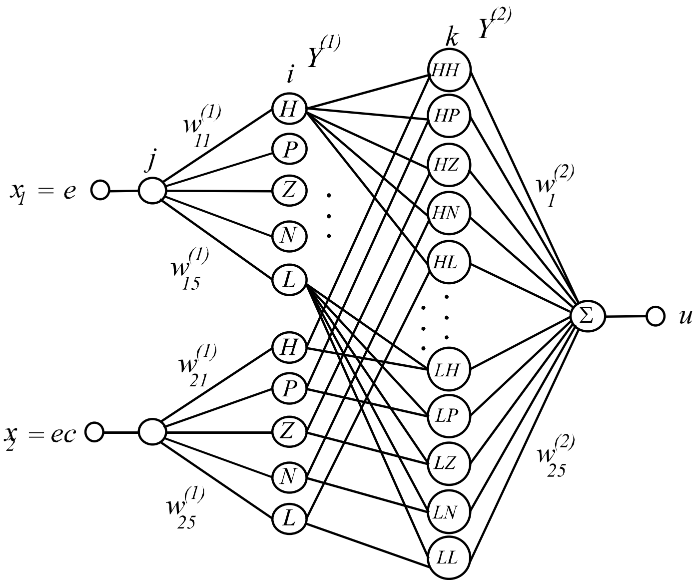


Fig. 5 The structure of the fuzzy neural network

$$y(x) = \frac{\sum_{j=1}^{m} \mu_j(x) y_k^2(x)}{\sum_{j=1}^{m} \mu_j(x)}$$

$$= H \cdot H \cdot W_1^{(2)} + H \cdot P \cdot W_2^{(2)} + \cdots + L \cdot N \cdot W_{24}^{(2)} + L \cdot L \cdot W_{25}^{(2)} \tag{9}$$

Compared with the fuzzy neural system, the local neural networks used in function approximation have the similar structure and input-output relational expression. Thus, through converting the primary function into the corresponding membership function, the local neural networks can become the corresponding fuzzy neural system. The ordinary CMAC neural network have only two states of '0', '1' in the mapping area of hidden layer. By using fuzzy neural system structure, the network can expand the states into real number field and improve its ability of generalization. Membership function $\mu_j(x)$ is chosen according to the condition and generally in proportion to the Euclidean number. The final output of fuzzy CMAC is:

$$y_k = \sum_{j=1}^{m} w_{ij} \frac{\mu_j}{\sum_{j=1}^{m} \mu_j} \quad i = 1, 2, \ldots, r \tag{10}$$

$\mu_j \propto \left\| X - C_j \right\|$, C_j is the center of No. j fuzzy subset.

According to (7) one can obtain that multi-variable B-spline function is the linear combination of several single-variable B-spline primary functions.

$$y(x) \approx \sum_{k=1}^{p} M_k(x) w_k = \sum_{k=1}^{p} \prod_{i=1}^{n} B_{i,m_i}^{k_i}(x_i) w_k \tag{11}$$

Through careful observation, it can be found that the order *k* of the B-spline basis function corresponds to the fuzzy membership functions of different shapes. For example, when $k = 2$, the curve made by B-spline just represents the triangular membership function. When k = 3, the curve is similar to the quadratic membership function. On the other hand, the number of vertices *N* corresponds to the number of fuzzy markers. The number of nodes determines the width of a membership function in a certain range. So a multi-variable B-spline function is an input/output relational expression of fuzzy neural system, which is made up by reasoning of sum product. It means that the corresponding relation exists between fuzzy neural system and multi-variable B-spline networks in structure. In the above formulae about multi-variable B-spline, $B_{i,m_i}^{k_i}(x_i), k_i = 1,2,\cdots,m_i, i = 1,2,\cdots n$ corresponds to the value of fuzzy membership function. $\prod_{i=1}^{n} B_{i,m_i}^{k_i}(x_i) w_k$ corresponds to the product ruler. The procession of inverse fuzzy is achieved by summation.

Thus, B-spline neural network and fuzzy neural system in the structures are equivalent on the ability to processing information. The transparency of B-spline network is the valuable characteristic and not existed in other general neural network. That is because that it allows designer to initialize network with natural nomenclatures which are used to explain its behavior by experts. The parameters of network also can be optimized by optimum algorithm. By using the polynomial calculation, the network avoids the complex calculation for nonlinear exponential active transfer function and is very timesaving and simple to design.

In the applications, Gaussian primary function in RBF (GRBF) is more often used as membership function of fuzzy neural system [8]. Its concrete form is:

$$a_i = e^{-\frac{\left\| W_i - X \right\|^2}{C_i^2}} \tag{12}$$

where, C_i is the width of membership function.

In general application, The GRBF network is more often used. Compared with the general RBF network, GRBF has an additional procession to get weighted mean. It improves the performance of network property. GRBF not only has good ability of function approximation, but also is smoother than general RBF. At the same time, the output of network is as followed [9]:

$$y(x)=\frac{\sum a_i(x)\omega_i}{\sum a_i(x)} \tag{13}$$

where, a_i is the output of hidden layer. Its value is belonging to 0~1 output of the exponential type function.

In formula (13), we can find that its form is almost totally the same with the output form of fuzzy system in formula (9), which can be solved by barycentric method. The only difference is that the GRBF uses the RBF function instead of the membership function existed in fuzzy system. As to the local input, GRBF can have a response of barycentric weight in the small receiving field through RBF function. This response is the same as the fuzzy system. Thus, through the two systems are developed based on different theories, they all achieve the same effect.

Fuzzy ART-2 neural network always uses the fuzzy C-mean clustering method. The network not only searches the neuron that is nearest to the input pattern, but also calculates all Euclidean distances between neurons and input pattern. Then the network uses the following formula to get the membership grade of input vector to each of neurons.

$$\mu_{ik}(x)=\begin{cases}\dfrac{1}{\sum_{j=1}^{c}\left(\dfrac{d_{ik}}{d_{jk}}\right)^{\frac{2}{m-1}}}, & I_k=\phi \\ 0, & \forall i\in\tilde{I}_k \\ 1, & \forall i\in I\end{cases} \tag{14}$$

where, $I_k=\{i\,|\,1\le i\le c, d_{ik}=0\}$, $d_{ik}=\|X_k-C_i\|$, X_k is the *k-th* input vector, C_i is the clustering center of *i-th* neuron.

From the above analyses about two kinds of different networks, we can see that, as to the local network to approach function such CMAC, B-spline and RBF networks, the fuzzy structure is implied in them. Though adding the fuzzy logic, they can be easily rebuilt to fuzzy neural network. These fuzzy neural systems not only have faster study speed and better adaptive capacity as the original neural networks, but also are provided with the characteristic of logic and transparency as the fuzzy systems. What's more, it is easy to set the initial parameters of networks through former known expert knowledge. Thus, combined with the fuzzy logic, this kind of neural networks for approaching function have widely used in the system control.

V. Conclusion

This paper analyzed the CMAC, B-spline and RBF local neural networks used to approach functions in detail. The merits and drawbacks of these networks were further pointed. By analyzing of three local neural networks mentioned, the approximation precision and centralization of CMAC, B-spline, RBF is better in turn. However, with the improvement of precision, they need more amounts of calculation and storage space. At the same time, the viewpoint of structure-algorithm separation is presented, which will be a good preparation to develop new networks. ART-2 network has the best performance to identify patterns among all adaptive networks. This network can learn and classify the patterns automatically and even can be used to approach functions after improvement. The fuzzy logic can be introduced into these local networks, which makes these networks more transparent in structure and more different in algorithm. Thus they should be chosen and applied according to given conditions. More applications of the different neural networks can be found in [10].

VI. Acknowledgment

This work was supported by the National Natural Science Foundation of China under grants no. 61973290.

References

[1] J. S. Albus, "Data Storage in the Cerebellar Model Articulation Controller (CMAC)," Transaction of ASME. J. Dynam. Syst. Meas. Control, vol. 97, pp. 228-233, 1975.

[2] S. Cong and G.D. Li, "The Design of Adaptive B-spline Fuzzy Neural Network Controller," Computer Engineering and Applications, vol. 35, pp. 66-68, 1999.

[3] D. T. Pham and M. F. Sukkar, "Supervised Adaptive Resonance Theory Nerual Network for Modelling Dynamic System," Systems, Man and Cyb., 1995. Intelligent Systems for the 21st Century. IEEE International Conf. on, Canada, pp. 2500-2505, 1995.

[4] P. D. Wasserman, "Advanced Methods in Neural computing", New York, Van Norstrand Reinhold, 1993.

[5] P. Duan, and H. Shao, "CMAC Neural Compute and Neural Control, Information and Control, vol. 28, pp. 197-207, 1999.

[6] S. Cong, "Comparison of Several Fuzzy Neural Network Systems (in Chinese)," Inform. and Control, vol. 30, pp. 486-491, 536, 2001.

[7] S. Cong, "The Function Analysis and Application Study of Radial Basis Function Network," Comp. Engineering and Applications, vol. 38, pp. 85-87, 200, 2002.

[8] X. Wang and H. Shao, "RBF neural network theory and its Application in Control, Inform. and Control," vol. 26, pp. 272-280, 1997.

[9] S. Cong, "Intelligent Control Systems and Their Applications (Second Edition)," Uni. of Sci. & Tech. of China Press, Hefei, Aug. 2021 (in Chinese)

[10] Qunchao Yu, Weiwei Shang, Zengzhi Zhao, Shuang Cong, and Yunjiang Lou, "Robotic grasping of novel objects from RGB-D Images by Using Multi-level Convolutional Neural Networks," IEEE Interna. Conf. on Inform. and Automa., Fujian, China, pp. 341-346, Aug. 2018.