2020 5th International Conference on Advanced
Robotics and Mechatronics (ICARM)

Hand Gesture Recognition Based on Multi-Classification Adaptive Neuro-Fuzzy Inference System and pMMG

Lei Wang, Jian Huang, Dongrui Wu, Tao Duan, Rui Zong, Shicong Jiang

Abstract— **In this paper, a multi-classification adaptive neuro-fuzzy inference system combining neural-network and a TSK fuzzy system is proposed to recognize six commonly used gestures. Several techniques including mini-batch gradient descent with L2 regularization, DropRule and AdaBound are integrated to improve the generalization ability of the system and the efficiency of training. Numerical results show that the average classification accuracy of the multiple classifier systems is 95.12%, and this value is higher than some other multiple classifier systems (CNN, LDA, etc.) using MMG signals as the inputs for hand gesture recognition.**

I. INTRODUCTION

The friendly human-machine interface is not only an important bridge to realize the natural interaction between patients and prosthetics, but also an important part of limb regeneration [1]. The electromyogram (EMG) is extracted directly from the skin surface and carries a large amount of control information from motor neurons to muscles, which is the most commonly used control signal source of human-machine interface [2]. Studies have shown that there is a causal relationship between muscle movements and EMG signals. The generation of EMG signals is about tens of milliseconds ahead of the occurrence of muscle movements. Despite its advantages, there are also some deficiencies in using EMG signals. For example, the amplitude of EMG signal is typically in the range of submillivolts, making it excessively sensitive to electrical noise. Besides, EMG signals and other biomedical signals are unrepeatable and random. More importantly, in order to reduce the influence of noise, the processings of EMG signals are more complicated compared with other biomedical signals, resulting in greater delay, which easily makes the biggest advantage of EMG gone. Since EMG is extracted directly from the skin surface, EMG sensors are needed to be attached to the skin surface. Therefore, EMG signals are also sensitive to other factors such as skin dryness and muscle fatigue.

An alternative way to detect muscle activity is mechanomyography (MMG), which measures mechanical vibrations during muscle contraction. Common ways to obtain MMG include accelerometers and microphones (by measuring mechanical vibrations), more robust force-sensitive force sensing resistors (FSRs) and air-bladder combined air pressure sensors. Although MMG signals typically have certain randomness, such randomness is generally relatively weak, and MMG can overcome the disadvantages of the sensitivity of EMG to electrical noise and skin humidity well. Therefore, we MMG signals as the original signals for gesture classification.

Adaptive neuro-fuzzy inference system (ANFIS), which was first proposed by Jang et al [3], is often regarded as the universal function approximator. It is obtained by embedding the fuzzy reasoning system into the adaptive neural network framework and iteratively updating parameters through the least square estimate-gradient decent (LSE-GD) hybrid learning method. Combining the learning and modeling capabilities of neural networks with human knowledge (fuzzy if then rules), ANFIS is able to approximate input-output mappings effectively no matter they are linear or non-linear. Such general approximator has been widely used in nonlinear regression and system identification [4], [5].

Although ANFIS is a type of effective learning system, it still has some shortcomings, such as weak generalization ability, poor training effect for big data and low convergence rate. In view of these shortcomings, MBGD-RDA algorithm was proposed in [6] recently, so that the ANFIS can achieve better generalization results and faster training speed in large data training set. While only the regression performance of MBGD-RDA was studied in [6] .

In this paper, we first proposed the MC ANFIS, that can be used to solve multiple classification problems, then we applied MBGD-RDA algorithm to MC ANFIS training process and found this method performed very well on gesture recognizing. Section II introduce generic ANFIS structure and MBGD-RDA algorithm, then the MC ANFIS and the learning algorithm are proposed in Section III. Section IV presents the experimental results. Finally, several conclusions are drawn in Section V.

II. MBGD-RDA ANFIS

This section introduces the structure of the generic ANFIS architecture and MBGD-RDA algorithm.

A. Generic ANFIS Structure

Consider a fuzzy reasoning system with M inputs x_1, x_2, $\cdots$, x_M and one output f. Suppose each input is fuzzified by R membership functions, where R is the size of the rule base.

Jian Huang (corresponding author), Lei Wang, Dongrui Wu, Tao Duan and Rui Zong are with the Key Laboratory of Image Processing and Intelligent Control, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China. Email: huang_jan@mail.hust.edu.cn, wml0531@hust.edu.cn, drwu@hust.edu.cn, mr_duan@hust.edu.cn and u201514441@alumni.hust.edu.cn.

Shicong Jiang is with the Laboratory for Economics and Computation, Department of Systems Engineering and Engineering Management, City University of Hong Kong, Kowloon 999077, Hong Kong SAR China. Email: shicong.jiang@my.cityu.edu.hk.

978-1-7281-6479-3/20/$31.00 ©2020 IEEE

Rule $i : (i = 1, 2, \cdots, R)$
if x_1 is $A_{i,1}$, x_2 is $A_{i,2}$, $\cdots$, x_m is $A_{i,m}$ then
$f_i = b_{i,1}x_1 + b_{i,2}x_2 + \cdots + b_{i,m}x_M + b_{1,0}$

A general structure of ANFIS is shown in the Fig.1. Each square node represents an adaptive node while the circle nodes represent fixed nodes in the figure.

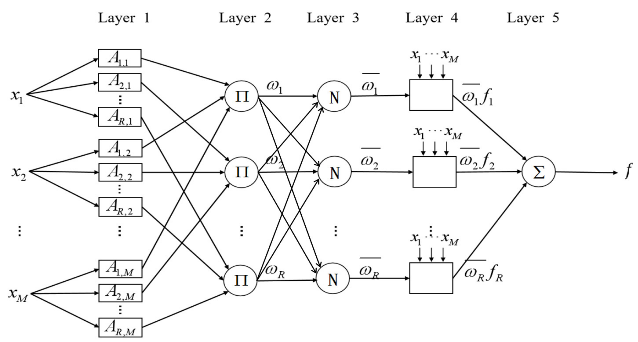

Fig. 1. M-input-1-output ANFIS architecture with first-order Sugeno fuzzy model and R rules.

Layer 1: each node in the Layer 1 is a square node with node equation:

$$O^1_{r,m} = A_{r,m}(x_m) \tag{1}$$

where x_m is the m-th element of the input vector. $A_{r,m}$ stands for the membership grade of x_m. Gaussian function is commonly used membership function with maximum being 1 and minimum being 0:

$$A_{r,m}(x_m) = exp\{-(\frac{x_m - c_{r,m}}{a_{r,m}})^2\} \tag{2}$$

where $\{a_{r,m}, c_{r,m}\}$ is the parameter set. As the values of these parameters change, so does the Gaussian function. Thus, various forms of membership functions are presented in linguistic label $A_{r,m}$. The parameters in this layer are called premise parameters.

Layer 2: all nodes in this layer are circle nodes marked Π, that multiply the inputs as the firing strength of the rule.

$$O^2_r = \omega_r = \Pi^2_{m=1} A_{r,m}(x_m) \tag{3}$$

Layer 3: each node in this layer is a circle node marked N. The r-th node calculates the ratio of r-th rule's firing strength to sum of the firing strength of all rules:

$$O^3_r = \overline{\omega_r} = \frac{\omega_r}{\Sigma^R_{r=1}\omega_r} \tag{4}$$

where R is the scale of the rule base. The output stands for the weighted activation of the r-th rule.

Layer 4: all nodes in this layer are square nodes with the following functions:

$$O^4_r = \overline{\omega_r} f_r = \overline{\omega_r}(\Sigma^2_{m=1} b_{r,m}x_m + b_{r,0}) \tag{5}$$

where $\overline{\omega_r}$ is the output of the third layer. f_r is the r-th rule output and $\{b_{r,m}\}$ is the set of consequent parameters.

Layer 5: a single node in this layer is a round node marked Σ, that calculates the sum of all input signals:

$$O^5 = \Sigma_r \overline{\omega_r} f_r = \frac{\Sigma_i \overline{\omega_r} f_r}{\Sigma_r \overline{\omega_r}} \tag{6}$$

For the regression part, the forward propagation comes to an end. Because layer 5 returns one single real value, and that is what we want in regression problem. As for classification, we often need an operator to transform the real value to integer, which stands for the class we finally get.

B. MBGD-RDA Algorithm

As mentioned in the introduction, MBGD-RDA is an optimized algorithm for Takagi-Sugeno-Kang fuzzy system. It integrates mini-batch gradient descent (MBGD) with regularization, DropRule and AdaBound algorithms to improve the generalization ability of the system and the efficiency of training.

In the back-propagation algorithm, gradient descent (GD) algorithms are often used to update parameters. The GD algorithms include three types: batch gradient descent (BGD), which uses all data sets; stochastic gradient descent (SGD), which uses a single random data; and MBGD, which uses a randomly selected data batch. Compared with BGD, MBGD is much more efficient in the case of larger batchsize and multiple training times. Compared with SGD, MBGD is better when the two algorithms share the same training time.

The training procedure of system parameters is to make the system perform better in the testing set, which means stronger generalization ability. Regularization is a common generalization method, which makes some system parameters close to zero by adding penalty items to the cost function to prevent overfitting. The improved loss function L is given as below:

$$L = L_0 + \frac{\lambda}{2} \Sigma^R_{r=1} \Sigma^M_{m=1} b^2_{r,m} \tag{7}$$

where L_0 is the original cost function. λ is the regularization coefficient, and $b_{r,m}$ is the consequent parameter. The effect of regularization on the original system is that the gradient of coefficient term is larger in the back propagation.

DropRule is a generic algorithm similar to DropOut and DropConnect, which is widely used in deep learning. By randomly discarding some neurons or the links between neurons in the training process, we can prevent overfitting and improve the generalization ability of the system. To our knowledge, there have been already two types of DropRule algorithms. In the first method, a set of random numbers in the second layer of ANFIS is generated and compared with the preset threshold, determining whether to activate the rule or not. If none of rules is activated, all rules are considered to be activated. Another method generates a set of random numbers of Bernoulli distribution in the calculation of rule output at the fourth layer of ANFIS to determine whether the rule output is zero [7]. Both methods are able to improve the generalization ability of the system. This paper mainly used the first.

Learning rate is very important during the training procedure. If the learning rate is too small or too large, the convergence rate and training effect of system will not be good. AdaBound [8] is an improved algorithm on Adam [9] algorithm, which can adaptively change the learning rate, add boundary restrictions on the basis of the adaptive learning rate, and ensure that the learning rate can converge to a single value after enough training.

In the beginning, the lower bound of learning rate tends to zero. In order to improve learning efficiency, the adaptive learning rate is the value between l and u , which are the lower and upper bound. With the increase of iteration times, the lower and upper bounds of learning rate converge to the preset learning rate α. In this way, the efficiency of updating parameters is greatly improved, and when the learning rate α is set appropriately, the system tends to be stable with the increase of iterations.

III. MC ANFIS STRUCTURE

As mentioned in Section II.A, the generic ANFIS architecture computes one single output, which is suitable for binary classification. There are at least two typical ways to extend this binary classifier to multiple classifier. One is to design C_m^2 classifiers between each kind of class, which is called one against one (OAO). Another way is to design m classifiers between one class to the others, which is called one against all (OAA). However, in order to relabel the data and generate networks, the OAA model require lots of extra preprocessing. Besides, some categories could have the same meaning. Therefore, training different networks may lead to learning redundant rules and wasting CPU cycles [7]. In order to overcome these shortcomings, we present an extended architecture to do the multiple classification work in the following. Fig. 2 is an illustration of the extended architecture.

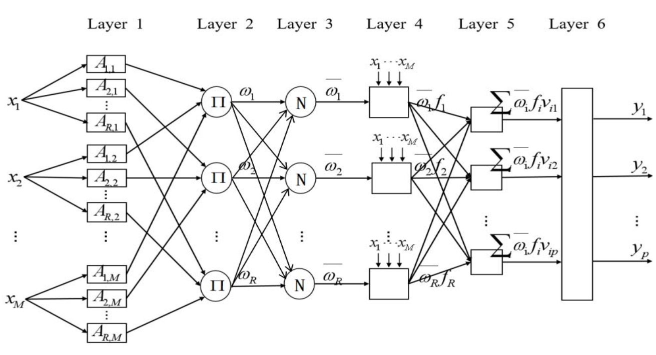

Fig. 2. The architecture of MC ANFIS.

The first four layers of the system are the same as the generic ANFIS structure shown in Section II. The fifth layer is the weighted sum of all the weighted rule outputs. Suppose there are P categories in total, then the weight is given by a matrix $V \in R^{R\times P}$, where R is the size of rule base. Then we have

$$O_p^5 = \Sigma_{r=1}^R O_r^4 \times v_{rp} \tag{8}$$

where $v_{rp} \in V$. The output of layer five are P real values. Thus, by using the softmax function in the sixth layer, P gestures can be classified. The output of softmax algorithm $O^6 \in R^{P\times 1}$ means the probability that the input belongs to P categories, and the index of the maximum probability is selected as the final output of the final classifier. For example, suppose O_i^6 is the maximum of the layer six, then i is the output category, that the input belongs to.

$$O_p^6 = y_p = \frac{exp(O_p^5)}{\Sigma_i exp(O_p^5)} \tag{9}$$

$$Output = index(max(O^6)) \tag{10}$$

Up to now, forward propagation is all introduced, and the necessary calculation formulas for back propagation are given below.

Cross-entropy is one of the commonly used loss functions in solving classification problems, and its output represents the closeness of two distributions. Different from other loss functions, cross entropy loss functions enable the gradient not sensitive to the gradient of the sixth layer when it is used with softmax algorithm. Select the cross entropy as the loss function of the system and add the regular term $\frac{\lambda}{2}\Sigma_{i=1}^R\Sigma_{m=1}^M b_{r,m}^2$, the loss function is

$$E = -\Sigma_{p=1}^P t_p ln(y_p) + \frac{\lambda}{2}\Sigma_{r=1}^R\Sigma_{m=1}^M b_{r,m}^2 \tag{11}$$

where $t_p \in \{1, 0\}$, stands for whether the input vector belongs to the p-th class or not. y_p is the p-th output of the layer 6 calculated from (9). λ is the regularization coefficient. R is the number of rules and M is the length of input vector. In order to minimize E, system parameters need to be adjusted. According to the chain rule, the gradients of E on system premise parameters are given as follows:

$$\begin{aligned}\frac{\delta E}{\delta O_{r,m}^1} &= \Sigma_{p=1}^P \frac{\delta E}{\delta O_p^6}\frac{\delta O_p^6}{\delta O_p^5}\frac{\delta O_p^5}{\delta O_r^4}\frac{\delta O_r^4}{\delta O_r^3}\frac{\delta O_r^3}{\delta O_{r,m}^2}\frac{\delta O_{r,m}^2}{\delta O_{r,m}^1} \\ &= -\Sigma_{p=1}^P t_p(1-O_p^6)v_{rp}f_r\frac{\overline{\omega_r}(1-\overline{\omega_r})}{A_{r,m}}\end{aligned} \tag{12}$$

$$\begin{aligned}\Delta c_{r,m} &= \Sigma_{p=1}^P \frac{\delta E}{\delta O_p^6}\frac{\delta O_p^6}{\delta O_p^5}\frac{\delta O_p^5}{\delta O_r^4}\frac{\delta O_r^4}{\delta O_r^3}\frac{\delta O_r^3}{\delta O_{r,m}^2}\frac{\delta O_{r,m}^2}{\delta O_{r,m}^1}\frac{\delta O_{r,m}^1}{\delta c_{r,m}} \\ &= \frac{\delta E}{\delta O_{r,m}^1}\frac{\delta O_{r,m}^1}{\delta c_{r,m}} \\ &= -2\Sigma_{p=1}^P t_p(1-O_p^6)v_{rp}f_r\overline{\omega_r}(1-\overline{\omega_r})\frac{x-c_{r,m}}{(a_{r,m})^2}\end{aligned} \tag{13}$$

$$\begin{aligned}\Delta a_{r,m} &= \Sigma_{p=1}^P \frac{\delta E}{\delta O_p^6}\frac{\delta O_p^6}{\delta O_p^5}\frac{\delta O_p^5}{\delta O_r^4}\frac{\delta O_r^4}{\delta O_r^3}\frac{\delta O_r^3}{\delta O_{r,m}^2}\frac{\delta O_{r,m}^2}{\delta O_{r,m}^1}\frac{\delta O_{r,m}^1}{\delta a_{r,m}} \\ &= \frac{\delta E}{\delta O_{r,m}^1}\frac{\delta O_{r,m}^1}{\delta a_{r,m}} \\ &= -2\Sigma_{p=1}^P t_p(1-O_p^6)v_{rp}f_r\overline{\omega_r}(1-\overline{\omega_r})\frac{(x-c_{r,m})^2}{(a_{r,m})^3}\end{aligned} \tag{14}$$

2020 5th International Conference on Advanced
Robotics and Mechatronics (ICARM)

Similarly, the gradients of consequent parameters can be obtained.

$$\begin{aligned}\Delta b_{r,m} &= \Sigma_{p=1}^{P} \frac{\delta E}{\delta O_p^6} \frac{\delta O_p^6}{\delta O_p^5} \frac{\delta O_p^5}{\delta O_r^4} \frac{\delta O_r^4}{\delta b_{r,m}} - \lambda \Sigma_{r=1}^{R} \Sigma_{m=1}^{M} b_{r,m} \\ &= -\Sigma_{p=1}^{P} t_p (1 - O_p^6) v_{rp} \overline{\omega_r} x_m - \\ &\quad \lambda \Sigma_{r=1}^{R} \Sigma_{m=1}^{M} b_{r,m}\end{aligned} \tag{15}$$

$$\begin{aligned}\Delta b_{r,0} &= \Sigma_{p=1}^{P} \frac{\delta E}{\delta O_p^6} \frac{\delta O_p^6}{\delta O_p^5} \frac{\delta O_p^5}{\delta O_r^4} \frac{\delta O_r^4}{\delta b_{r,0}} \\ &= -\Sigma_{p=1}^{P} t_p (1 - O_p^6) v_{rp} \overline{\omega_r}\end{aligned} \tag{16}$$

$$\begin{aligned}\Delta v_{rp} &= \frac{\delta E}{\delta O_p^6} \frac{\delta O_p^6}{\delta O_p^5} \frac{\delta O_p^5}{\delta v_{rp}} \\ &= -t_p (1 - O_p^6) \overline{\omega_r} f_r\end{aligned} \tag{17}$$

The applying of MBGD-RDA algorithm to MC ANFIS is given in Algorithm 1.

Algorithm 1 MC ANFIS Using MBGD-RDA Algorithm

Input: X:the set of training data;
Y:the set of training labels;
N_{max}:the maximum times of iteration;
N_{mb}:the size of mini-batch;
λ:the regularization coefficient;
P:the threshold of DropRule;
λ_1 , λ_2: adaptive learning rate of AdaBound;
Output: c:the centers of Gauss membership functions;
a:the standard deviation of Gauss membership functions;
b:the consequent parameters of MC ANFIS;
V:the weight matrix of the fifth Layer;
1: Initialize c,a,b,V,Δc,Δa,Δb,Δv;
2: **repeat**
3: Randomly select N_{mb} training examples from X (**MBGD**);
Compute system $Output \leftarrow (1)(3)(4)(5)(8)(9)(10)$ by using **DropRule**;
Compute $\Delta c \leftarrow (13)$;
Compute $\Delta a \leftarrow (14)$;
Compute $\Delta b \leftarrow (15, 16)$ by using **regularization**;
Compute $\Delta v \leftarrow (17)$;
update c,a,b,V by using **AdaBound**
4: **until** Iteration times $> N_{max}$
5: **return** c,a,b,V;

IV. EXPERIMENTS AND RESULTS

In this part, the MC ANFIS and OAA ANFIS models were used to classify the six gestures use pressure-based MMG (pMMG) data of four healthy adult males under two learning algorithms, i.e. the MBGD-RDA and the conventional batch gradient decent with regularization (BGD-R). The performance was evaluated by the raw classification accuracy (RCA) and average training time. It should be pointed out that in order to reflect the training efficiency, this experiment was completed on a computer with 3.2GHz Core i7 8700k CPU, and the training time of each model was recorded in the experiment (excluding the time of pMMG data processing).

A. pMMG data acquisition and preprocessing

The pMMG is obtained by using the device mentioned in [10] , which is shown in Fig. 3. The assumptions of this kind of device can be conclude as below.

- Muscles are rigid and attached to elastic tendons of coefficient k.
- The total volume of each muscle remains the same.
- Each muscle is cylindrical in shape and has a uniform cross-sectional area.
- The total cross-sectional area of the air-bladder and muscles is constant.
- The sacs do not move relative to the muscles, and the total cross-sectional area of a group of air-bladder and muscle groups does not change.

Based on the assumptions given above, a nearly linear relationship between muscle strength and pressure changes can be obtained.

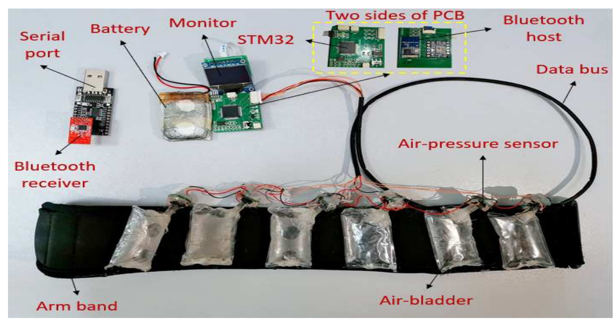

Fig. 3. The device that collects pMMG signals.

In order to get more information of muscle activity corresponding to the six gestures, we fixed the sensors with the armband at the position of the forearm near the elbow joint, as shown in Fig. 4. The names of the six main muscles measured and the role of muscles in hand movement are shown in TABLE I.

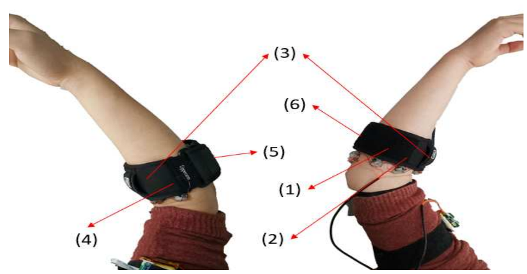

Fig. 4. location where air-bladder is fixed

We recruited four volunteers and collected training data from them following the experimental paradigm. Firstly, volunteers were required to relax for two seconds, and then

978-1-7281-6479-3/20/$31.00 ©2020 IEEE

performed each gesture in seven seconds. At the first five seconds, volunteers performed the gesture movement with maximum strength and kept still, then relax in the next two seconds. The whole process was repeated five times, and the gesture data in relax time were not collected. The pMMG data collected in one process are shown in Fig. 5.

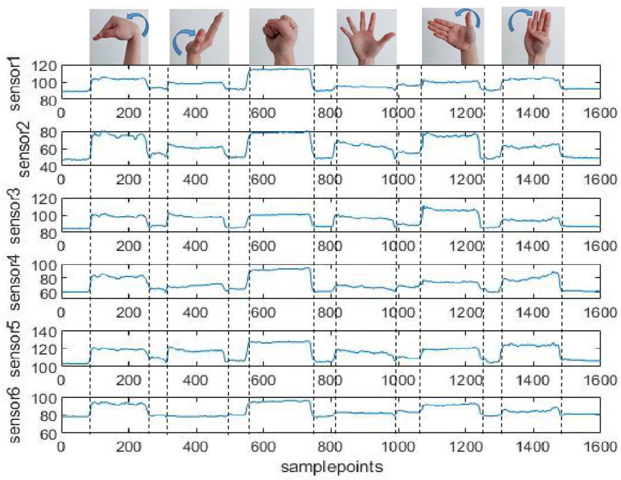

Fig. 5. Original pMMG of six gestures. The data sampled from sensor $1 \sim 6$ were recorded from top to bottom. the pMMG data of wrist flexion, wrist extension, finger flexion, finger extension, turn left and turn right were recorded from left to right.

The sampling rate of the sensor used in this experiment is about 30Hz, and around 6000 pMMG data of six gestures were collected as the original data in each of four volunteers. After dimensionality reduction through PCA, 70% of the processed data were randomly selected as the training set and input to four kind of classifiers, and the rest of data were used as the testing set to evaluate the classification performance of the system.

B. Off-line classification experiment

In the experiment of OAA ANFIS using MBGD-RDA algorithm, the label value was given as 1 or 0, while the label value was given as $1, 2, \cdots, 6$ in the experiment of MC ANFIS using MBGD-RDA algorithm. The loss function was the cross-entropy function. Considering the influence of the learning rate to the experiment, the learning rate α was set as 0.1, 0.05 and 0.025. Batchsize was set as 64. Iteration time was set as 200. Regularization coefficient λ was set as 0.05, and DropRule threshold P was set as 0.5. AdaBound adaptive learning rate $\lambda_1 = 0.9$, $\lambda_2 = 0.999$. The results of the training and testing classification accuracy curve of the system are shown in Fig. 6.

TABLE I

LOCATION AND ROLE OF THE SIX MUSCLES MEASURED

Location	Name of muscle	Role
1	Flexor carpi ulnaris	Wrist flexion and ulnar deviation
2	Flexor digitorum	Fingers flexion
3	Flexor carpi radials	Wrist flexion and radial deviation
4	Extensor carpi radials	Wrist extension and radial deviation
5	Extensor digitorum	Fingers extension
6	Extensor carpi ulnaris	Wrist extension and ulnar deviation

The average RCA of four models' testing set in five times experiments and the average training time (ATT) is shown in TABLE II. The result showed that, the MBGD-RDA applied MC ANFIS had the minimum training time, and its training results were superior at the same time.

TABLE II

RCA AND ATT OF THE TESTING DATASETS USING DIFFERENT METHODS AND LEARNING RATES

Method	ATT[1]	Average RCA		
		$\alpha = 0.1$	$\alpha = 0.05$	$\alpha = 0.025$
MC[2]	61.8560	94.84%	95.12%	91.32%
MC[3]	568.0859	85.13%	86.82%	87.41%
OAA[4]	184.1379	97.78%	98.02%	97.04%
OAA[5]	1206.1000	97.33%	93.84%	96.98%

The average training time.(unit: second)
MC ANFIS model using MBGD-RDA alorithm
MC ANFIS model using BGD-R alorithm
One Against All ANFIS using MBGD-RDA algorithm
One Against All ANFIS using BGD-R algorithm

TABLE III reports the average testing classification accuracy of several methods with MMG input. Overall OAA ANFIS MBGD-RDA achieved the best classification accuracy compared with other methods. Though MC ANFIS MBGD-RDA performed slight worse on classification accuracy than OAA ANFIS MBGD-RDA, it takes almost 2 times shorter time than the OAA ANFIS MBGD-RDA in training procedure in average. Both of the two kinds of multi-classification model performed better after using MBGD-RDA algorithm.

TABLE III

AVERAGE CLASSIFICATION ACCURACY OF SEVERAL METHODS WITH MMG INPUT

Method	Average RCA
MC ANFIS MBGD-RDA	95.12%
MC ANFIS BGD-R	87.41%
OAA ANFIS MBGD-RDA	98.04%
OAA ANFIS BGD-R	97.33%
CNN(convolutional neural network) [11]	94%
SVM(support vector machine) [12]	95.20%
BPNN(BP neural network) [13]	86.41%
LDA(linear discriminant analysis) [14]	92%

As can be seen from Fig. 6, under some model parameters, the classification error rate of some testing sets is lower than that of the training set, which is not common in general classification problems.

V. CONCLUSIONS

The MC ANFIS we proposed can save a lot of time and memory in the training procedural compared with OAA model, while the optimization algorithm is needed to improve classification accuracy. Both the training efficiency and the RCA were improved a lot after using MBGD-RDA algorithm. It is reasonable to say that MBGD-RDA is a reliable learning algorithm in fuzzy inference system.

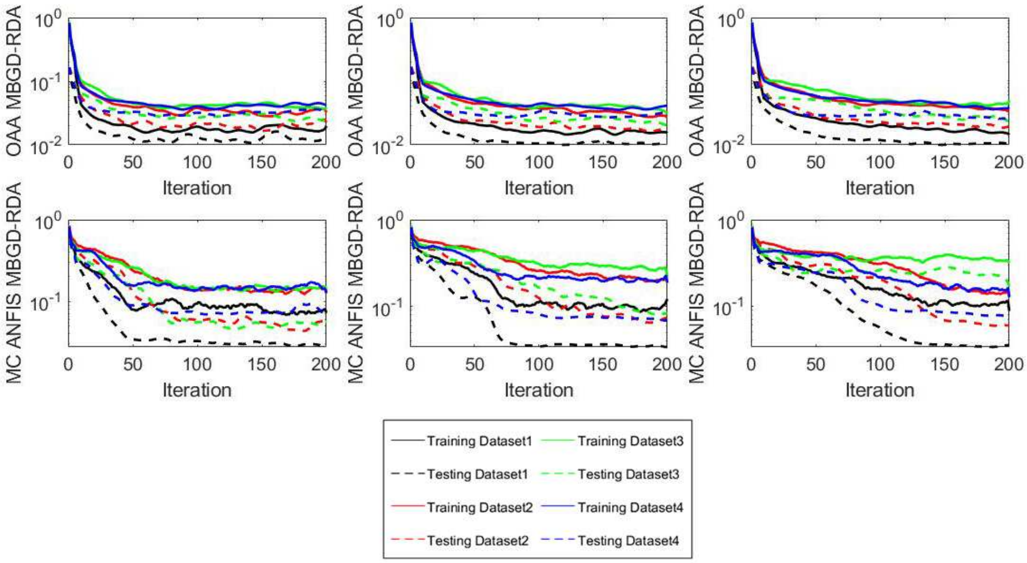

Fig. 6. The five times experimental average training (solid lines) and testing (dashed lines) classification accuracy curves of four volunteers, and each kind of color stands for a dataset collected from a volunteer. The plot shows the outputs of two method optimized by MBGD-RDA algorithm, at the learning rate of 0.1, 0.05, 0.025, from left to right.

According to the assumptions of pMMG sensor, if the air-bladder can cover the entire cross-sectional area of the arm (in fact, the coverage rate is about 80%), and the width can be narrow enough. Then whatever position the user wears the device in the forearm, similar well-performed parameter sets will be obtained.

In the future, if there is a major breakthrough in the study of interval type-2 ANFIS, it will make a lot of sense to extend MC ANFIS framework to MC IT2 ANFIS to improve the robustness of the system. In practical application, the system can be extended to medical treatment, AR and some other fields that require gesture recognition technology.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China under Grant U1913207, the Research Fund of PLA of China (BWS17J024) and the Fundamental Research Funds for the Central Universities (HUST: 2019kfyRCPY, 2019kfyXKJC019).

REFERENCES

[1] E. Biddiss and T. Chau, "Upper-limb prosthetics: critical factors in device abandonment," *American journal of physical medicine & rehabilitation*, vol. 86, no. 12, pp. 977–987, 2007.

[2] V. S. Ramachandran and W. Hirstein, "The perception of phantom limbs. the do hebb lecture," *Brain: a journal of neurology*, vol. 121, no. 9, pp. 1603–1630, 1998.

[3] J.-S. Jang, "Anfis: adaptive-network-based fuzzy inference system," *IEEE transactions on systems, man, and cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.

[4] J.-S. Jang, "Input selection for anfis learning," in *Proceedings of IEEE 5th International Fuzzy Systems*, vol. 2, pp. 1493–1499, IEEE, 1996.

[5] M. Wei, B. Bai, A. H. Sung, Q. Liu, J. Wang, and M. E. Cather, "Predicting injection profiles using anfis," *Information Sciences*, vol. 177, no. 20, pp. 4445–4461, 2007.

[6] D. Wu, Y. Yuan, J. Huang, and Y. Tan, "Optimize tsk fuzzy systems for regression problems: Mini-batch gradient descent with regularization, droprule and adabound (mbgd-rda)," *IEEE Transactions on Fuzzy Systems*, 2019.

[7] A. B. Khalifa and H. Frigui, "Mcmi-anfis: A robust multi class multiple instance adaptive neuro-fuzzy inference system," *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Vancouver, Canada, 2016, pp. 1991–1998.

[8] L. Luo, Y. Xiong, Y. Liu, and X. Sun, "Adaptive gradient methods with dynamic bound of learning rate," *arXiv preprint arXiv:1902.09843*, 2019.

[9] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[10] P.-G. Jung, G. Lim, S. Kim, and K. Kong, "A wearable gesture recognition device for detecting muscular activities based on air-pressure sensors," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 2, pp. 485–494, 2015.

[11] Y. Ma, Y. Liu, R. Jin, X. Yuan, R. Sekha, S. Wilson, and R. Vaidyanathan, "Hand gesture recognition with convolutional neural networks for the multimodal uav control," *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, Linkoping, Sweden, 2017, pp. 198–203.

[12] H. Ding, Q. He, L. Zeng, Y. Zhou, M. Shen, and G. Dan, "Motion intent recognition of individual fingers based on mechanomyogram," *Pattern Recognition Letters*, vol. 88, pp. 41–48, 2017.

[13] L. Cheng, J. Wang, C. Li, X. Zhan, C. Zhang, Z. Qi, and Z. Zhang, "Classification of mmg signal based on emd," *Advanced Computational Methods in Life System Modeling and Simulation*, Nanjing, China, 2017, pp. 23–34.

[14] P. B. Shull, S. Jiang, Y. Zhu, and X. Zhu, "Hand gesture recognition and finger angle estimation via wrist-worn modified barometric pressure sensing," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 4, pp. 724–732, 2019.