Design of a Hybrid Neural Network Based on Kernel Structure Fusion

Hui Wen*, Tao Yan, Yan Che, Tongbin Li, Yuanmo Lin
Institute of Information Engineering, Putian University, Putian 351100, Fujian Province
**Correspondence author's email: wenhui [AT] szu.edu.cn*

Abstract—**To improve the network performance of single BP networks and RBF networks, a hybrid neural network classifier based on kernel structure fusion is presented in this paper. The presented method does not depend solely on a single RBF network and BP network framework. By establishing the effective connection mechanism between the Gaussian kernels with different parameters and sigmoid kernels in different hidden layers, the fused network can effectively extract the local characteristics of sample spatial distribution, and the non-linear learning and classification in the Gaussian kernel space can be accomplished. The fused network structure combines the advantages of RBF network and BP network, and reduce the dependence on parameters selection of hidden nodes in single RBF and BP networks effectively. Experiments on two artificial data sets and three benchmark data sets validate the superiority of the presented network structure.**

Keywords- kernel; Structure Fusion; hybrid; neural networks

I. INTRODUCTION (HEADING 1)

Nowadays, neural networks have been widely used in many fields of human life, such as image classification, medical diagnosis, and so on. Among all kinds of neural networks, BP networks [1] and RBF networks [2] are one of the most widely used network models. The main difference in determining the network model is the kernel function and mapping mechanism used by different hidden nodes in the network. Taking BP neural networks as an example, the kernel function used by hidden nodes is sigmoid kernel, which has good global response characteristics. Its learning rules adopt the method of gradient descent based on error back propagation to iteratively update network weights, thus minimizing the sum of square errors of network output. The main disadvantage of BP network is that the convergence speed is slow, and it often falls into the local minimum. With the increase of the number of hidden layers of the network, this situation becomes more serious.

RBF networks is another typical neural network model. Different from BP Neural Network, the kernel function used by hidden nodes is radial basis function, which has good local response characteristics. In RBF networks, the most common radial basis function kernels are Gauss kernels. By mapping the Gauss kernels n, the separability of the original sample space can be effectively improved, and then a linear classifier such us LMS or RLS is connected to complete the classification of the kernel space. However, for complex classification problems, once the parameter selection of hidden nodes is inappropriate, the improper mapping of the Gauss kernels may lead to the burden of the subsequent weight updating process.

To improve the network performance of single BP networks and RBF networks, and reduce the dependence on parameters selection of hidden nodes in BP and RBF networks, a hybrid neural network classifier based on kernel structure fusion is presented in this paper. By establishing the effective connection mechanism between the Gaussian kernels with different parameters and sigmoid kernels in the hidden layer of the BP network, the fused network can effectively extract the local characteristics of sample spatial distribution, the separability of sample space can be effectively improved by mapping input samples through different Gaussian kernels, and then the non-linear learning and classification of networks can be completed in the Gaussian kernel space, which is accomplished by the cascaded BP network. Thus, the presented method can combine the advantages of RBF network and BP network, and overcome the shortcomings of the single BP network and RBF network.

When the hybrid network structure is established, the subsequent task is to complete the algorithm implementation of the fused hybrid neural network. The learning algorithm of the presented hybrid neural network is mainly divided into two stages. Firstly, the parameters of the Gaussian kernels are established by introducing the fuzzy C-means clustering algorithm, and then the weight parameters connected to the sigmoid kernels are updated by the backward propagation algorithm with stochastic gradient descent.

The presented network structure is compared with the BP network based on stochastic gradient descent and the radial basis function network (RBF) based on "fuzzy C-means, RLS" algorithm. The experimental results show the superiority of the presented network structure.

II. RELATED WORK

To improve the deficiency of BP networks and RBF networks, researchers have studied various improvements. Most of the existing methods are based on the improvement of a single network structure. Typical methods for optimizing BP networks include initialization weights based on global optimization [3,4], adding momentum term to local gradient [5], adaptive adjustment of learning rate [6], modification of error cost function [7], and dynamic adjustment of network structure [8], which overcome the shortcomings of BP network to a certain extent.

For RBF networks, the key tasks involves determining the number of kernels and optimizing the kernel parameters. For the different classification problems, by adjusting the

number of kernels and kernel parameters, the reliability of the kernel mapping can be improved to varying degrees. Typical determining the number of kernels include minimum resource allocation network (MRAN) [9], sequential learning algorithm for growing and pruning the RBF (GAP-RBF) [10] and other incremental design of radial basis function networks [11, 12]. Existing methods for optimizing network kernel parameters include k-means clustering [13], fuzzy C-means clustering [14], orthogonal forward selection algorithm [15], evolutionary algorithm [16], and so on. Once the number of kernels and kernel parameters are determined, the corresponding linear classification algorithm such as LMS or RLS is used to optimize output weights. How to optimize RBF network structure and kernel parameters is still a open question.

To realize interconnection of different types of neural network structures, cascaded feedforward neural networks [17,18] connected with each other have been applied to some fields. Note that the cascaded neural networks is composed by the connection of several different sub-networks, where each sub-network is independent of each other. By decomposing a complex problem into several sub-problems, each different sub-network is used to solve different sub-problems. Its essence is still to solve the corresponding problems by combining the existing methods. For a specific classification task, from the perspective of network structure, how to fuse and adjust different types of neural networks is a work worth exploring。

On this issue, the convolution neural network in deep learning [19-21] cascades the convolution layer with the hidden layer of BP network, where the output of the convolution layer is processed as the input of BP network, and finally uses BP algorithm to adjust the weight of the whole network. In [22], the BP network cascaded with the convolution layer in the original network was changed to ELM network. Reference [23] realizes the adjustment of network structure by adding an interference pattern layer to BP neural network, where the interference pattern layer refers to a new hidden layer added between input layer and BP hidden layer. In the learning of network parameters, the original input samples are mapped through the interference mode layer, and then the weight of the latter BP network is adjusted by using BP algorithm. In this way, the parameter selection dependence of the original BP neural network is reduced, and the classification performance is improved to a certain extent.

III. Hybrid Neural Network Based on Kernel Structure Fusion

To illustrate the characteristics and advantages of the presented method, Fig. 1 gives a mapping effect of Gauss kernels with different parameters. The Gaussian kernels connected with the input samples are used to localize the training sample space. It maps the samples of different regions to the vertices of the unit hypercube, which can effectively improve the separability of the sample space. Then, the learning and classification of the BP network is completed in the Gaussian kernel space. In this way, the local corresponding properties the Gaussian kernel and the global response characteristics of the sigmoid kernel can be effectively combined.

Set the number of Gaussian kernels is K . For an arbitrary input sample $\mathbf{x} \in R^t$, when the sample passes through Gaussian kernels, the localized mapping relationship can be expressed as $f : R^t \to (0,1]^K$. In this way, the geometric shape of sample distribution in sample space can be used as mapping feature to form new feature vectors, which can improve the separability of original sample space. Then, the effective classification of feature space samples through Gaussian kernel mapping can be completed by using non-linear BP network. On the premise that the separability of sample space is improved, the risk of local minimum can be avoided, and the convergence speed of network can be accelerated. Thus, the presented BP network based on kernel fusion can improve the BP network effectively.

According to the above description, Fig. 2 shows the structure of the BP neural network based on kernel fusion in this paper. The presented network structure consists of three components:

1. Input layer. This layer consists of t nodes, where t is the dimentionality of input sample $\mathbf{x}$.

2. Fusion layer. This layer consists of a set of different parameters of Gaussian kernels and sigmoid kernels with the same parameters. The connection weight between input nodes and Gaussian kernels is 1. The weights connecting Gaussian kernels and sigmoid kernels are randomly initialized. Set the number of kernels is c , the mapping of Gaussian kernels to input sample $\mathbf{x}$ can be expressed as

$$\phi_j(\mathbf{x}) = \exp(-\frac{1}{2\sigma_j^2} \| \mathbf{x} - \mu_j \|^2),\ j = 1, 2, \ldots c \tag{1}$$

where the parameters μ_j , σ_j are the center and width of the jth Gaussian kernel, respectively.

When $\phi_j(\mathbf{x})$ passes through the first sigmoid layer, the jth node of the lth sigmoid layer induced local field to be expressed as

$$v_j^{(l)} = \sum_i \omega_{ji}^{(l)} y_i^{(l-1)} \tag{2}$$

where $\omega_{ji}^{(l)}$ is the weight of the ith node from the layer $l-1$ to the layer l , and $y_i^{(l-1)}$ is the output of the ith node from layer $l-1$.

In this paper, the sigmoid kernel selected is hyperbolic tangent function, thus

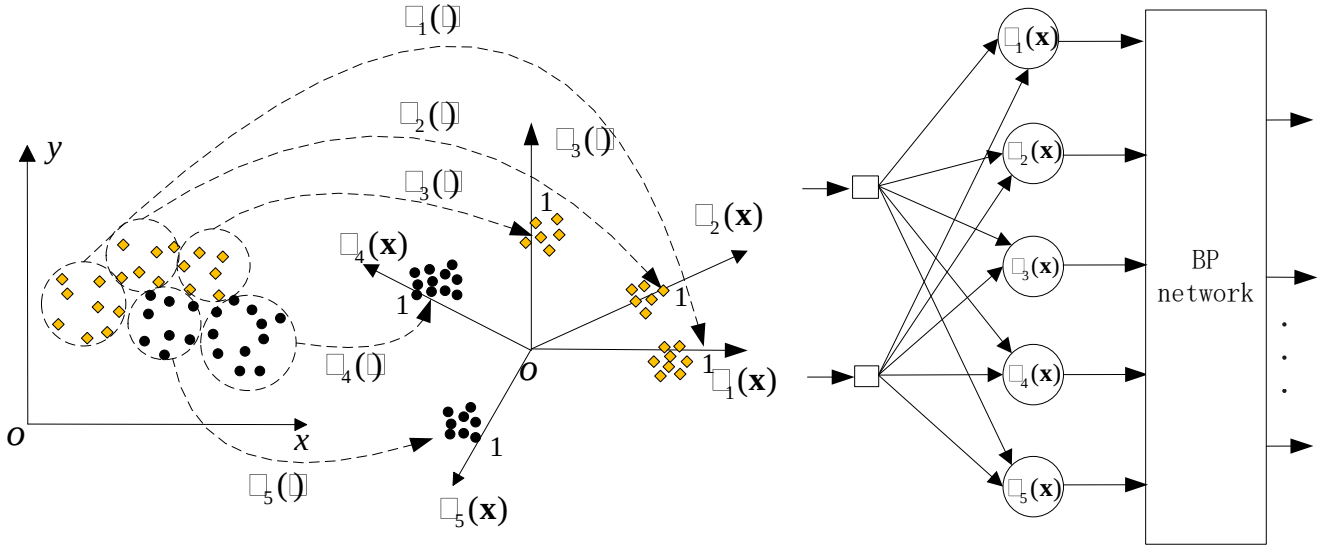


Figure 1. The mapping effect of Gauss kernels with different parameters

$$y_j^{(l)} = a\tanh(bv_j) \tag{3}$$

where a and b are constants.

If $l = 1$, the node j is in the first sigmoid layer, here we can get

$$y_j^{(0)} = \phi_j(\mathbf{x}) \tag{4}$$

3. Output layer. The ith output node of the hybrid network based on kernel fusion can be denoted as

$$o_i(n) = y_j^{(L)} \tag{5}$$

where L is in the last sigmoid layer.

When the hybrid network structure is established, the subsequent task is to complete the algorithm implementation of fused hybrid neural network, the steps of the network learning algorithm are as follows:

1. Initialization. Mainly includes setting the number of Gaussian kernels, setting the number of hidden layers and hidden nodes of the BP network components, and randomly initializing the weight parameters of the BP network components.

2. Fuzzy c-means clustering of training samples. The specific implementation algorithm shows in Table I.

3. The clustering mean $\mu_j\,(j = 1, 2, \ldots, c)$ is used as the center of the Gaussian kernel function. Using (2) to compute each mapping value. Here set $\sigma = d_{\max}/\sqrt{2c}$, $d_{\max}$ is the furthest Euclidean distance between all the centers.

4. The threshold ε is set as the stopping condition of iteration. The Gaussian kernel mapping value $\varphi(\mathbf{x})$ is set as the input of sigmoid kernel, Here $\varphi(\mathbf{x}) = (\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \ldots, \phi_c(\mathbf{x}))$.

5. Using (2)-(5) to realize the forward calculation of the network weights connected to each sigmoid kernel.

6. Calculating the mean square error function of the network.

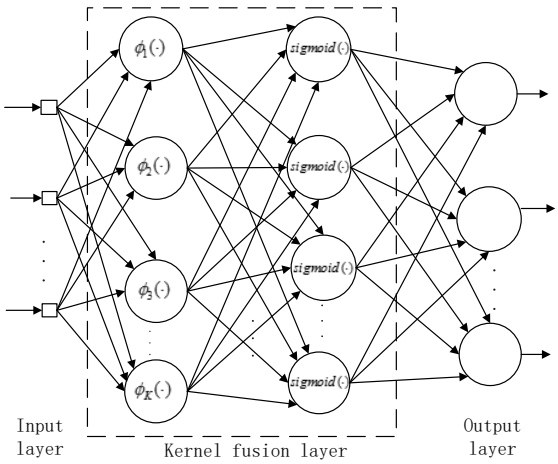
7. Backward calculation of the local gradient of the presented network

$$\delta_j^{(l)} = \begin{cases} e_j^{(L)}\phi_j'\left(v_j^{(L)}\right), & \text{for node } j \text{ in the output layer } L \\ \phi_j'\left(v_j^{(L)}\right)\sum \delta_k^{(l+1)}\omega_{kj}^{(l+1)}, & \text{for node } j \text{ in the sigmoid layer } l \end{cases} \tag{6}$$

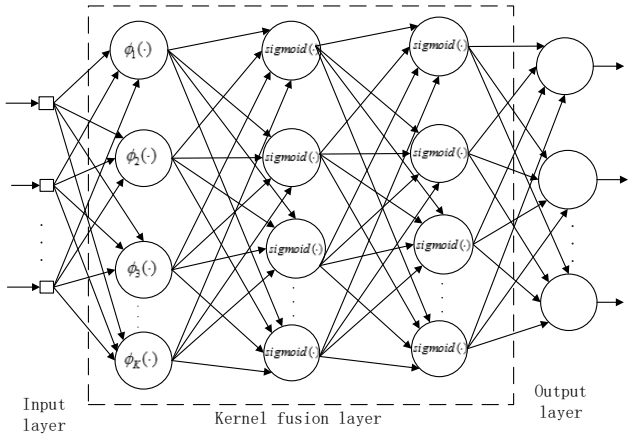
8. Updating the weights of each layer connected to the sigmoid kernel

$$\omega_{ji}^{(l)}(n+1) = \omega_{ji}^{(l)}(n) + \eta\delta_j^{(l)}(n)\,y_i^{(l-1)}(n) \tag{7}$$

where η is the learning rate, n is the iteration step.



（a）



（b）

Figure 2. Hybrid neural ntwork based on kernel structure fusion. （a）With a RBF layer and a single sigmoid layer of cascade（b）With a RBF layer and two sigmoid layers of cascade

9. Iteration of network layers connected with sigmoid kenel. Presenting a new round sample to the presented network and iterative calculation using step 5-8 until $J(\omega) < \varepsilon$, the algorithm terminated.

IV. Hybrid Neural Network Based on Kernel Structure Fusio

To verify the performance of the presented network, in this paper, two artificial datasets [24] and three UCI [25] benchmark datasets are introduced, and the presented network is compared with the BP neural networks based on stochastic gradient descent and the RBF networks with Gaussian kernels. The number of hidden layer units in BP network and RBF networks is optimized by cross validation. All the algorithms are carried out under the same conditions. For the BP networks, the momentum term α is chosen in the set

TABLE I. Algorithm Implementation of Fuzzy C-Means Clustering

Given the sample set $\{x_i\}_{i=1}^{N}$, set c is the number of clustering, μ_i (i=1,2,…c) is the clustering center.

1. Initializing the membership matrix U with random numbers to satisfy the constraints in the formula $\sum_{i=1}^{c} u_{ij} = 1, \forall j = 1,...,n$, where u_{ij} is between 0 and 1.

2. Calculating the clustering center, where $\mu_i = \sum_{j=1}^{n} u_{ij}^m x_j \Big/ \sum_{j=1}^{n} u_{ij}^m$.

3. Calculating the objective function $J(U, \mu_1, ..., \mu_c) = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c}\sum_{j}^{n} u_{ij}^m d_{ij}^2$. If $J < \xi$, the algorithm stops. Here $d_{ij} = \| \mu_i - x_j \|$. m is a weighted index number and $m \geq 1$.

4. Updating U Matrix, where $\mu_i = 1 \Big/ \sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}} \right)^{2/(m-1)}$. Return to step 2 for iterative execution.

TABLE II. Description of Classifying Datasets

Datasets	No. of classes	No. of features	Training samples	Testing samples
Double moon	2	2	300	4000
Twist	2	2	500	4000
Diabetes	2	8	576	192
German credit (GC)	2	24	200	151
Ionosphere	2	34	500	500

$[0.1, 0.2, 0.3]$. The presented BP network with kernel fusion is chosen as $\alpha = 0$. Set the sigmoid kernel learning parameter is $a = 1.716$, $b = 2/3$. Each experiment was repeated 10 times under Intel (R) Core (TM) i5, 3.2 GHZ CPU, 4 G RAM, and MATLAB 2013. Table II gives the description of classification datasets.

A. Double moon classification problem

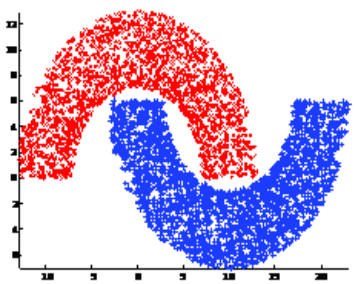
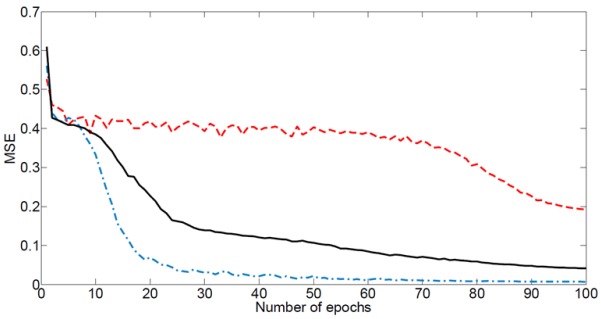
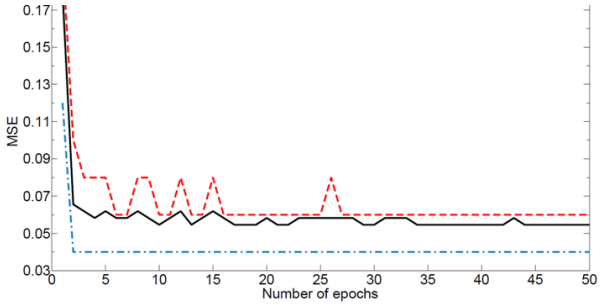


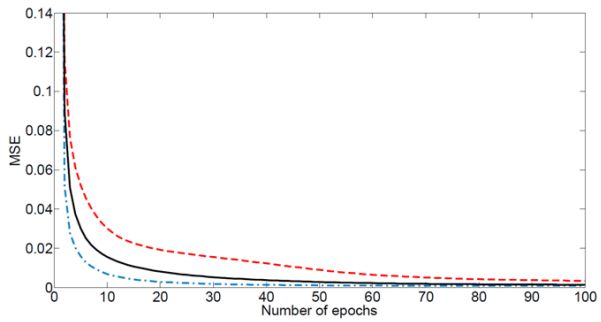
Figure 3. Double moon classification dataset



(a)



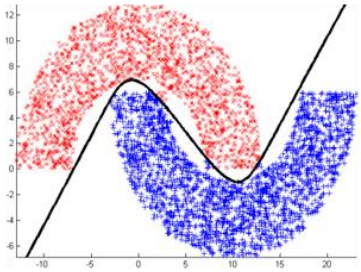
(b)



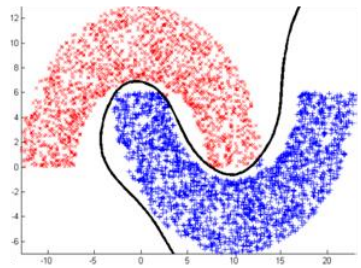
(c)

Figure 4. Comparison of mean square error learning curves of different networks on Double Moon dataset. (a) BP network (b) RBF network (c) Hybrid Neural Network Based on Kernel Structure Fusion

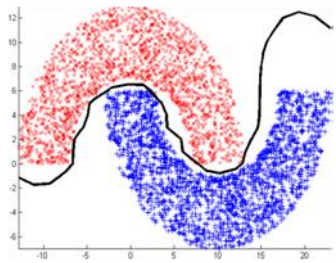
Fig. 3 gives the graphical display of the Double moon classification dataset. Fig. 4 shows the comparison of the mean square error learning curves of the presented network with the BP network and the RBF network, respectively, where the red dotted line represents the maximum mean square error curve, the blue dotted line represents the minimum mean square error learning curve, and the black solid line represents the average mean square error learning curve. The sigmoid layer number is set to 1. It can be seen that the presented network structure greatly improves the convergence speed of a single BP network, and its mean square error can converge to a smaller value. Compared with the RBF network, the presented network structure inherits the advantages of good stability of RBF network. In the first 10 rounds of iteration step, the mean square error converges rapidly. With the increase of iteration step size, the mean



(a)



(b)



(c)

Figure 5. Comparison of classification effects of different networks on the Double moon dataset.(a) BP network (b)RBF network (c) Hybrid Neural Network Based on Kernel Structure Fusion

square error of the presented network is further reduced, so the learning effect is effectively improved.

To measure the influence of the number of Gaussian kernels of the presented network, Table III gives the experimental comparison under different parameters in a single sigmoid hidden layer and two sigmoid hidden layers. It can be seen when the number of Gaussian kernels is within a certain range, by changing the number of sigmoid layers and hidden nodes, its mean square error in the training sample set can reach a lower value, and its classification accuracy in the test set is relatively higher. On the basis of learning the kernel mapping of training sample space, the subsequent connected non-linear BP network can provide a better classification. Thus, the presented network combines the advantages of strong stability of the Gaussian kernel and strong generalization ability of the sigmoid kernel. It effectively simplifies the setting of the number of the Gaussian kernels and sigmoid kernels in the fusion layer, the whole network can get a relatively higher classification performance.

TABLE III. PERFORMANCE COMPARISON OF DIFFERENT PARAMETERS OF HYBRID NEURAL NETWORK BASED ON KERNEL STRUCTURE FUSION ON THE DOUBLE MOON DATASET

The presented network	**No. of Gaussian kernels**	**No. of sigmoid kernel**		**Training error**	**Testing misclassification**
		First layer	***Second layer***		
With a RBF layer and a single sigmoid layer of cascade	8	8	--	0.000575	10(0.25%)
	12	4	--	0.000541	10 (0.25%)
	12	2	--	0.000485	3 (0.07%)
	14	5	--	0.000456	2 (0.05%)
	14	3	--	0.000652	4 (0.10%)
	16	5	--	0.000675	7 (0.18%)
	16	8	--	0.000834	8 (0.20%)
	16	3	--	0.000506	5（0.13%）
	18	5	--	0.000490	7 (0.18%)
	18	7	--	0.000516	7 (0.18%)
	18	9	--	0.000498	6 (0.15%)
	18	2	--	0.000604	7 (0.18%)
	20	6	--	0.000581	8 (0.20%)
	20	8	--	0.000582	7 (0.18%)
	20	4	--	0.000450	6 (0.15%)
	22	5	--	0.000341	5（0.13%）
With a RBF layer and two sigmoid layers of cascade	8	8	8	0.000314	5 (0.13%)
	10	7	2	0.000574	10 (0.25%)
	12	3	2	0.000290	2 (0.05%)
	12	6	8	0.000405	1 (0.03%)
	14	6	2	0.000768	1 (0.03%)
	16	3	2	0.000274	3 (0.07%)
	16	5	4	0.000964	3 (0.07%)
	18	5	4	0.000654	4 (0.10%)
	20	6	7	0.000824	5 (0.13%)

TABLE IV. EXPERIMENTAL COMPARISON OF DIFFERENT NETWORK STRUCTURES ON THE DOUBLE MOON DATASET

Network structure	**No. of sigmoid layers**	**No. of sigmoid/Gaussian kernels**	**Training error**	**Testing misclassification**
BP network	1	12	0.0257	31(%0.77)
BP network	2	8,6	0.005438	27(%0.67)
RBF network	--	20	0.0267	34(%0.85)
Presented network	1	14,5	0.000456	2(%0.05)
The presented network	2	12,6,8	0.000405	1 (%0.08)

Fig. 5 and Table IV shows the comparison of the classification effect of the presented network with BP network and RBF network on the testing sample set. Compared with BP network and RBF network, the presented network has higher classification accuracy when the learning effect of training sample set is improved.

B. Twist classification problem

Fig. 6 gives the graphical display of the Double moon classification dataset. Fig. 7 shows the comparison of the mean square error learning curve between the presented network, BP network and RBF network under the Twist classification dataset. It can be seen that under the complex sample set, the advantages of the presented network are further validated. Compared with the single BP network and RBF network, the learning effect of the presented network structure on the training sample set has obvious advantages. Table 5 further validates that when the parameters of presented netwrok change, the overall network can remain a stable performance .

On the premise that the mean square error of training set is improved, Fig. 8 and Table VI show the presented network has higher classification accuracy than the BP netwrok and RBF netwrok obviously. The advantages of the BP network with kernel fusion presented in this paper are verified.

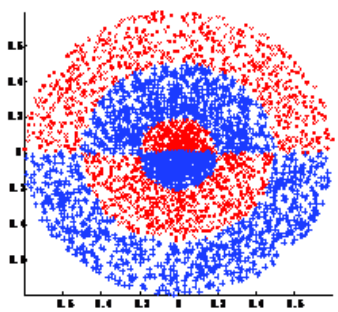
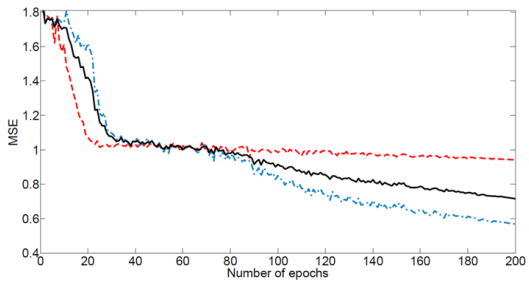


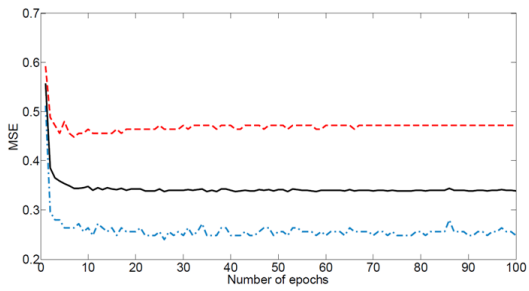
Figure 6. Twist classification dataset

TABLE V. PERFORMANCE COMPARISON OF DIFFERENT PARAMETERS OF HYBRID NEURAL NETWORK BASED ON KERNEL STRUCTURE FUSION ON THE TWIST DATASET

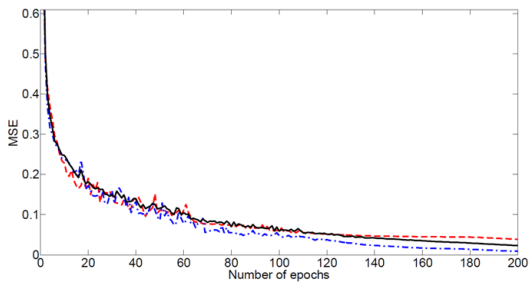
The presented network	**No. of Gaussian kernels**	**No. of sigmoid kernel**		**Training error**	**Testing misclassification**
		First layer	***Second layer***		
With a RBF layer and a single sigmoid layer of cascade	10	9	--	0.042473	216 (5.40%)
	12	6	--	0.065661	245 (6.13%)
	12	7	--	0.062862	232 (5.80%)
	14	4	--	0.130702	295 (7.38%)
	14	6	--	0.084620	247 (6.17%)
	14	9	--	0.080650	221 (5.53%)
	16	7	--	0.071156	202 (5.05%)
	16	5	--	0.047947	239 (5.97%)
	16	8	--	0.047947	239 (5.97%)
	18	5	--	0.093920	214 (5.35%)
	18	9	--	0.030399	208 (5.20%)
	20	6	--	0.053798	203 (5.08%)
	20	8	--	0.053590	277 (6.93%)
	22	7	--	0.037089	216 (5.40%)
	22	9	--	0.063314	209 (5.22%)
	24	8	--	0.047853	230 (5.75%)
	30	7	--	0.02083	203 (5.08%)
With a RBF layer and two sigmoid layers of cascade	10	8	8	0.030517	184 (4.60%)
	12	5	3	0.102097	226 (5.65%)
	12	6	4	0.070572	233 (5.83%)
	12	7	5	0.035021	165 (4.13%)
	14	7	5	0.032849	176 (4.40%)
	14	5	7	0.076504	239 (5.97%)
	14	8	4	0.020085	152 (3.80%)
	16	4	7	0.074303	228 (5.70%)
	16	5	5	0.066062	173 (4.32%)
	16	7	4	0.034389	173 (4.32%)
	18	5	8	0.049008	165 (4.13%)
	18	6	8	0.039821	153 (3.82%)
	20	5	7	0.035847	153 (3.82%)
	20	8	4	0.034634	153 (3.82%)
	24	5	6	0.090377	177 (4.42%)



(a)



(b)

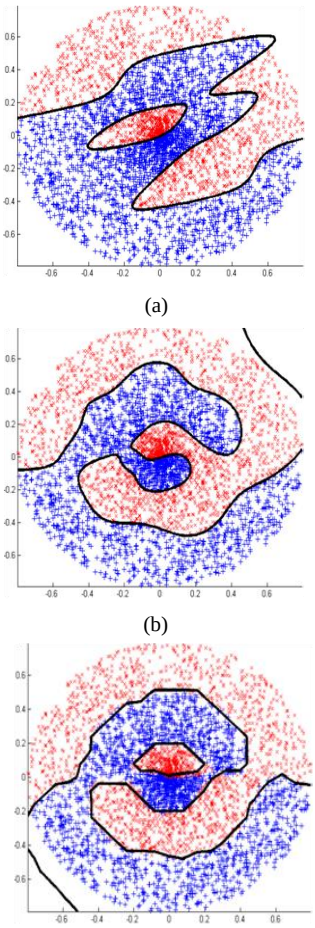


(c)

Figure 7. Comparison of mean square error learning curves of different networks on Twist dataset. (a) BP network (b) RBF network (c) Hybrid Neural Network Based on Kernel Structure Fusion

C. UCI benchmark datasets classification problem Twist

Table VII gives the performance comparison among the BP network, RBF network and the presented network under UCI benchmark datasets, where the number of sigmoid layers in the BP network and the presented network is 1. The results show that the presented network structure has better



(a)

(b)

(c)

Figure 8. Comparison of classification effects of different networks on the Twist dataset.(a) BP network (b)RBF network (c) Hybrid Neural Network Based on Kernel Structure Fusion

classification performance. The advantages of the presented hybrid neural network based on kernel structure fusion are further fully verified.

TABLE VI. EXPERIMENTAL COMPARISON OF DIFFERENT NETWORK STRUCTURES ON THE DOUBLE MOON DATASET

Network structure	No. of sigmoid layers	No. of sigmoid/Gaussian kernels	Training error	Testing misclassification
BP network	1	50	0.734741	836 (20.90%)
BP network	2	15,7	0.417626	672 (16.80%)
RBF network	--	36	0.2960	365 (9.13%)
The presented network	1	20,6	0.053798	203 (5.08%)
The presented network	2	18,6,8	0.039821	153 (3.82%)

TABLE VII. PERFORMANCE COMPARISON OF DIFFERENT NETWORK ON UCI BENCHMARK DATASETS

Datasets	Network structure	No. of sigmoid/Gaussian kernels	Testing accuracy
Diabetes	BP network	7	38.62%
	RBF network	20	76.25%
	Presented network	20, 7	79.71%
GC	BP network	8	22.92%
	RBF network	52	73.75%
	Presented network	40, 8	79.26%
Ionosphere	BP network	7	73.30%
	RBF network	38	88.24%
	Presented network	30, 7	91.42%

V. CONCLUSION

In this paper, a hybrid network based on kernel structure fusion is presented. By establishing the effective connection mechanism between the Gaussian kernels with different parameters and sigmoid kernels in different hidden layers, the fused network can effectively extract the local characteristics of the spatial distribution of samples, which combines the advantages of RBF network and BP network, and reduce the dependence on parameters selection of hidden nodes in single RBF and BP networks effectively. Experiments on artificial data sets and benchmark data sets show that the presented method can significantly improve the generalization performance and convergence speed of BP network.

ACKNOWLEDGMENT

This paper is supported by the National Natural Science Foundation of China (No. 61741111), Natural Science Foundation of Fujian Province (No. 2019J01815, 2019J01816), New Century Excellent Talents in Fujian Province University (2018, Yantao), Natural Science Foundation of Fujian Province (No. 20181BAB202011, 20161BAB212031), Fujian Middle-aged and Young Teachers'Education and Scientific Research Project (JT180486, JAT170504), Putian Science and Technology Bureau (2018ZP10, 2018RP4004) and Putian University's Initiation Fee for Introducing Talents into Scientific Research (2018088).

REFERENCES

[1] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations of back-propagating errors, ” Nature, vol. 323, pp. 533-536, 1986.

[2] D. S. Broomhead,and D. Lowe, “Multivariable function interpolation and adaptive network, ” Complex Systems, vol. 2, pp. 321-355, 1988.

[3] Y. Liu, J. Yang, L. Li and W. Wu, “Negative effects of sufficiently small initial weights on back-propagation neural networks, ” Journal of Zhejiang University-SCIENCE C-Computers & Electronics, vol. 13, pp. 585-592, 2012.

[4] S. F. Ding, C. Y. Su and J. Z. Yu, “An optimizing BP neural network algorithm based on genetic algorithm, ” Artificial Intelligence Review, vol. 36, pp. 153-162, 2011.

[5] A. Bhaya, and E.Kaszkurewicz, “Steepest descent with momentum for quadratic functions is a version of the conjugate gradient method, ” Neural Networks, vol. 17, pp. 65-71, 2004.

[6] J. E. Vetela and J. Reifman, "Premature saturation in back-propagation networks: mechanism and necessary conditions, " Neural Networks, vol. 10, pp. 721-735, 1997.

[7] C. H. Chen, T. K. Yao, C. M. Kuo and C.Y. Chen, "Evolutionary design of constructive multilayer feedforward neural network," Journal of Vibration and Control, vol. 19, pp. 2413-2420, 2013.

[8] S. Amari, "A theory of adaptive pattern classifiers, " IEEE Transactions on Electronic Computers, vol. 16, pp. 299-307, 1967.

[9] L. Yingwei, N. Sundararajan and P. Saratchandran, "A sequential learning scheme for function approximation using minimal radial basis function," Neural Computation, vol. 9, pp. 461–478, 1997.

[10] G-B. Huang, P. Saratchandran and N. Sundararajan, "A generalized growing and pruning RBF (GAP-RBF) neural network for function approximation, " IEEE Trans. Neural. Netw. vol. 16, 57-67, 2005.

[11] H. Yu, P. D. Reiner, T. Xie, T. Bartczak and B. M. Wilamowski, "An incremental design of radial basis function networks, " IEEE Trans. Neural Netw. and Learning Systems, vol. 2, pp. 1793–1803, 2014.

[12] S. Suresh, D. Keming, H. J. Kim, "A sequential learning algorithm for self-adaptive resource allocation network classifier," Neurocomputing, vol. 73, pp. 3012-3019, 2010.

[13] J. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing, " Neural Computation, 1989, vol. 1, pp. 281-294, 1989.

[14] A. D. Niros and G. E. Tsekouras, "A novel training algorithm for RBF neural network using a hybrid fuzzy clustering approach," Fuzzy Sets and Systems, vol. 193, pp. 62-84, 2012.

[15] S. Chen, X. Hong, C. J. Harris and L. Hanzo, "Fully complex-valued radial basis function networks: Orthogonal least squares regression and classification, " Neurocomputing, vol. 71, pp. 3421-3433, 2008.

[16] R. Mohammadi, S. Ghomi and F. Zeinali, "A new hybrid evolutionary based RBF networks method for forecasting time series: A case study of forecasting emergency supply demand time series," Engineering Applications of Artificial Intelligence, vol. 36, pp. 204-214, 2014.

[17] V. N. Ghate and S. V. Dudul, "Cascade Neural-Network-Based Fault Classifier for Three-Phase Induction Motor, " IEEE Transactions on Industrial Electronics, vol.58, pp.1555-1566, 2011.

[18] M. S. S. Abujazar, S. Fatihah, I. A. Ibrahim, A. E. Kabeel and Sharil S, "Productivity modelling of a developed inclined stepped solar still system based on actual performance and using a cascaded forward neural network model, " Journal of Cleaner Production, vol. 170, pp. 147-159, 2018.

[19] Y. LeCun, Y. Bengioand, and G. Hinton, "Deep learning," Nature, vol. 521, pp. 436-444, 2015

[20] H.D. Shao, H.K. Jiang, H. Zhang, and T.C. Liang, "Electric Locomotive Bearing Fault Diagnosis Using a Novel Convolutional Deep Belief Network, " IEEE Transactions on Industrial Electronics, vol. 65, pp. 2727-2736, 2018.

[21] S. Qian, H. Liu, C. Liu, S. Wu and H. S. Wong, "Adaptive activation functions in convolutional neural networks," Neurocomputing, vol. 272, pp. 204-212, 2018.

[22] J. X. Tang and G. B. Huang, "Extreme Learning Machine for Multilayer Perceptron," IEEE Transactions on Neural Networks and Learning Systems, vol. 27, pp. 809-821, 2016.

[23] M. Lehtokangas, "Feedforward neural network with adaptive reference pattern layer, " International Journal of Neural Systems, vol. 9, pp.1-9, 1999.

[24] S. Hayin. "Neural networks and learning machines (Third Edition), " Beijing: China Machine Press, China, 2009.

[25] C. Blake and C. Merz. "UCI repository of machine learning databases, " Department of Information and Computer Sciences, University of California, Irvine. Available at *http://archive.ics.uci.edu/ml/35*, 1998.