Available online at www.sciencedirect.com
ScienceDirect
Neurocomputing 70 (2006) 592–596
NEUROCOMPUTING
www.elsevier.com/locate/neucom

Letters

A novel full structure optimization algorithm for radial basis probabilistic neural networks

Ji-Xiang Du[a,b,c,*], De-Shuang Huang[c], Guo-Jun Zhang[c], Zeng-Fu Wang[b]

[a]*Department of Computer Science and Technology, Huaqiao University, PR China*
[b]*Department of Automation, University of Science and Technology of China, PR China*
[c]*Intelligent Computing Lab, Hefei Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei, Anhui 230031, PR China*

Received 17 September 2005; received in revised form 8 May 2006; accepted 8 May 2006
Communicated by R.W. Newcomb
Available online 3 June 2006

Abstract

In this paper, a novel full structure optimization algorithm for radial basis probabilistic neural networks (RBPNN) is proposed. Firstly, a minimum volume covering hyperspheres (MVCH) algorithm is proposed to heuristically select the initial hidden layer centers of the RBPNN, and then the recursive orthogonal least square (ROLS) algorithm combined with the particle swarm optimization (PSO) algorithm is adopted to further optimize the initial structure of the RBPNN. Finally, the effectiveness and efficiency of our proposed algorithm are evaluated through a plant species identification task involving 50 plant species.

Keywords: Radial basis probabilistic neural networks; Full structure optimization; Minimum volume covering hyperspheres algorithm

1. Introduction

The radial basis probabilistic neural networks (RBPNN) model, as shown in Fig. 1, integrates the advantages of radial basis function neural networks (RBFNN) and probabilistic neural networks (PNN), and avoids or reduces the disadvantages of the RBFNN and the PNN [2]. The construction of an RBPNN involves four different layers: one input layer, two hidden layers and one output layer. The first hidden layer is a nonlinear processing layer, which generally consists of hidden centers selected from a training samples set. The second hidden layer selectively sums the outputs of the first hidden layer, which generally has the same size as the output layer for a labeled pattern classification problem. In general, the weights between the first and the second hidden layer are set as fixed values (1 or 0) and do not require learning. Generally, the first hidden layer is tightly interrelated to the performance of the RBPNN.

Just as for the RBFNN, in the first hidden layer of the RBPNN, the hidden centers number and locations as well as the controlling parameters of the kernel function are quite important indices. Too many hidden centers will lead to very lengthy training and testing time, and poor generalization capability, while, too few hidden centers can lead to quite great convergent error. In addition, the selected hidden centers will require especial controlling parameters in order to realize the entire overlay of training samples in space. The tightly correlative characteristic between the hidden centers and controlling parameters shows that while investigating the structure optimization for the RBPNN, the hidden centers (including number and locations) and the controlling parameters must be simultaneously considered. Therefore, this paper will discuss how to optimize the full structure of the RBPNN to improve the classification performance and generalization capability of the networks.

This paper is organized as follows: in Section 2, the full structure optimization algorithm for RBPNN is discussed in details. The experimental results are presented in Sections and 4 concludes the whole paper and gives related conclusions.

*Corresponding author. Intelligent Computing Lab, Hefei Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei, Anhui 230031, PR China.
E-mail address: du_jx@iim.ac.cn (J.-X. Du).

0925-2312/$ - see front matter
doi:10.1016/j.neucom.2006.05.003

2. Optimizing RBPNN using MVCH and ROLS combined with PSO

2.1. Heuristically initializing the hidden centers by minimum volume covering hyperspheres (MVCH) algorithm

Considering a q-class pattern classification problem with N sample data points in a d-dimensional feature space, denote the data points $\mathbf{a}^i = (a_1^i, a_2^i, \ldots, a_d^i)$, where i indexes the data point number. A number of hyperspheres will be designed to cover the data points of the considered class as compact as possible. To achieve that, we should obtain a minimum volume covering hypersphere (MVCH) that encompasses all points of the same class. This can be derived by posing the problem as a quadratic optimization problem:

$$\text{Min} \;\; R = r^2 \;\; \text{s.t.} \;\; \|\mathbf{x} - \mathbf{a}_i\|_2^2 \leqslant R, \quad i = 1, 2, \ldots, N, \tag{1}$$

where $x = (x_1, x_2, \ldots, x_d)$ is the hypersphere center and r its radius. This is a convex optimization problem and can be solved using the dual form [6]

$$\text{Min} \quad L_D = \sum_{j=1}^{d} \left(\sum_{i=1}^{N} \lambda_i a_j^i \right)^2 - \sum_{j=1}^{d} \sum_{i=1}^{N} \lambda_i \left(a_j^i \right)^2$$
$$\text{s.t.} \quad \sum_{i=1}^{N} \lambda_i = 1, \quad \lambda_i \geqslant 0, \quad i = 1, 2, \ldots, N, \tag{2}$$

$$x_j = \sum_{i=1}^{N} \lambda_i a_j^i, \quad j = 1, 2, \ldots, d. \tag{3}$$

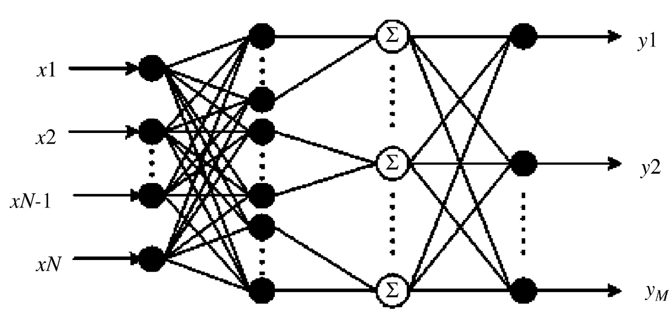

Fig. 1. The topology scheme of the RBPNN.

Then, the steps of the MVCH algorithm can be described as follows. An illustration of the algorithm is shown in Fig. 2:

Step 1: set $k = 1$, $C = 1$.

Step 2: set point set $P_S =$ all set of points of class C.

Step 3: find the hypersphere K that encompasses points in P_S, and the furthest point $\mathbf{y}$ different from class C from the center, which is encompassed by the hypersphere K; compute its distance d_y from the center; drop points from P_S, whose distances from the center are greater than or equal to $d_s = \eta d_y$, $\eta \subset (0, 1)$.

Step 4: repeat Steps 3 until there are no points that need to be dropped and remove those points encompassed by hypersphere K from P_S.

Step 5: if P_S is not empty, set $k = k + 1$ and go to *Step* 2; else if $C < q$, Set $C = C + 1$ and go to *Step* 2; else end.

2.2. Selecting the hidden centers by the recursive orthogonal least square (ROLS) algorithm

Assume that $\mathbf{Y}_d$, $\mathbf{H}$, $\mathbf{W}$ and $J(\mathbf{W})$, respectively, denote the desired signal matrix, the output matrix of the second hidden layer, the weight matrix between the second hidden layer and the output layer and the cost function of the RBPNN. In the form of Euclidean norm, the cost function of the RBPNN can be given by

$$\mathbf{Y}_W = \mathbf{Y}_d - \mathbf{H}\mathbf{W},$$
$$J(\mathbf{W}) = \|\mathbf{Y}_W\|_2^2 = \sum_{i=1}^{d} y_{Wi}^2. \tag{4}$$

By conducting the orthogonal decomposition operation, we have

$$\mathbf{H} = \mathbf{Q}[\mathbf{R}, 0]^{\mathrm{T}},$$
$$\mathbf{Q}^{\mathrm{T}}\mathbf{Y}_d = \left[\widehat{\mathbf{Y}}\;\tilde{\mathbf{Y}}\right], \tag{5}$$

$$J(\mathbf{W}) = \left\| Q \left\{ \begin{bmatrix} \widehat{\mathbf{Y}} \\ \tilde{\mathbf{Y}} \end{bmatrix} - \begin{bmatrix} \mathbf{R} \\ 0 \end{bmatrix} \mathbf{W} \right\} \right\|_2^2 = \left\| \widehat{\mathbf{Y}} - \mathbf{R}\mathbf{W} \right\|_2^2 + \left\| \tilde{\mathbf{Y}} \right\|_2^2, \tag{6}$$

where $\left\| \tilde{\mathbf{Y}} \right\|_2^2$ is the residual error (RE) of $J(\mathbf{W})$, and it is also written as

$$E_R = \|\mathbf{Y}_d - \mathbf{H}\mathbf{W}\|_2^2 = \left\| \tilde{\mathbf{Y}} \right\|_2^2. \tag{7}$$

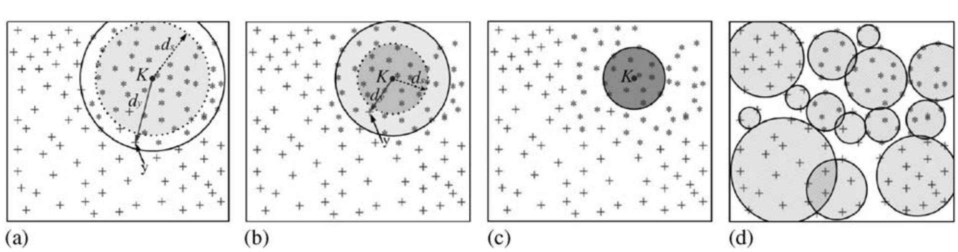

Fig. 2. Illustration of the MVCH algorithm: (a) find the minimum volume hypersphere K encompassing points in P_S, the farthest point $\mathbf{y}$ of a different class from the center, then drop points from P_S whose distances from the center $d_i \geqslant d_S = \eta d_y$; (b) repeat the above procedure; (c) the first obtained hypersphere there are no points need to be dropped; and (d) hyperspheres obtained using MVCH.

For pattern classification problems, the classification error (CE) is defined as

$$E_C = \left\| \mathbf{Y}_d - round(\mathbf{HW}) \right\|_2^2, \tag{8}$$

where $round(\cdot)$ is the round operation. Both the RE and CE are adopted into our algorithm, which is also called as the double error criterion. In order to decrease the computational complexity, a recursive algorithm is introduced to obtain the updating **W** and the double errors. The further details can be referred to the literature [3].

2.3. Solving the controlling parameter by the particle swarm optimization (PSO)

For the recursive orthogonal least square (ROLS) algorithm in literature [3], the controlling parameter must be given beforehand, which can cause that the finally selected hidden centers are the optimal combinations only matching the pre-given controlling parameter. Generally, the controlling parameter is a function of many relative factors, and it is difficult to solve using the traditional methods. Therefore, to solve the optimal controlling parameter matching the currently selected hidden centers, the use of the PSO, a relatively new population-based evolutionary computation technique [4], is here preferred. To decrease the computational cost, assume that only one controlling parameter is used without any prior knowledge. And the corresponding fitness function f, in this paper, is defined as RE: $f = E_R$.

2.4. Summary of the MVCH–ROLS–PSO

Consequently, the implementation procedure for our proposed MVCH–ROLS–PSO algorithm is described as follows.

Step 1: Initially select the hidden centers of the RBPNN by the MVCH method.

Step 2: By the PSO, compute the optimal controlling parameter matching the currently selected hidden centers.

Step 3: When those currently selected hidden centers are removed independently one by one, compute the respective RE of the RBPNN consisting of the left hidden centers, then find out the remind hidden centers corresponding to the minimum RE, and again compute the CE. Moreover, the hidden centers of combinations with the minimum RE are also used as the initial hidden centers of the next iteration.

Step 4: If only one hidden center exists in each class, go to next step; else, go to Step 2.

Step 5: Search for the final CE turning point, which corresponds to the optimal structure of RBPNN, then exit.

3. Experimental results

To evaluate our proposed algorithm, the RBPNN was applied to a plant's species identification task through plant's leaves. The leaf image database used in the following experiment was collected and built by our lab, which includes 50 species of different plants, 600 images. Each species includes at least 10 leaves of images, 5 of which are randomly selected as training samples, and the rest are used as the test samples. So there are 250 training samples. A Gabor texture description method from the literature [5] was applied, and five scales and six orientations were used to extract 60 features.

First of all, the RBPNN without optimization was directly applied, where all the 250 training samples were selected as the hidden centers of the first hidden layer and the controlling parameter was set to 0.4. In addition, with the same training and test samples, other classifiers of the k k-nearest neighbor classifier (k-NN), the multi-layer perceptron networks (MLPN) [1], and the RBFNN [1] were also used to compare with the RBPNN. Note that all the classifiers are programmed with C++. And all the parameters and the experimental results were shown in Table 1. Thus, it can be seen that the classification performance of the RBPNN is higher than other classifiers; moreover, the training speed and test speed of the RBPNN are also very fast.

Table 1
Classification performance comparison for the plant species identification between the RBPNN and other classifiers of the MLPN, the RBFNN, the k-NN

Classifiers		Number of hidden nodes		Control parameter	Recognition rate (%)	CPU time (s)	
		1st HL	2nd HL			Training	Classification
RBPNN	Not optimized	250	50	0.400	84.4	1.614	0.008
	Optimized	87	50	0.419	84.9	1.020	0.006
RBFNN	Not optimized	250	50	0.400	84.1	2.030	0.013
	Optimized	97	50	0.409	84.5	1.263	0.010
MLPN	One hidden layer	100	—	—	80.6	279	0.083
	Two hidden layers	100	50	—	79.2	456	0.134
k-NN	$k = 1$	—	—	—	81.6	—	2.360
	$k = 4$	—	—	—	82.9	—	3.058

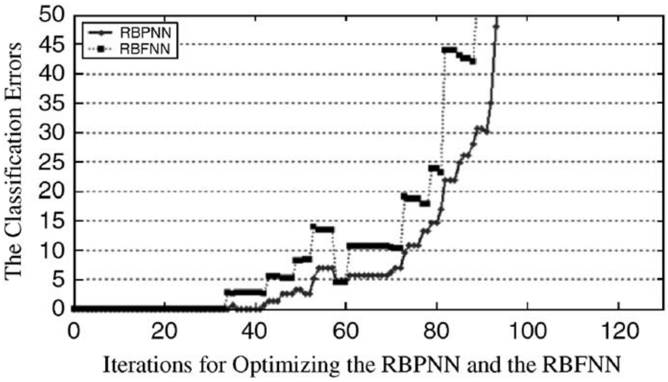

Fig. 3. The curve of CEs versus iterations for optimizing the RBPNN and the RBFNN by the MVCH–ROLS–PSO.

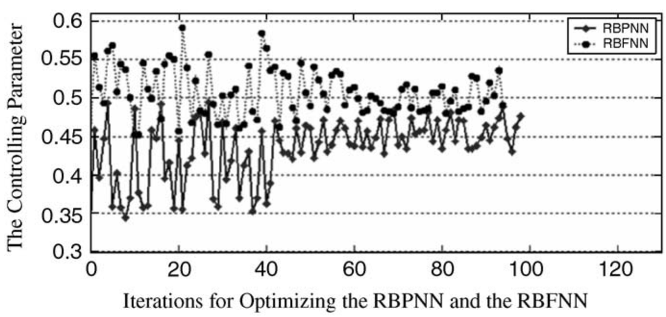

Fig. 4. The learning curve of the controlling parameters versus iterations for optimizing the RBPNN and the RBFNN by the MVCH–ROLS–PSO.

Then, the MVCH–ROLS–PSO was applied to optimize the structures of the RBPNN and the RBFNN. Note that after clustering by the MVCH method, there were 119 training samples selected from all the 250 ones as the initial hidden center vectors. The corresponding experimental results were shown in Table 1, Figs. 3 and 4, from which, it can be seen that the MVCH–ROLS–PSO is feasible and efficient to optimize the RBPNN and the RBFNN. But compared with the optimized structure of the RBFNN, the optimization ratio of structure for the RBPNN is higher.

Finally, to further validate and compare the generalization capability of the RBPNN and the RBFNN optimized by our proposed algorithm, the test sample sets formed by the original test samples, mixed with 20 groups of Gaussian zero mean white noises with the variances at the interval of $(0, 0.2)$, were used. Consequently, the experimental results were plotted in Fig. 5. Obviously, Fig. 5 showed that in the sense of generalization capability, the RBPNN optimized by our proposed algorithm is better than the RBFNN optimized by the same algorithm. Moreover, the RBPNN also has better performance for noise toleration.

4. Conclusions

This paper proposes a novel MVCH–ROLS–PSO approach to entirely optimize the structure of RBPNN, which is a new improvement strategy based on the original

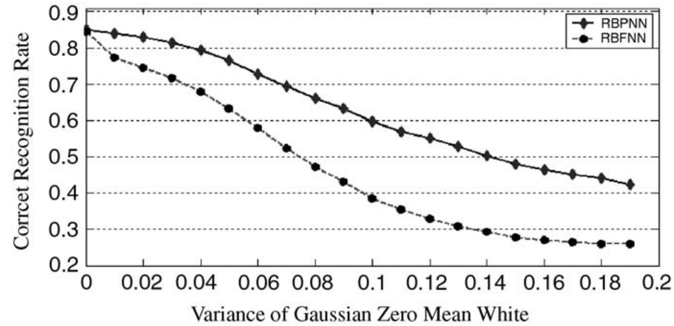

Fig. 5. The generalization capability comparison between the optimized RBPNN and the RBFNN by the MVCH–ROLS–PSO.

ROLS method [3]. The advantage of the proposed approach is that the structure of the RBPNN can be heuristically initialized; moreover, both the hidden centers and the controlling parameter can be entirely and simultaneously optimized. The experimental results show that our proposed MVCH–ROLS–PSO algorithm is feasible and efficient to optimize the RBPNN. In particular, the experimental results also illustrate that the generalization performance of the optimized RBPNN by the MVCH–ROLS–PSO was obviously better than the one of the optimized RBFNN.

Acknowledgments

This work was supported by the NSF of China (Nos. 60472111 and 60405002).

References

[1] D.S. Huang, Systematic Theory of Neural Networks for Pattern Recognition, Publishing House of Electronic Industry of China, Beijing, May 1996.
[2] D.S. Huang, Radial basis probabilistic neural networks: model and application, Int. J. Pattern Recogn. Artif. Intell. 13 (7) (1999) 1083–1101.
[3] D.S. Huang, W.B. Zhao, Determining the centers of radial basis probabilistic neural networks by recursive orthogonal least square algorithms, Appl. Math. Comput. 162 (1) (2005) 461–473.
[4] C.T. Ioan, The particle swarm optimization algorithm: convergence analysis and parameter selection, Inform. Process. Lett. 85 (6) (2003) 317–325.
[5] B.S. Manjunath, W.Y. Ma, Texture features for browsing and retrieval of large image data, IEEE Trans. Pattern Anal. Mach. Intell. 18 (8) (1996) 837–842.
[6] P. Wolfe, A duality theorem for nonlinear programming, Q. Appl. Math. 19 (3) (1961) 239–244.

Ji-Xiang Du took courses as B.Sc. degree candidate from September 1995 to July 1999, in Vehicle Engineering, Hefei University of Technology, and obtained B.Sc. degree in July 1999. From September 1999 to July 2002, took courses as M.Sc. degree candidate in Vehicle Engineering, Hefei University of Technology, and obtained B.Sc. degree in July 2002. From February 2003 on, in pursuit for Ph.D. degree in Pattern Recognition & Intelligent System in University

of Science and Technology of China (USTC), Hefei, China, and in December 2005, he received Ph.D. degree. Now, he is engaged in the postdoctoral study in Department of Automation of USTC.

De-Shuang Huang (SM'98) is a Professor and Ph.D. Advisor in the University of Science and Technology of China (USTC), Hefei, China, and Professor of Graduate School of the Institute of Intelligent Machines, Chinese Academy of Sciences (CAS). From September 2000 to March 2001, he worked as Research Associate in Hong Kong Polytechnic University. From April 2002 to June 2003, he worked as Research Fellow in City University of Hong Kong. From October to December 2003, he worked as Research Fellow in Hong Kong Polytechnic University. From July to December 2004, he worked as the University Fellow in Hong Kong Baptist University. Dr. Huang is currently a senior member of the IEEE. He has published over 190 papers.

Guo-Jun Zhang is a postgraduate of Master in Pattern Recognition & Intelligent System in University of Science and Technology of China (USTC), Hefei, Anhui, China. In July 2004, he obtained B.Sc. degree on Computer Science in Nanjing Normal University. His Research Interests are in Pattern Recognition and Image Processing.

Zeng-Fu Wang is a Professor and Ph.D. Advisor in the University of Science and Technology of China (USTC), Hefei, Anhui, China. In June 1992, he obtained Ph.D. degree on control engineering in Osaka university of Japan. The research work focuses on 3D computer vision, emotional computation, and intelligent robot.