A New Radial Basis Probabilistic Neural Network Model

Huang Deshuang Ma Songde
National Laboratory of Pattern Recognition, Institute of Automation
, Chinese Academy of Sciences, Beijing, 100080, P. R. China
masd@prlsun2. ia. ac. cn huangds@prlsun2. ia. ac. cn

Abstract: This paper proposes a new neural network model called radial basis probabilistic neural network (RBPNN) based on the radial basis function network (RBFN) and the probabilistic neural network (PNN). This new model inherits the merits of the two old models and tries to avoid their defects. Finally, the experimental results show that this new model is effective and practical.

1 Introduction

Generally speaking, the training speed of RBFNs is faster than the one of MLPNs in the case which the center vectors in the former hidden units have been decided [1, 2, 3]. But the link weight vectors in the PNNs needn't be trained, the classification testing can be immediately realized [4]. However, the PNNs have two drawbacks [4]: (1). the PNNs have no consideration of the neighbouring and overlapping among pattern sample vectors from different categories, thus the formed discrimination surface is biased; (2). All training vectors must be used as center vectors in thehidden layer, which becomes difficulty for hardware realization, especially, for huge number of training samples. On the other hand, RBFNs can well solve the first problem, and make the second problem lessened, but the training speed is not satisfying. So based on the two models, one new model, called radial basis probabilistic neural network (RBPNN), is proposed to make a compromise between the total performance.

2 Radial Basis Probabilistic Neural Network Model

The first layer of the RBPNN is identical with the one of the RBFNs. The transfer functions, like RBFN and PNN, meet with the condition of Parzen window function. The weight vectors in the first layer of the RBPNN are equal to center vectors $\mathbf{c}_i (i = 1, 2, \cdots, H_1)$ in radial basis function:

$$\mathbf{w}_i^{(1)} = \mathbf{c}_i \quad \mathbf{w}_i^{(1)} \in R^n \quad i = 1, 2, \cdots, H_1 \quad (1)$$

where H_1 is the node number in the first hidden layer. Assume that the transfer function in the first hidden layer is $K(\cdot)$, the output in the i-th node of the first hidden layer is:

$$y_i = K\left(\frac{\mathbf{x} - \mathbf{c}_i}{a}\right) \qquad i = 1, 2, \cdots, H_1 \quad (2)$$

where a is a parameter controlling the distributing shape.

The second layer of the RBPNN is identical with the one of the PNNs. The node of the second layer makes a sum calculation and forms a link with the selective nodes of the foregoing (first) layer, The weight vectors in the second layer of the RBPNN are all equal to 1:

$$\mathbf{w}_i^{(2)} = 1 \qquad i = 1, 2, \cdots, H_2 \quad (3)$$

where H_2 is the node number in the second hidden layer; the dimensional number of the $\mathbf{w}_i^{(2)}$ depends on the node number of the upper layer (the first hidden layer) with which the i-th node of the second hidden layer links. Assume the i-th output of the second hidden layer is z_i,

0-7803-2912-0

then the m-th output of the third layer

$$u_m = \sum_{k=1}^{H_2} w_{mk}^{(3)} z_k \tag{4}$$

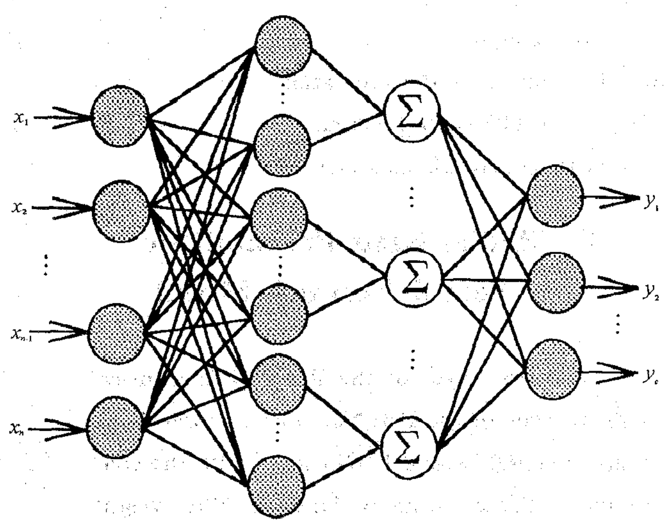

Fig. 1 Structure Scheme of the RBPNN Model

The third layer of the RBPNN is identical with the second one of the RBFNs, which is a one-layer linear perceptron. The training for the weight vectors in the third layer is equivalent to the one of the output layer in the RBFNs. But because of the fewer second hidden node number, the traing for the RBPNN is simpler than the one of the RBFNs. Fig. 1 gives the structure scheme of the RBPNN model.

For the pattern recognition problem for c categories, each category has $N_1, N_2, \cdots, N_c$ training sample vectors, respectively. Let $\sum_{i=1}^{c} N_i = N$. Supposed that the supervised signals, corresponding to the network outputs, are expressed as the form of 0 or 1, then the link weight matrix in the output layer will follow following equation:

$$\underbrace{\begin{bmatrix} z_{11}^{(1)} & z_{12}^{(1)} & \cdots & z_{1H_2}^{(1)} \\ \cdots & \cdots & \cdots & \cdots \\ z_{N_1 1}^{(1)} & z_{N_1 2}^{(1)} & \cdots & z_{N_1 H_2}^{(1)} \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ z_{11}^{(c)} & z_{12}^{(c)} & \cdots & z_{1H_2}^{(c)} \\ \cdots & \cdots & \cdots & \cdots \\ z_{N_c 1}^{(c)} & z_{N_c 2}^{(c)} & \cdots & z_{N_c H_2}^{(c)} \end{bmatrix}}_{X} \underbrace{\begin{bmatrix} w_{11}^{(3)} & w_{21}^{(3)} & \cdots & w_{c1}^{(3)} \\ w_{12}^{(3)} & w_{22}^{(3)} & \cdots & w_{c2}^{(3)} \\ \cdots & \cdots & \cdots & \cdots \\ w_{1H_2}^{(3)} & w_{2H_2}^{(3)} & \cdots & w_{cH_2}^{(3)} \end{bmatrix}}_{W^{(3)}} = \underbrace{\begin{bmatrix} 1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}}_{D} \tag{5}$$

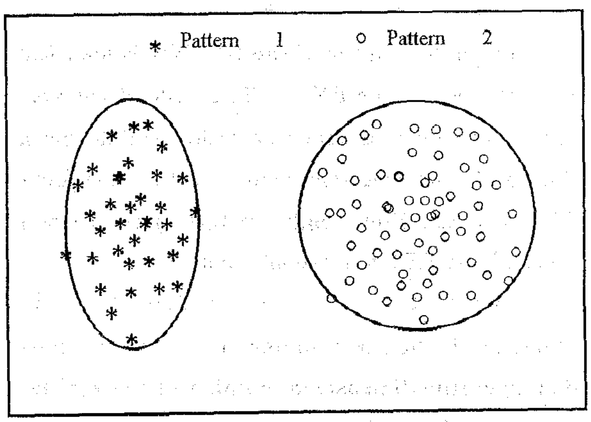

Fig. 2 Connected or Dense Scattered Pattern

The above formula can be rewritten as:

$$XW^{(3)} = D \tag{6}$$

So, the error cost function can be defined as:

$$E(W^{(3)}) = \|D - XW^{(3)}\|_F^2 \tag{7}$$

For a connected or dense scattered pattern set (as shown in Fig. 2), those typical samples from all training samples in each category space can be extracted and used as the center vectors in the first hidden layer. Assume that the typical sample number in c categories is $M^{(1)}, M^{(2)}, \cdots, M^{(c)}$, respectively, then the node numbers H_1, H_2 in the first and second hidden layer of the RBPNN are:

$$H_1 = \sum_{k=1}^{c} M^{(k)} \tag{8}$$

$$H_2 = c \tag{9}$$

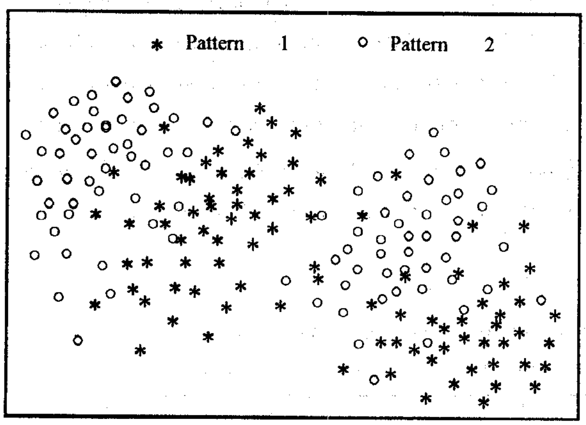

Fig. 3 Disconnected or Sparse Scattered Pattern

For a disconnected or sparse scattered pattern set (as shown in Fig. 3), those typical samples from different connected domains in each category space can be extracted and used as the center vectors in the first hidden layer. Assume that the typical sample number in c categories is $M_1^{(1)}, \cdots, M_{n_1}^{(1)}, M_1^{(2)}, \cdots, M_{n_2}^{(2)}, \cdots, M_1^{(c)}, \cdots, M_{n_c}^{(c)}$, respectively, where n_k is the connected domain number in the k-th category. Then the node numbers H_1, H_2 in the first and second hidden layer of the RBPNN are:

$$H_1 = \sum_{k=1}^{c} \sum_{j=1}^{n_k} M_j^{(k)} \tag{10}$$

$$H_2 = \sum_{k=1}^{c} n_k \tag{11}$$

Assume that the numbers of the input node, hidden node and output node in a RBFN are N, M, L, respectively, and total training sample number is S which is completely used as center vectors in the hidden layer of the RBFN and a PNN, and half used as the ones in the first hidden layer of the RBPNN, i.e., $H_1 = S/2$. In addition, let $H_2 = c$, then the complexities of computation (multiplication and division) of one iteration for three kinds of networks are shown in Table 1

Table 1 A Comparison of the Complexities of Computation (Multiplication and Division) and Training Speed of One Iterating for Three Kinds of Networks (N: Network, C: Content, T: Training or Testing, P: Parameter)

P \ C \ T \ N	RBFN (supervised)		PNN (self-supervised)		RBPNN (supervised)	
	training	testing	training	testing	training	testing
complexities of computation	$2S^2 + 3S + SN + L + 4$	$SN + S$	—	SN	$2L^2 + \frac{SN}{2} + 4L + 4$	$\frac{SN}{2} + L$
$S = 150$, $N = 30$, $L = 5$	49959	4650	—	4500	2324	2255

3 Experimental Results

We use the one-dimensional images of five kinds of planes by radar to verify the classification performance of the proposed new model — RBPNN. Assume the orientation of the planes' head to be 0°, the used data is from 0° ~ 100° of five kinds of planes (Plane4 has only − 16° ~ 16°). Assume that the dimension for each data vector is 30. The 30 training samples are selected every 3° (Plane4 every 0.5°). The testing samples of Plane1, Plane2, Plane3, Plane4, plane5 are 138, 150, 113, 162, 18, respectively[5]. We use the recursive least square algorithm [5, 6] to train the RBFN and RBPNN. After that, those testing samples are used to test the recognition results. Tab. 2 gives the rate of recognition of five kinds of planes for three kinds of networks.

Assume that the center vectors are composed of 20 samples from the 30 training ones, then there are 100 units in the first hidden layer of the RBPNN. Suppose the termination error to be 10^{-3}, we use the 30 training samples to train the RBFN and RBPNN by means of the recursive least square algorithm[5,6]. For the PNN, the total 150 training samples are used as center vectors in the hidden layer, the network needn't be trained, and only waits for testing. After that, those testing samples are used to test the performance of three kinds of networks. Table 2 gives the rate of recognition of five kinds of airplanes for three kinds of networks.

Table 2 The Rate of Recognition for Five Kinds of Airplanes Using Three Kinds of Feedforward Networks (RBFN, PNN, RBPNN) (A: Airplane, N: Network)

A / N	Airplane 1	Airplane 2	Airplane 3	Airplane 4	Airplane 5	$\overline{P}_D$ ①
RBFN	98.3%	97%	98.8%	99.6%	100%	98.74%
PNN	91.3%	92.5%	93.6%	90.4%	100%	93.56%
RBPNN	97.5%	98.4%	98.5%	99.6%	100%	98.8%

4 Conclusions

This paper proposes a new neural model called a radial basis probabilistic neural network. This model preserves the merits of the radial basis function network and probabilistic neural network, and reduces their drawbacks, which is proved by the analyses of Tab. 1 and the experimental results. However, in application, we must pay attention to the domains of pattern distribution so as to better select the center vectors in the first hidden layer.

Reference

[1] D. R. Hush and B. G. Horne, "Progress in supervised neural networks", IEEE Signal Processing Magazine, pp. 8 ~ 39, 1993.

[2] G. Vrckovnik, C. R. Carter, S. Haykin, "Radial basis function classification of impulse radar waveforms", IJCNN, San Diego, Vol. I, pp. 45 ~ 50, 1990.

[3] Rogers, Steven K. and others. An Introduction to Biological and Artificial Neural Networks. wpafb: afit, 1990.

[4] D. F. Specht, "Probabilistic neural networks", Neural Networks, Vol. 3, pp. 109 ~ 118, 1990.

[5] Lv Xiaoxing, Yuan Keping, Huang Deshuang, "A study of backpropagation learning algorithm of multilayer percetron networks based on recursive least squares", Proc. of the Sixth Japan-China International Conference on Cmputer Applications, sept. 16 ~ 18, 1994, Sapporo, Japan.

[6] D. S. Huang, The study of recognition technique based on one-dimensional image of radar targets. Ph.D dissertation, Xidian University, (1992).

① $\overline{P}_D$ denotes the averaging rate of recognition