A Radial Basis Function Neural Network Training Mechanism for Pattern Classification Tasks

Antonios D. Niros and George E. Tsekouras

Abstract This chapter proposes a radial basis function network learning approach for classification problems that combines hierarchical fuzzy clustering and particle swarm optimization (PSO) with discriminant analysis to elaborate on an effective design of radial basis function neural network classifier. To eliminate the redundant information, the training data are pre-processed to create a partition of the feature space into a number of fuzzy subspaces. The center elements of the subspaces are considered as a new data set which is further clustered by means of a weighted clustering scheme. The obtained cluster centers coincide with the centers of the network's basis functions. The method of PSO is used to estimate the neuron connecting weights involved in the learning process. The proposed classifier is applied to three machine learning data sets, and its results are compared to other relative approaches that exist in the literature.

Keywords Radial basis function neural networks • Hierarchical fuzzy clustering • Particle swarm optimization • Discriminant analysis

1 Introduction

Radial basis function (RBF) networks have been used in various cases such as classification [1–4], system identification [5], function approximation [6–8], nonlinear systems modeling [1, 9, 10], and pattern recognition [11–15]. The challenges in designing efficient RBF neural networks refer to the network's parameter estimation procedure. The parameters are the centers and the widths of the basis functions, and the neuron connection weights, as well. An approach able to efficiently train RBF networks is the implementation of fuzzy cluster analysis [6, 9, 16]. Fuzzy clustering attempts to identify the underlying structure of the training data, and then generates a distribution of the RBFs that better describe this structure. This distribution is

A.D. Niros (✉) • G.E. Tsekouras
Department of Cultural Technology and Communication, University of the Aegean, University Hill, 81100 Mytilene, Greece
e-mail: aneiros@aegean.gr

M.E. Celebi, K. Aydin (eds.), *Unsupervised Learning Algorithms*,
DOI 10.1007/978-3-319-24211-8_8

established by determining the appropriate values for the center elements and widths of the basis functions. Finally, to establish the input–output relationships, optimal values for the connection weights can be extracted in terms of a wide range of iterative optimization methods, such as back-propagation [1, 17], gradient descent [1], and particle swarm optimization (PSO) [4, 5, 18–25].

In this chapter we propose a three-staged approach to elaborate on an effective design of RBF neural network for classification problems. The first step uses fuzzy partitions to dismember the feature space into a number of subspaces. Since these subspaces come from direct processing of the available data, they code all the necessary information related to the data distribution in the input space. Therefore, in order to reduce the computational efforts we choose to elaborate them (instead of using directly the training data). To accomplish this task, we assign weights to each fuzzy subspace. The representatives (i.e. center elements) along with the corresponding weights of the aforementioned fuzzy subspaces are treated as a new data set, which is partitioned in terms of a weighted version of the fuzzy c-means. This process yields the network's basis function centers. The network's widths are determined by means of the centers. Finally, the connection weights are provided by applying the PSO.

The material is presented as follows. Section 2 describes the standard topology of a typical RBF neural network. In Sect. 3 we analytically describe the proposed methodology. The simulations experiments take place in Sect. 4. Finally, the chapter concludes in Sect. 5.

2 RBF Neural Network

The basic topology of an RBF network consists in sequence of an input layer, a hidden layer, and a linear processing unit forming the output layer. The standard topology of the RBF network is depicted in Fig. 1.

The hidden layer consists of a number of nodes, each of which corresponds to an RBF. Herein, the RBF will also be referred to as kernel function or simply kernel. The set of input–output data pairs is denoted as

$$Z = \{(\boldsymbol{x}_k, y_k) \in R^p x R^s : 1 \leq k \leq N\} \tag{1}$$

with $\boldsymbol{x}_k = \left[x_{k1}, x_{k2}, \ldots, x_{kp}\right]^T$ is the k-th input vector and $y_k \in \{1, 2, \ldots, s\}$ is the class the vector $\boldsymbol{x}_k$ belongs to.

In the case of the Gaussian type of RBFs, the basis functions have the form,

$$h_i\left(\boldsymbol{x}_k\right) = \exp\left(-\frac{\|\boldsymbol{x}_k - \boldsymbol{v}_i\|^2}{\sigma_i^2}\right) \tag{2}$$

where $\boldsymbol{v}_i$ $(1 \leq i \leq c)$ are the centers, σ_i $(1 \leq i \leq c)$ the respective widths, and c the number of hidden nodes. The estimated output of the network utilizes the linear regression functional:

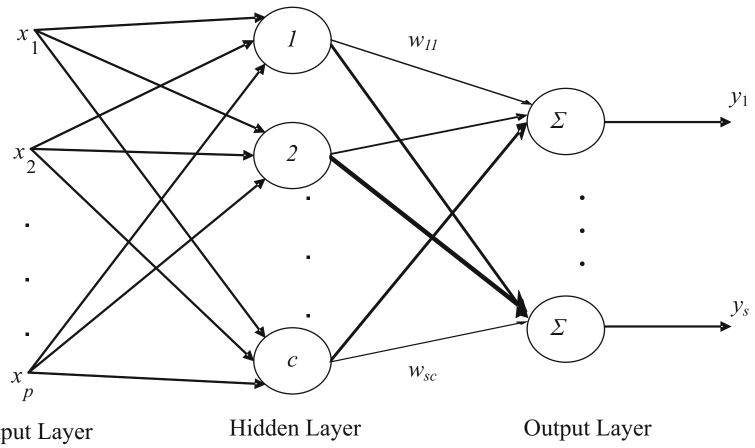


Fig. 1 Standard topology of a multi-input multi-output RBF neural network

$$\tilde{y}_j\left(\boldsymbol{x}_k\right) = \sum_{i=1}^{c} w_{ji} h_i\left(\boldsymbol{x}_k\right) \tag{3}$$

where w_{ji} is the connection weight of the i-th hidden node and j output node ($j = 1, \ldots,$ s). The training procedure of RBF networks aims to obtain a set of values for the kernel parameters $(\boldsymbol{v}_i, \sigma_i)$ and the connecting weights w_{ji}. Usually, the connecting weights are estimated through a regression-based analysis.

3 Particle Swarm Optimization

PSO has been widely used on numerical optimization tasks, in particular on strongly nonlinear problems [5, 19]. The basic design element is the particle, which is a real valued vector $\boldsymbol{p} \in \Re^r$ that represents a possible, yet complete, solution of the problem at hand. The set of particles forms the swarm. In this chapter the swarm size is denoted as M. Every particle $\boldsymbol{p}_i$ $(1 \leq i \leq M)$ is assigned a velocity vector $\boldsymbol{v}_i \in \Re^r$. In the t-th iteration, each particle informs others that make up the respective group of informants $Q_i(t)$. As $\boldsymbol{p}_i^{\text{best}}$ we symbolize the position of the lowest value of the fitness function obtained so far by the particle $\boldsymbol{p}_i$. The position associated with the lowest value of fitness function obtained so far by all particles belonging to $Q_i(t)$ is denoted as $\boldsymbol{p}_j^{\text{best}}(t)$. Then, the velocity $\boldsymbol{v}_i$ is calculated as:

$$\begin{aligned} \boldsymbol{v}_i\left(t+1\right) = \vartheta \boldsymbol{v}_i(t) + f_1 V\left(0,1\right) \circ \left(\boldsymbol{p}_i^{\text{best}}(t) - \boldsymbol{p}_i(t)\right) \\ + f_2 V\left(0,1\right) \circ \left(\boldsymbol{p}_j^{\text{best}}(t) - \boldsymbol{p}_i(t)\right) \end{aligned} \tag{4}$$

and the position of each particle is updated according to the next learning rule,

$$\boldsymbol{p}_i\,(t+1) = \boldsymbol{p}_i(t) + \boldsymbol{v}_i\,(t+1) \tag{5}$$

In the above equations, $\circ$ stands for the point-wise vector multiplication, $V(0,1)$ is a function that returns an r-th dimensional vector whose coordinates are numbers randomly generated by a uniform distribution in $[0, 1]$, f_1 and f_2 are constant positive numbers called the cognitive and social parameters and ϑ is also a positive constant called the inertia factor. Point-wise (or element-wise) vector multiplication is the element-by-element multiplication of two vectors. In accordance with the existing literature, typical values for these parameters are [26]: $\vartheta \in [0, 1]$, while f_1 and f_2 should be around 1.5. In this paper, for all the experiments the value 2 was chosen for the parameters f_1 and f_2. For the inertia factor ϑ, a pseudo-random selection in (0.5,1) took place for all the simulations.

4 RBF Network Training Algorithm

In this section we analytically describe the proposed algorithm, the flow sheet of which is illustrated in Fig. 2.

In view of this figure, three steps can be distinguished. In the first step we define an ordinary fuzzy partition in the input space and pre-process the data in terms of a suitable unsupervised learning process to extract a number of fuzzy subspaces. The second elaborates the resulting fuzzy subspaces by means of a weighted version

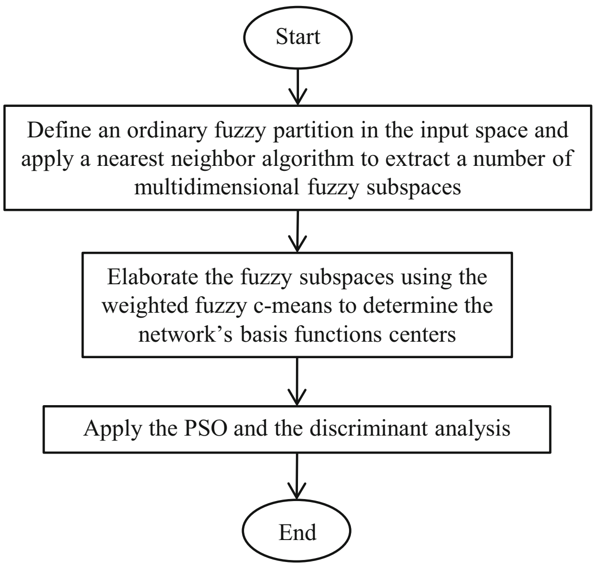


Fig. 2 The sequential steps of the proposed algorithm

of the fuzzy c-means model. The outcome of this clustering process is the basis function centers. Finally the last step involves the PSO and the discriminant analysis in order to infer the output classes. The detailed presentation of the above steps is provided by the next subsections.

4.1 Extraction of the Multidimensional Fuzzy Subspaces

Let us partition each universe of discourse X_j $(1 \leq j \leq p)$ into q_j symmetric triangular fuzzy sets $A_j^1, A_j^2, \ldots, A_j^{q_j}$ with membership functions of the form

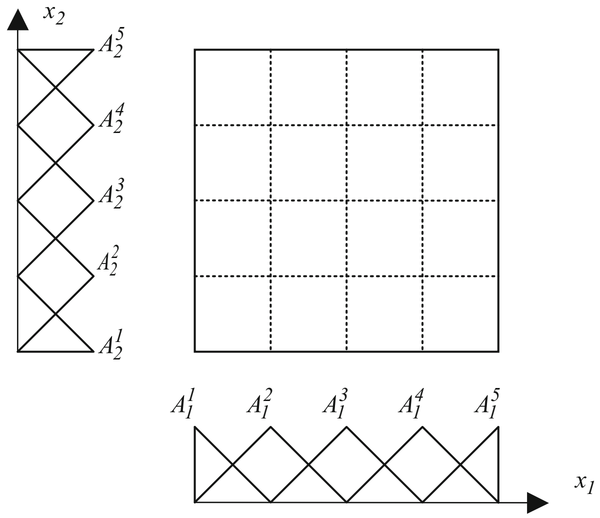
$$A(x) = \begin{cases} 1 - \frac{|x-\alpha|}{\delta\alpha}, & \text{if } x \in [\alpha - \delta\alpha, \alpha + \delta\alpha] \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where α is the center element and $\delta\alpha$ is the width. Each fuzzy set A_j^l is described as $A_j^l = \left\{a_j^l, \delta_j^l\right\}$. This procedure will create $L = \prod_{j=1}^{p} q_j$ multidimensional fuzzy sets in the feature space X. The set $\boldsymbol{V}_\ell$ $(1 \leq \ell \leq L)$ is composed by p one-dimensional fuzzy sets, $\boldsymbol{V}_\ell = \left\{A_1^\ell, A_2^\ell, \ldots, A_p^\ell\right\} = \left\{\boldsymbol{a}^\ell, \delta\boldsymbol{a}^\ell\right\}$, with $\boldsymbol{a}^\ell = \left[a_1^\ell, a_2^\ell, \ldots, a_p^\ell\right]$ and $\delta\boldsymbol{a}^\ell = \left[\delta a_1^\ell, \delta a_2^\ell, \ldots, \delta a_p^\ell\right]$. Figure 3 depicts a fuzzy partition in the two-dimensional space.

The matching degree between the training vector $\boldsymbol{x}_k$ and the ℓth multidimensional fuzzy set is [27]:

$$V_\ell\left(\boldsymbol{x}_k\right) = 1 - rd^\ell\left(\boldsymbol{x}_k\right) \tag{7}$$

Fig. 3 Fuzzy partition in a two-dimensional input space



where $rd^{\ell}(\boldsymbol{x}_k)$ is the relative Euclidean distance [27]:

$$rd^{\ell}\left(\boldsymbol{x}_k\right)=\begin{cases}\dfrac{\left[\sum_{j=1}^{p}\left(\alpha_j^{\ell}-x_{kj}\right)^2\right]^{1/2}}{\left[\sum_{j=1}^{p}\left(\delta\alpha_j^{\ell}\right)^2\right]^{1/2}} & \text{if } \left[\sum_{j=1}^{p}\left(\alpha_j^{\ell}-x_{kj}\right)^2\right]^{1/2}\leq\left[\sum_{j=1}^{p}\left(\delta\alpha_j^{\ell}\right)^2\right]^{1/2}\\ 1 & \text{otherwise}\end{cases} \tag{8}$$

The application of Eqs. (7) and (8) produces a spherical fuzzy subspace $\boldsymbol{C}^{\ell}=\left\{\boldsymbol{a}^{\ell},\ \rho^{\ell}\right\}$ centered at $\boldsymbol{a}^{\ell}=\left[a_1^{\ell},a_2^{\ell},\ldots,a_p^{\ell}\right]$ with radius $\rho^{\ell}=\sqrt{\sum_{j=1}^{p}\left(\delta\alpha_j^{\ell}\right)^2}$. As it was shown in [28], we can select the fuzzy subspaces that better describe the training data set according to the following algorithm:

Algorithm 1

Step 1) Set $k=1$ and $n=1$. Using (7)–(8) generate the first fuzzy subspace $\boldsymbol{C}^n=\{\boldsymbol{a}^n,\rho^n\}$.
Step 2) For $k=2$ to N do
    $rd_{\min}\left(\boldsymbol{x}_k\right)=\min_{1\leq i\leq n}\left\{rd^i\left(\boldsymbol{x}_k\right)\right\}$
    If $rd_{\min}\left(\boldsymbol{x}_k\right)<1$ then
        Assign the $\boldsymbol{x}_k$ to the fuzzy subspace that correspond to the $rd_{\min}(\boldsymbol{x}_k)$.
    else
    Set $n=n+1$ and using Eqs. (7)–(8) generate the nth fuzzy subspace $\boldsymbol{C}^n=\{\boldsymbol{a}^n,\ \rho^n\}$.
    endif
  endfor

The above algorithm dismembers the input space into n fuzzy subspaces denoted as $\boldsymbol{C}^l$ with $l=1,2,\ldots,n$. The fuzzy cardinality of the lth fuzzy subspace is:

$$\aleph\left(\boldsymbol{C}^l\right)=\sum_{k=1}^{N}V_l\left(\boldsymbol{x}_k\right)\quad(1\leq l\leq n) \tag{9}$$

To this end we define the relative fuzzy cardinality of the $\boldsymbol{C}^l$ as:

$$r_l=\frac{\aleph\left(\boldsymbol{C}^l\right)}{\sum_{j=1}^{n}\aleph\left(\boldsymbol{C}^j\right)}\quad(1\leq l\leq n) \tag{10}$$

Note that the radii $\rho^{\ell}\ (1\leq\ell\leq L)$ were used to extract the corresponding cardinalities. Therefore, they will not be used any more. Instead, each fuzzy subspace will be represented by its center element $\boldsymbol{a}_l$ and the respective relative cardinality r_l. Since $r_l \in [0,\ 1]$ it can be viewed as the weight of significance of the subspace $\boldsymbol{C}_l$.

4.2 Estimation of the Network's Basis Function Parameters

The standard methodology to determine the basis functions centers of the RBF network is the utilization of cluster analysis. In this section, we attempt to cluster the fuzzy subspaces resulting from the previous step. In general, the arsenal to cluster fuzzy data consists of a wide range of algorithmic tools [6, 29, 30]. In this section we employ the weighted fuzzy c-means method that was developed in [30, 31]. To implement this clustering scheme, we recall that each fuzzy subspace can be represented by its center element and the weight of significance. Therefore, the pairs $\{(\boldsymbol{a}_l,\ r_l) :\ 1 \le l \le n\}$ constitute a new data set, which we intend to cluster by using the aforementioned cluster analysis in order to obtain c fuzzy clusters in the input space with centers $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_c$. The objective function has the following form [30]:

$$J = \sum_{l=1}^{n} \sum_{i=1}^{c} r_l\, u_{il}^m\, \|\boldsymbol{a}_l - \boldsymbol{v}_i\|^2 \tag{11}$$

where u_{il} is the membership degree and m the fuzziness parameter.

The task is to minimize J under the constraint:

$$\sum_{i=1}^{c} u_{il} = 1 \quad \forall l \tag{12}$$

The membership degrees and cluster centers that solve the above optimization problem are provided by the following equations:

$$u_{il} = \frac{1}{\sum_{j=1}^{c} \left(\frac{\|\boldsymbol{a}_l - \boldsymbol{v}_i\|}{\|\boldsymbol{a}_l - \boldsymbol{v}_j\|} \right)^{2/(m-1)}} \tag{13}$$

$$\boldsymbol{v}_i = \frac{\sum_{l=1}^{n} r_l u_{ik}^m \boldsymbol{a}_l}{\sum_{l=1}^{n} r_l\, u_{il}^m}, \quad (1 \le l \le n) \tag{14}$$

The weighted fuzzy c-means is carried out by iteratively applying the Eqs. (13) and (14) until convergence. The interesting reader in referred to [30, 31] for a more detailed presentation of the clustering process. As mentioned, the results of this clustering approach is a set of cluster centers $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_c\}$, which coincide with the network's RBFs center elements. In order to calculate the kernel width of the i-th RBF, we use the approach developed by Niros and Tsekouras [32]. Specifically, the widths of the network's basis functions are as follows:

$$\sigma_i = \frac{2 * d_{\max}^i}{3} \quad (1 \le i \le c) \tag{15}$$

The distance function $d_{\max}^i$ is:

$$d_{\max}^i = \left\{ \max_{1 \le l \le n} \left\{ || \boldsymbol{x}_l - \boldsymbol{v}_i ||^2 \right\} : \ \boldsymbol{x}_l \in C_i \text{ such that } u_{il} \ge \tau \right\} \tag{16}$$

where C_i is the i-th fuzzy cluster, u_{il} is the membership degree of the l-th training data vector to the i-th final fuzzy cluster, and τ a small positive number such that $\tau \in (0,\ 1)$.

4.3 Discriminant Analysis and PSO Implementation

Given that the number of classes is denoted as s, the outputs of the network when the input is the vector $\boldsymbol{x}_k$ $(1 \le k \le N)$ are denoted as $y_1(\boldsymbol{x}_k), y_2(\boldsymbol{x}_k), \ldots, y_s(\boldsymbol{x}_k)$. By defining the vectors:

$$\boldsymbol{w}_j = \begin{bmatrix} w_{j1} \ w_{j2} \cdots w_{jc} \end{bmatrix}^T \quad (1 \le j \le s) \tag{17}$$

and

$$\boldsymbol{Hx}_k = \begin{bmatrix} h_1 (\boldsymbol{x}_k) \ h_2 (\boldsymbol{x}_k) \cdots h_c (\boldsymbol{x}_k) \end{bmatrix}^T \quad (1 \le j \le s)$$

the classifier assigns the vector $\boldsymbol{x}_k$ to the class j_k when the subsequent condition is fulfilled,

$$f_{j_k} (\boldsymbol{x}_k) = \max_{1 \le \ell \le s} \{ f_l (\boldsymbol{x}_k) \} \tag{18}$$

where f_j is the discriminant function, which is defined in terms of the inner product operation,

$$f_j (\boldsymbol{x}_k) = \boldsymbol{w}_j \cdot \boldsymbol{Hx}_k \tag{19}$$

According to the above analysis, the classifier is viewed as a network that computes s discriminant functions and for the current input selects the class that appears to have the largest value of all the rest of the discriminant functions. Note that the network's

w_{j1}	w_{j2}	$\cdots$	w_{jc}

Fig. 4 The structure of the particle in the PSO algorithm for the jth class

Table 1 Parameter setting

$\lambda = 0.71$	$f_2 = 2$
$q_j = 21$	Swarm size $M = 20$
$\tau = 0.001$	Number of intervals $= 5$
$f_1 = 2$	$\theta = 0.7$

Table 2 Machine learning data sets used

Data set	Number of classes	Number of inputs	Number of patterns (data)
WDBC	2	30	569
Wine	3	13	178
Pima	2	8	768

output will typically give a positive weight to the RBF neurons that belong to its category, and a negative weight to the others.

The estimation of the connection weights is carried out by the PSO algorithm. Due to the appearance of s outputs, the PSO is implemented separately with respect to each output class. For the jth class the structure of the particle is depicted in Fig. 4.

5 Evaluation Experiments

The experiments are conducted by using a split of the available data set into 60–40 % training and testing subsets, namely 60 % of the whole patterns are selected randomly for training and the remaining patterns are used for testing purposes.

We used a computer with a dual core CPU (i3-3120M) at 2.50 GHz with 4GB Ram Memory and the MATLAB software. Table 1 reports the parameter values that remain constant.

We consider several data sets concerning classification problems taken from the Machine Learning UCI repository provided by the website: http://archive.ics.uci.edu/ml/datasets.html.

Table 2 summarizes the pertinent details of the data set such as the number of features and number of patterns.

Table 3 Classification rates (%) for the WDBC data set

Number of hidden nodes	Training data set	Testing data set
2	76.83 ± 0.01	76.32 ± 0.01
4	86.22 ± 0.01	85.96 ± 0.01
5	89.74 ± 0.29	89.91 ± 0.35
6	88.27 ± 0.01	89.92 ± 0.01

Table 4 Comparison of the average performances for the WDBC data set

Classifier model	Classification rate (%)
Bayes Net [33]	95.81
DigaNN [34]	97.9
FSM [35]	98.3
MLP [36]	85.92
MPANN [37]	98.1
RBF2 [38]	97.13
RVM [39]	97.2
SVM [40]	96.68
Proposed RBF (12 hidden nodes)	97.92 ± 1.11

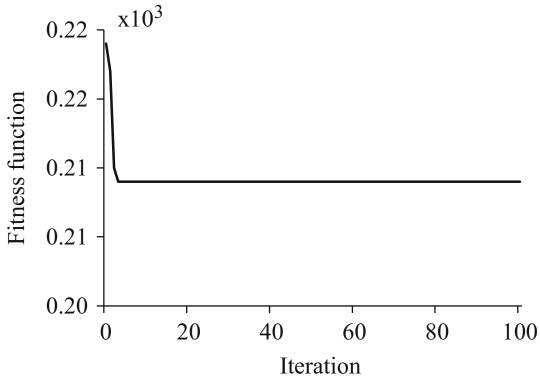


Fig. 5 Mean values of the fitness function obtained by the PSO for the WDBC data set ($c = 12$)

5.1 WDBC Data Set

In this experiment, we are concerned with data of high dimensionality (see Table 2). Initially, we evenly partitioned each input universe of discourse into 21 fuzzy sets of the form (6). By applying Algorithm 1 we obtained $n = 31$ fuzzy subspaces.

Table 3 displays the results when the number of hidden nodes is equal to 2, 4, 5, and 6. Table 4 provides a comparison study where the average classification rates obtained for several models are shown. According to this table, the proposed classifier is quantified as 97.92 reported for 12 hidden nodes.

Finally, Fig. 5 describes the convergence capabilities of the PSO when the number of hidden node is 1, which seems to be quite smooth.

Table 5 Classification rates (%) for the wine data set

Number of hidden nodes	Training data set	Testing data set
2	61.68 ± 0.01	74.65 ± 0.01
4	62.14 ± 0.01	74.83 ± 0.01
6	68.41 ± 2.34	74.04 ± 3.00
9	69.82 ± 1.68	80.04 ± 2.15
12	78.12 ± 2.34	90.12 ± 2.53

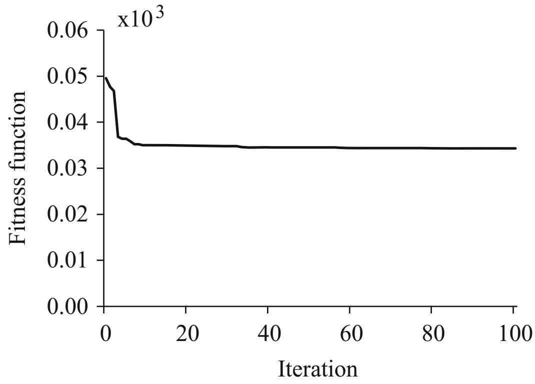


Fig. 6 Mean values of the fitness function J obtained by the PSO for the example 2 ($c = 12$)

5.2 *Wine Data Set*

Initially, we evenly partitioned each input universe of discourse into 21 fuzzy sets of the form (6). By applying Algorithm 1 we obtained $n = 22$ initial fuzzy subspaces.

Table 5 shows the final results for various number of hidden nodes raining and testing data. Note that the best performance comes in the case of 12 hidden nodes with a 90.12 % classification rate in the testing data case. Figure 6 describes the convergence capabilities of the PSO when the number of hidden node is 12.

5.3 *Pima Indians Diabetes Data Set*

In this example, each universe of discourse was partitioned into 21 symmetric triangular fuzzy sets. The implementation of Algorithm 1 gave $n = 25$ multidimensional fuzzy subspaces.

Table 6 shows the result when the number of RBF hidden nodes is 4, 6, 8, 10, and 12. The best performance value is 75.97 and it is reported in case of 12 hidden nodes. Table 7 shows the average classification rate obtained for several models and the proposed RBF classifier shows the best performance. Figure 7 describes the convergence capabilities of the PSO when the number of hidden node is 12.

Table 6 Classification rates for the pima data set with 4, 6, 8, 10, and 12 hidden nodes

Number of hidden nodes	Training data set	Testing data set
4	57.57 ± 0.01	53.09 ± 0.01
6	57.82 ± 0.01	55.92 ± 0.01
8	62.08 ± 0.29	59.86 ± 0.35
10	70.15 ± 1.90	65.06 ± 1.13
12	70.98 ± 0.66	68.16 ± 0.58

Table 7 Comparison of the average performance of classifiers for the pima Indians diabetes data set

Classifier model	Classification rate (%)
MLP [41]	73.10
RBF1 [42]	76.48
RBF2 [42]	76.96
RVM [43]	74.83
SVM [40]	74.72
Proposed RBF (14 hidden nodes)	75.97 ± 1.03

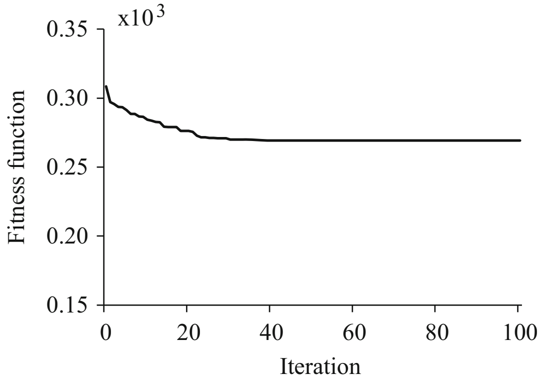


Fig. 7 Mean values of the fitness function J obtained by the PSO for the example 3 ($c = 12$)

6 Conclusion

In this chapter, we have discussed a new method for classification problems based on hierarchical fuzzy clustering and PSO. The proposed RBF classifier is a trainable mechanism consisting of three steps. The first step includes an ordinary fuzzy partition defined in the input space and a pre-processing unit which elaborates the data in terms of a suitable unsupervised learning process to extract a number of fuzzy subspaces. The second step clusters the fuzzy subspaces resulting from the previous step by means of a weighted version of the fuzzy *c*-means model. The outcome of this clustering process is the basis function centers. Finally in the last step, the PSO is implemented in order to infer the network's output. The method is applied to three machine learning data sets. As shown in the results of the experimental study, the proposed classifier exhibits a highly reliable performance and can be considered as an effective tool in two-class or multi-class pattern classification.

References

1. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall, Upper Saddle River (1999)
2. Schwenker, F., Kestler, H.A., Palm, G.: Three learning phases for radial-basis-function networks. Neural Netw. **14**, 439–458 (2001)
3. Xie, T., Yu, H., Hewlett, J., Rózycki, P., Wilamowski, B.: Fast and efficient second-order method for training radial basis function networks. IEEE Trans. Neural Netw. **23**(4), 609–619 (2012)
4. Chen, Y., Peng, L., Abraham, A.: Hierarchical radial basis function neural networks for classification problems. Advances in Neural Networks. Lecture Notes in Computer Science, vol. 3971, pp. 873–879 (2006)
5. Chen, Y., Yang, B., Zhou, J.: Automatic Design of Hierarchical RBF Networks for System Identification. PRICAI'06, Lecture Notes on Artificial Intelligence, **4099**, 1191–1195 (2006)
6. González, J., Rojas, I., Pomares, H., Ortega, J., Prieto, A.: A new clustering technique for function approximation. IEEE Trans. Neural Netw. **13**(1), 132–142 (2002)
7. Hou, M., Han, X.: Constructive approximation to multivariate function by decay RBF neural network. IEEE Trans. Neural Netw. **21**(9), 1517–1523 (2010)
8. Huan, H.X., Hien, D.T.T., Tue, H.H.: Efficient algorithm for training interpolation RBF networks with equally spaced nodes. IEEE Trans. Neural Netw. **22**(6), 982–988 (2011)
9. Babuska, R., Verbruggen, H.: Neuro-fuzzy methods for nonlinear system identification. Annu. Rev. Control **27**, 73–85 (2004)
10. Behloul, F., Lelieveldt, B.P.F., Boudraa, A., Reiber, J.H.C.: Optimal design of radial basis function neural networks for fuzzy-rule extraction in high dimensional data. Pattern Recognit. **35**, 659–675 (2002)
11. Haddadnia, J.H., Faez, K., Ahmadi, M.: A fuzzy hybrid algorithm for radial basis function neural network application in human face recognition. Pattern Recognit. **36**, 1187–1202 (2003)
12. Wong, Y.W., Seng, K.P., Ang, L.M.: Radial basis function neural network with incremental learning for face recognition. IEEE Trans. Syst. Man. Cybern. B Cybern. **41**(4), 940–949 (2011)
13. Valdez, F., Melin, P., Castillo, O.: Evolutionary method combining particle swarm optimisation and genetic algorithms using fuzzy logic for parameter adaptation and aggregation: the case neural network optimisation for face recognition. Int. J. Artif. Intell. Soft Comput. **2**(1–2), 77–102 (2010)
14. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, 2nd edn. Elsevier, San Diego (2003)
15. Bishop, C.M.: Neural Networks for Pattern Recognition. Clarendon, Oxford (1995)
16. Staiano, A., Tagliaferri, R., Pedrycz, W.: Improving RBF networks performance in regression tasks by means of a supervised fuzzy clustering. Neurocomputing **69**, 1570–1581 (2006)
17. Park, H.S., Pedrycz, W., Oh, S.K.: Granular neural networks and their development through context-based clustering and adjustable dimensionality of receptive fields. IEEE Trans. Neural Netw. **20**(10), 1604–1616 (2009)
18. Fathi, V., Montazer Gh, A.: An improvement in RBF learning algorithm based on PSO for real time applications. Neurocomputing **111**, 169–176 (2013)
19. Carlisle, A., Dozier, G.: Adapting particle swarm optimization to dynamic environments. In: Proceedings of International Conference on Artificial Intelligence, pp. 429–434 (2000)
20. Clerc, M.: Particle Swarm Optimization. ISTE, London (2006)
21. Clerc, M., Kennedy, J.: The particle swarm – explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. Evol. Comput. **6**(1), 58–73 (2002)
22. Oh, S.K., Kima, W.D., Pedrycz, W., Park, B.J.: Polynomial-based radial basis function neural networks (P-RBF NNs) realized with the aid of particle swarm optimization. Fuzzy Sets Syst. **163**, 54–77 (2011) (Elsevier)
23. Tsekouras, G.E., Tsimikas, J.: On training RBF neural networks using input-output fuzzy clustering and particle swarm optimization. Fuzzy Sets Syst. **221**, 65–89 (2013)

24. Qasem, S.N., Shamsuddin, S.M.: Improving performance of radial basis function network with particle swarm optimization. IEEE International Conference on Systems, Man and Cybernetics, pp. 534–540 (2009)
25. Qasem, S.N., Shamsuddin, S.M.: Radial basis function network based on time variant multi-objective particle swarm optimization for medical diseases diagnosis. Appl. Soft Comput. **11**, 1427–1438 (2011)
26. Chen, S., Hong, X., Harris, C.J.: Particle swarm optimization aided orthogonal forward regression for unified data modeling. IEEE Trans. Evol. Comput. **14**(4), 477–499 (2010)
27. Linkens, D.A., Chen, M.Y.: Input selection and partition validation for fuzzy modelling using neural network. Fuzzy Sets Syst. **107**, 299–308 (1999)
28. Tsekouras, G.E., Sarimveis, H., Bafas, G.: A simple algorithm for training fuzzy systems using input–output data. Adv. Eng. Softw. **34**(5), 247–259 (2003)
29. Looney, G.G.: Fuzzy connectivity clustering with radial basis kernel functions. Fuzzy Sets Syst. **160**, 1868–1885 (2009)
30. Tsekouras, G.E.: On the use of the weighted fuzzy c-means in fuzzy modeling. Adv. Eng. Softw. **36**, 287–300 (2005)
31. Tsekouras, G.E., Sarimveis, H., Kavakli, E., Bafas, G.: A hierarchical fuzzy-clustering approach to fuzzy modeling. Fuzzy Sets Syst. **150**(2), 245–266 (2005)
32. Niros, A.D., Tsekouras, G.E.: On training radial basis function neural networks using optimal fuzzy clustering. In: 17th IEEE Mediterranean Conference on Control and Automation, pp. 395–400 (2009)
33. Jensen, F.: An Introduction to Bayesian Networks. UCL Press/Springer, Berlin, Germany (1996)
34. Anagnostopoulos, I., Maglogiannis, I.: Neural network-based diagnostic and prognostic estimations in breast cancer micro scopic instances. Med. Biol. Eng. Comput. **44**(9), 773–784 (2006)
35. Adamczak, R., Duch, W., Jankowski, N.: New developments in the feature space mapping model. In: Third Conference on Neural Networks and Their Applications, Kule, October 1997, pp. 65–70 (1997)
36. Duda, R.O., Hart, P.E.: Pattern Classification and Scene Analysis. Wiley, New York (2002)
37. Abbass, H.A.: An evolutionary artificial neural networks approach for breast cancer diagnosis. Artif. Intell. Med. **25**(3), 265–281 (2002)
38. Young, Z.R.: A novel radial basis function neural network for discriminant analysis. IEEE Trans. Neural Netw. **17**(3), 604–612 (2006)
39. Tipping, M.E.: The relevance vector machine. Adv. Neural Inf. Process. Syst. **12**, 652–658 (2000)
40. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
41. Aeberhard, S., Coomans, D., de Vel, O.: Comparison of classifiers in high dimensional settings, Department of Computer Science and Department of Mathematics and Statistics, James Cook. University of North Queensland, Technical Report 92-02 (1992)
42. Kotsiantis, S.B., Pintelas, P.E.: Logitboost of simple Bayesian classifier. Informatica **29**, 53–59 (2005)
43. Lim, T., Loh, W., Shih, Y.: A comparison of prediction accuracy, complexity, and training time of thirty three old and new classification algorithms. Mach. Learn. **40**, 203–228 (2000)