CoEvRBFN: An Approach to Solving the Classification Problem with a Hybrid Cooperative-Coevolutive Algorithm

M. Dolores Pérez-Godoy[1], Antonio J. Rivera[1], M. José del Jesus[1], and Ignacio Rojas[2]

[1] Dept. of Computer Science,
University of Jaén, Jaén, Spain
{lperez,arivera,mjjesus}@ujaen.es
[2] Dept. of Computers Technology and Arquitecture
University of Granada, Granada, Spain
irojas@atc.ugr.es

Abstract. This paper presents a new cooperative-coevolutive algorithm for the design of Radial Basis Function Networks (RBFNs) for classification problems. The algorithm promotes a coevolutive environment where each individual represents a radial basis function (RBF) and the entire population is responsible for the final solution. As credit assignment three quality factors are considered which measure the role of the RBFs in the whole RBFN. In order to calculate the application probability of the coevolutive operators a Fuzzy Rule Base System has been used. The algorithm evaluation with different datasets has shown promising results.

Keywords: Radial Basis Function Network, Cooperative-Coevolution, Classification, Fuzzy Rule Base System.

1 Introduction

Nowadays, Radial Basis Function Networks (RBFNs) are one of the most important Artificial Neural Network paradigms in the machine learning field and have been used successfully in many areas such as pattern classification [4], function approximation [7] and time series prediction [22], among others. RBFs were initially used for numerical interpolation and function approximation [17], but the first research on neural networks based on RBFs [10][3] was carried out at the end of the eighties. RBFNs have interesting characteristics, such as a simple topological structure, universal approximation ability [14] and a local response which depends on the center and the width (radius) of the RBF.

The goal of RBFN learning is to determine the centers, widths and the linear output weights connecting the RBFs to the output neuron layer. The most traditional learning procedure has two stages: first, the centers and widths are determined and finally the output weights are established. Clustering techniques [15] are normally used to adjust the centers. The widths may be set with the same value, or may reflect the width of the clusters/RBFs previously calculated, or the average distance between RBFs, etc.

F. Sandoval et al. (Eds.): IWANN 2007, LNCS 4507, pp. 324–332, 2007.

In order to obtain the weights in the second stage, algorithms such as SVD [6] or gradient-based [23] can be used.

Another important paradigm for the RBFN design is evolutionary computation [1]. In most of the proposals within this evolutionary paradigm [4] an individual represents a whole RBFN, and different operators are applied to the entire population to improve individual fitness. Nevertheless evolutionary computation for this learning problem has some difficulties, especially the evaluation of independent sub-components (RBFs) [16]. Cooperative Coevolution [16] extends the basic computational model of evolution to provide a framework within which the individuals in the population represent only a part of the solution and evolve in parallel, not only competing to survive but also cooperating in order to find a common solution at the same time.

The authors have developed a hybrid proposal for RBFN design [18] which includes techniques like cooperative-coevolution, fuzzy rule base systems and traditional minimization algorithms, applied to function approximation and time series prediction. In this paper we present an important adaptation of our hybrid model for solving classification problems. For this objective it is necessary to adapt the structure of the network and its training algorithms as well as the method for calculating the credit assignment for an individual.

The organization of this paper is as follows: Section II introduces RBFNs and their optimization. In Section III our coevolutionary proposal for the design of the RBFNs is shown. The experimentation carried out is described in Section IV and finally, in Section V conclusions and future work are described.

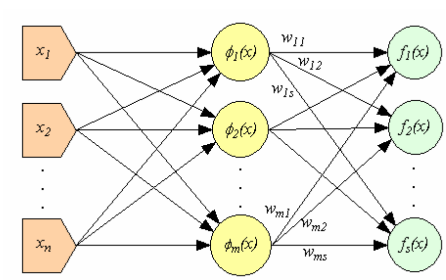

Fig. 1. Radial Basis Function Network

2 Classification with Radial Basis Function Networks

An RBFN is a feedforward neural network with three layers: an input layer with n nodes, a hidden layer with m neurons or RBFs, and an output layer with one or several nodes, see Figure 1. The m neurons of the hidden layer are activated by a radially-symmetric basis function, $\phi_i: R^n \rightarrow R,$ which can be defined in several ways. From all the possible choices for ϕ_i, the Gaussian function is the most widely used: $\phi_i(\vec{x}) = \phi_i(e^{-(\|\vec{x}-\vec{c}_i\|/d_i)^2})$, where $\vec{c}_i \in R^n$ is the center of basis function ϕ_i, $d_i \in R$ is the width (radius), and $\|\ \|$ is typically the Euclidean norm on R^n. This expression is the one used in this paper as Radial Basis Function (RBF). The output nodes implement equation 1.

$$f_j(\vec{x}) = \sum_{i=1}^{m} w_{ij}\phi_i(\vec{x}) \tag{1}$$

In a classification environment, the RBFN has to perform a mapping from an input space X^n to a finite set of classes Y with k classes. For this, a typical training set S is:

$$S = \{(\vec{x}_u, y_u) \mid x_u \in X^n,\ y \in Y,\quad u = 1,\ldots, p\} \tag{2}$$

where $\vec{x}_u$ is the feature vector and y_u is the class it belongs to. Usually, in the classification scenario, the number of outputs of the RBFN corresponds to the number of classes (k). To train the network, the class membership y_u is encoded into a binary vector $\vec{z}_u \in \{0,1\}^k$ through the relation $\vec{z}_u^{\,i}$ = 1 iff $y_u = i$, and $\vec{z}_u^{\,i}$ = 0 otherwise. The output class of the network will be the network output with maximum activation.

Different methods for the learning of RBFNs for classification problems have been set out in the specialized bibliography, and some of these use evolutionary algorithms (see [4][8][12] among others). Nevertheless, existing approaches represent a complete RBFN in an individual and typically suffer from the problems of a high runtime and a premature convergence in local minima. These problems can be overcome with the evolution of single RBFs in a cooperative-competitive scenario, as our proposal considers. In the specialized bibliography few cooperative coevolutionary procedures have been implemented up to now ([22][18][20]), due to difficulties in the credit assignment strategy which must promote competition among similar RBFs and cooperation among the different ones at the same time.

3 CoEvRBFN: A Coevolutive Hybrid Algorithm for RBFNs Design

A hybrid coevolutive approach for solve classification problems is proposed . In this approach each individual of the population represents a basis function and the entire population is responsible for the final solution. This allows for an environment where the individuals cooperate towards a definitive solution. However, they also compete for survival, since if their performance is poor they will be eliminated.

This scenario of coevolution reinforces the local operation (neurons with local response) and the interpretability of this kind of network and establishes an important design guideline in our algorithm. In order to measure the credit assignment of an individual, three factors have been proposed to evaluate the role of the RBF in the network. To decide the operators' application probability over a certain RBF the algorithm uses a Fuzzy Rule Based System (FRBS). The factors proposed for the credit assignment have been used as input parameters of the FRBS. In this proposal a new operator has been introduced and the expert knowledge has been adjusted.

The main steps of CoEvRBFN, explained in the following subsection, are shown in the pseudocode:

```
1. Initialize RBFN
2. Train RBFN
3. Evaluate RBFs
4. Apply operators to RBFs
5. Substitute the RBFs that were eliminated
6. Select the best RBFs
7. If the stop-condition is not verified go to step 2
```

3.1 RBFN Initialization

To define the initial network a simple process is used. The number of neurons specified (i.e. the size of population, m) is randomly allocated among the different classes of the training set.

Each RBF center , $\vec{c}_i$, is randomly established to a pattern of the training set, taking into account that the RBFs must be distributed equitably between the different classes. The widths, d_i, will be set to half of the average distance among the centers. Finally the weights, w_{ij} , are set to zero.

3.2 RBFN Training

During this stage weights, widths and centers of RBFs are trained. The proposed training exploits the local information that can be obtained from the local RBF behaviour. The technique used to calculate the weights is LMS [23]. In the present paper new algorithms to train centers and widths have been introduced.

A clustering-based technique for training centers has been used. In this way the RBF center, $\vec{c}_i$, is modified as follows:

$$c_{ij}' = c_{ij} \pm h \quad \forall j = 1 \ldots n \tag{3}$$

The increase or decrease of the old center is decided by means of a random number h $(0 \leq h \leq 0.1 \cdot d_i)$. The center is varied in order to approximate it to the average of the patterns which belong to the RBF class and inside its RBF width.

The objective of the width training is that the most of the patterns belonging to the RBF class will be inside the RBF width. In this way the RBF width is modified as follows:

$$\begin{cases} d' = d + h & \textit{if } (mdpco \leq 2 * d) \textit{ and } (npco > 0) \\ d' = d - h & \textit{if } npci * 0.1 \leq npnci \end{cases} \tag{4}$$

where h, is a random number $(0 \leq h \leq 0.1 \cdot d)$; *npco* and *npci*, are the number of patterns belonging to the RBF class respectively outside and inside the RBF width; *npnci* determines the number of patterns not belonging to the RBF class which are inside its width; and *mdpco* is the minimum distance between the RBF center and the patterns belonging to the RBF class outside its width.

3.3 RBF Evaluation

A credit assignment mechanism is required in order to evaluate the role of each base function in the coevolutive environment. For this purpose, three parameters, a_i, e_i, o_i are used for each RBF ϕ_i.

The contribution, a_i, of the RBF ϕ_i, $i=1\ldots m$, for the RBFN output is determined by considering the weight, w_i, and the number of patterns of the training set inside its width, pi_i, in order to penalize the RBF with a low weight and few patterns inside its width:

$$a_i = \begin{cases} |w_i| & \text{if } pi_i > q \\ |w_i| * (pi_i / q) & \text{otherwise} \end{cases} \tag{5}$$

where q is the average of the pi_i values minus twice standard deviation of the pi_i values.

The error measure, e_i, for each RBF ϕ_i, is obtained by counting the wrongly classified patterns inside its radius:

$$e_i = \frac{pibc_i}{pi_i} \tag{6}$$

where $pibc_i$ and pi_i are the number of wrongly classified patterns and the number of all patterns inside the RBF width respectively.

The overlapping of the RBF ϕ_i and the other RBFs is quantified by using the parameter o_i. This parameter is calculated by taking into account the *fitness sharing* [5] methodology, whose aim is to maintain the diversity in the population. The factor is expressed as:

$$o_i = \sum_{j=1}^{m} o_{ij} \qquad o_{ij} = \begin{cases} \left(1 - \|\phi_i - \phi_j\| / d_i\right) & \text{if } \|\phi_i - \phi_j\| < d_i \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where o_{ij} measures the overlapping of the RBF ϕ_i y ϕ_j, $j=1\ldots m$

3.4 Applying Operators to RBFs

In this paper three operators have been proposed to be applied to the RBFs. With respect to the previous work [18] the mutation operator has been changed and the new operator has been introduced.

- Operator REMOVE: is an operator which eliminates an RBF.
- Operator MUTATION: is an operator which modifies the width of the RBF, with a probability inversely proportional to the number of features of the classification problem (n), in a percentage between 10% and 20% of the old width. This operator also alters the center, modifying its coordinates in the same proportion as the width mutation. The number of coordinates to mutate is randomly obtained between 1% and 25% of the total number of features.
- Operator NULL: in this case no operator is applied to the RBF.

The operators will be applied to the whole population of RBFs. The probability for choosing an operator is determined by means of the Mamdani [9] fuzzy system, whose inputs are the parameters a_i, e_i and o_i. These determine the credit assignment to each RBF. These inputs are considered as linguistic variables va_i, ve_i and vo_i, and the outputs are p_{remove}, $p_{mutation}$ and p_{null}, which represent the probability of applying the operators REMOVE, MUTATION and NULL respectively.

Figure 2 shows the membership functions for the inputs and outputs linguistic labels respectively. The number of linguistic labels has been empirically determined, with centers and bases directly related to their meaning. Table 1 shows the rule base used to relate the described antecedents and consequents. The low number of rules allows a simpler fuzzy system to be designed.

To design the set of rules we take into account the fact that an RBF is worse if its contribution (a_i) is low, its error (e_i) is high and its overlapping (o_i) is also high. On the other hand an RBF is better when its contribution is high, its error is low and its overlapping is also low. Therefore, as the probability of eliminating a basis function increases, the associated RBF becomes worse. However, as the probability of not modifying an RBF increases, the associated basis function improves.

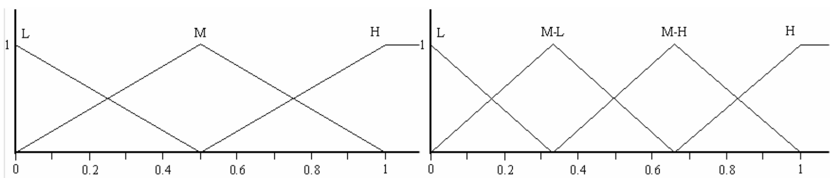

Fig. 2. Right: inputs variables membership functions for the FRBS. Left: output variables membership function.

Table 1. Rule base used in the FRBS

	Antecedents			Consequents		
	v_a	v_e	v_o	p_{remove}	$p_{mutation}$	p_{null}
R1	L			M-L	M-H	L
R2	M			L	H	M-L
R3	H			L	H	M-L
R4		L		L	H	M-H
R5		M		M-L	H	M-L
R6		H		M-H	M-H	L
R7			L	L	H	L
R8			M	M-L	H	M-L
R9			H	M-H	M-H	M-H

3.5 Introduction of New RBFs

In this step of the algorithm, the eliminated RBFs are substitute by new RBFs. A new technique of introduction of new RBFs has been developed in order to solve classification problems. The new RBFs will be located in the center of the largest zones wronly classified outside of any RBF width. The width of the new RBF will be set to the average of the RBFs present in the population.

3.6 Selection of the Best RBFs

After applying the mutation operator new RBFs appear. In this stage the new RBFs are compared with their parents in order to determine the RBFs with the best behaviour.

4 Experimental Results

The data sets used in this section were obtained from the UCI Repository of Machine Learning Database: Iris, Wine, Wbcd and Glass. The population size is the same as the number of classes in the benchmark used. The estimation of the generalization capacity for the RBFNs is obtained by means of the ten-fold cross-validation, and the number of generations is fixed at 200. Tables 2-5 show the results obtained with CoEvRBFN and with different RFBN learning algorithms described in the specialized bibliography. An analysis of the results shows that:

- CoEvRBFN obtains RBFNs with a simple structure (the number of RBFs equals the number of classes) and with results comparable to other methods. It implies that the final network is more interpretable, an important characteristic in classification problems.
- The generalization capacity is higher than the other methods in Iris, Wine and Wbcd problems, and a bit lower in the Glass problem, but with a very low number of RBFs.

Table 2. Results obtained with Iris dataset

Algorithm	*RBf nodes*	*Classification rate (%)*
Netlab[11]	4.5	96
Bing Yu[2]	6	97.33
Newrb[13]	9	79.37
Wallace[21]	3	98
Tian [19]	14.1	97.2
Topchy[20]	5	95.6
CoEvRBFN	3	98.3

Table 3. Results obtained with Wine dataset

Algorithm	*RBf nodes*	*Classification rate (%)*
Netlab[11]	3	98.9
Bing Yu[2]	20	96.3
Newrb[13]	58	92.8
Tian[19]	81.9	95.0
CoEvRBFN	3	98.9

Table 4. Results obtained with Wbcd dataset

Algorithm	*RBf nodes*	*Classification rate (%)*
Netlab[11]	2.2	97.1
Wallace[21]	2	96.6
CoEvRBFN	2	98.2

Table 5. Results obtained with Glass Dataset

Algorithm	*RBf nodes*	*Classification rate (%)*
Bing Yu[2]	27	86.2
Newrb[13]	87	78.5
CoEvRBFN	7	74.7

5 Conclusions

In this work a new hybrid coevolutive algorithm for the optimization of the parameters defining an RBFN for classification problems has been proposed. An important key point of the presented proposal is the identification of the role (credit assignment) of each basis function in the whole network. In order to evaluate this value of a given RBF, three factors are used: the RBF contribution to the network's output, a_i; the error in the basis function radius, e_i; and the degree of overlapping among RBFs, o_i. In order to drive the coevolutive process three operators are used: elimination, mutation or maintaining the given individual/RBF. The application of these is determined by a fuzzy rule-based system. The inputs of this system are the three parameters, a_i, e_i, and o_i, used for credit assignment. Finally the RBFN characteristic parameters, centers, widths and weights are trained with local methods among coevolutive generations.

The proposed approach has been evaluated using well-known benchmarks, and the results obtained are comparable with other more mature methods.

As future work a deeper study of the operators and training methods of the individuals will be carried out.

Acknowledgements. This work has been partially supported by the CICYT Spanish Project TIN2004-01419 and TIN2005-04386-C05-03.

References

[1] Bäck, T., Hammel, U., Schwefel, H.: Evolutionary computation: comments on the history and current state. IEEE Trans. on Evolutionary Computation 1(1), 3–17 (1997)

[2] Yu, B., He, X.: Training Radial Basis Function Networks with Differencial Evolution. In: IEEE International Conference on Granular Computing, pp. 369–372 (2006)

[3] Broomhead, D., Lowe, D.: Multivariable functional interpolation and adaptive networks. Complex System 2, 321–355 (1988)

[4] Buchtala, O., Klimek, M., Sick, B.: Evolutionary optimization of radial basis function classifiers for data mining applications. IEEE Transactions on Systems, Man. and Cybernetics, Part. B 35(5), 928–947 (2005)

[5] Goldberg, D., Richardson, J.: Genetic algorithms with sharing for multimodal function optimization. In: Grefenstette (ed.) Proceedings of the Second International Conference on Genetic Algorithms, pp. 41–49. Lawrence Erlbaum Associates, Mahwah (1987)

[6] Golub, G., Van Loan, C.: Matrix computations, 3rd edn. J. Hopkins University Press, Baltimore (1996)

[7] González, J., Rojas, I., Ortega, J., Pomares, H., Fernández, J., Fco, A.: Multiobjective evolutionary optimization of the size, shape, and position parameters of radial basis function networks for function approximation. IEEE Transactions on Neural Networks 14(6), 1478–1495 (2003)

[8] Lacerda, E., Carvalho, A., Braga, A., Ludermir, T.: Evolutionary Radial Functions for Credit Assessment. In: Applied Intelligence 22, pp. 167–181. Springer, Netherlands (2005)

[9] Mandani, E., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. Int. J. Man-Machine Stud. 7(1), 1–13 (1975)

[10] Moody, J., Darken, C.J.: Fast learning in networks of locally-tuned processing units. Neural Computing 1, 281–294 (1989)

[11] Nabvey, J., Darken, C.J.: Netlab Neural Network Software. http://www.ncrg.aston.ac.uk/netlab

[12] Neruda, R., Kudová, P.: Learning methods for radial basis function networks. Future Generation Computer Systems 21(7), 1131–1142 (2005)

[13] Newrb, Matlab neural networks toolbox

[14] Park, J., Sandberg, I.: Universal approximation using radial-basis function networks. Neural Comput 3, 246–257 (1991)

[15] Pedrycz, W.: Conditional fuzzy clustering in the design of radial basis function neural networks. IEEE Transactions on Neural Networks 9(4), 601–612 (1998)

[16] Potter, M., De Jong, K.: Cooperative Coevolution: an architecture for evolving coadapted subcomponents. Evolutionary Computation 8(1), 1–29 (2000)

[17] Powell, M.: Radial basis functions for multivariable interpolation: A review. In: IMA. Conf. on Algorithms for the approximation of functions and data, pp. 143-167 (1985)

[18] Rivera, A.J., Rojas, I., Ortega, J., del Jesús, M.J.: A new hybrid methodology for cooperative-coevolutionary optimization of radial basis function networks Soft Computing (2006) ISSN 1432-7643. D.O.I: http://dx.doi.org/10.1007/s00500-006-0128-9

[19] Tian, J., Li, M-Q., Chen, F-Z.: A three-phase rbfnn learning algorithm for complex classification. In: International Conf. on Machine Learning and Cybernetics, pp. 4134-4139 (2005)

[20] Topchy, A., Lebedko, O., Miagkikh, V., Kasabov, N.: Adaptive training of radial basis function networks based on co-operative evolution and evolutionary programming. In: Kasabov, N., et al. (eds.) Prog. in connectionist-based information syst. pp. 253–258. Springer, Heidelberg (1998)

[21] Wallace, M., Tsapatsoulis, N., Kollias, S.: Intelligent initialization of resource allocating RBF networks. Neural Networks 18(2), 117–122 (2005)

[22] Whitehead, B., Choate, T.: Cooperative-competitive genetic evolution of Radial Basis Function centers and widths for time series prediction. IEEE Trans. on Neural Networks 7(4), 869–880 (1996)

[23] Widrow, B., Lehr, M.A.: 30 Years of adaptive neural networks: perceptron, madaline and backpropagation. In: Proceedings of the IEEE 78(9), 1415–1442 (1990)