PARAMETERIZATION OF RBF NEURAL NETWORKS VIA COMBINATION OF UNSUPERVISED PROCEDURES AND A NEW WAY OF SCALING PARAMETERS

Carlos Alberto Murari Pinheiro[1], *Flavia Aparecida Oliveira Santos*[2], *Pedro Paulo Balestrassi*[3] *and Benedito Isaias Lima Lopes*[4]

[1,2] Institute of Systems Engineering and Information Technology, Federal University of Itajubá, Itajubá, Brazil
pinheiro@unifei.edu.br; flaviaoliveira@unifei.edu.br
[3] Institute of Production Engineering and Management, Federal University of Itajubá, Itajubá, Brazil
ppbalestrassi@gmail.com
[4] Institute of Electrical Systems and Energy, Federal University of Itajubá, Itajubá, Brazil
isaiasdlima@gmail.com

Phone: (55) 35 3629-1875
Mailbox: 50 - Zip code: 37500 903 - Itajubá - MG

ABSTRACT

This paper addresses a procedure for the selection of parameters in Radial Basis Function Neural Networks. The approach consists of the combination of unsupervised procedures and a new way to scale parameters associated to the values of widths of the radial functions. The application of the approach will be illustrated with examples of approximation of functions and time series forecasting.

KEYWORDS

Radial basis function networks, parameter adjustment, artificial neural networks, approximation of functions, time series forecasting.

1. INTRODUCTION

Radial Basis Function (RBF) neural networks are effectively utilized in time series forecasting [16, 8], approximation of functions [14], control systems [7], pattern classification [1], among other applications.

In its most basic form, RBF networks have only one hidden layer, while Multilayer Perceptron (MLP) neural networks present one or more hidden layers, according to its specifications [6].

Hidden layers in RBF networks have activation functions modeled by radial-basis functions, where the more usual are Gaussian functions defined by their centers and widths. The output layer has weights that aggregate the outputs of the radial functions, providing the output information of the network (which can have more than one output, according to the corresponding application).

There are some strategies used to adjust the parameters (centers, widths and weights) in RBF networks, typically classified as empirical, auto-organized and supervised. In empirical strategies, the centers of the functions are adjusted randomly, and the width values are given by the relationship between the positions of the centers [6]. The objective is to obtain uniform distributions for the radial functions in order to enable a good capacity of generalization of the network. The weights of the output layer are calculated by the least squares method based on the output values of the established Gaussian functions, with a set of input training data and the corresponding pattern for the network output in question. Auto-organized strategies [6] use data clustering algorithms (K-means, for example) to select the values of the centers of each radial basis function. The values of the widths are adjusted empirically, and the weights of the output layer are adjusted by the least squares method. The paper of [16] uses an optimized data partitioning method which is based on the distance between the centers of the data clustering. An additional cost function is associated with an optimization method with the objective of adjusting the centers and widths of the radial basis functions. Supervised strategies [6, 10, 15] employ methods such as gradient descent to adjust all parameters of a determined RBF network, where the error information between the information network output concerned and its desired pattern are used in the training process.

In reference [5] it was shown the algorithm Output Value-Based Initializer, which calculates the initial values for the centers and widths of the radial functions according to the output values of the target function. In paper [17] also suggested the use of supervised learning, where a new discrete-continuous algorithm is proposed for the construction of a RBF model. First, the orthogonal least squares is used for an initial model, and the Levenberg–Marquardt method is used to optimize the hidden nodes and output weights in the continuous space.

This paper addresses a procedure for the selection of parameters in RBF neural networks. The approach consists of the combination of unsupervised procedures and a new way to scale parameters associated to the values of widths of the radial functions. Applications in approximations of functions and time series forecasting will be exemplified.

This paper is organized as follows. Section 2 contains a basic review of the RBF networks and the description of typical unsupervised methods for setting parameters. Section 3 describes the methodology proposed in this paper. Application examples are illustrated in Section 4. And in Section 5 is the conclusion of the paper and suggestions for future work.

2. BASIC REVIEW

A typical structure of RBF networks is shown in Figure 1.

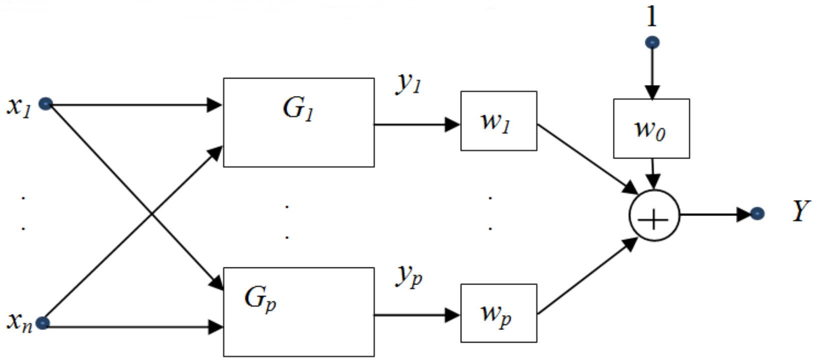


Fig. 1 *Typical RBF neural network structure.*

There are no weights in the first layer of the neural network. The input variables are the data of the activation functions (G_j), which are themselves nodes of the hidden layer of the network. The typical expression of a radial basis function is given by (1), where c_j denotes its central value of the function in question, and σ_j is the width of the RBF.

$$y_j = G_j(\mathbf{x}) = e^{-(\frac{(x_1-c_1)^2}{2\sigma_1^2}+\cdots+\frac{(x_n-c_n)^2}{2\sigma_n^2})} \qquad (1)$$

The weights (w_i) of the output layer of the network multiply the values (y_j) supplied by radial basis functions of the model, and the aggregation of these values results in output information Y of the network, which is added to a unitary bias value multiplied by the associated weight w_0. RBF neural networks can have other outputs associated with other corresponding weights. The information for each output of a given neural network is modeled by equation (2).

$$Y = \sum_{j=1}^{p} w_j \, G_j(\mathbf{x}) \qquad (2)$$

2.1. Typical Unsupervised Methods for Setting Parameters

This item addresses typical procedures unsupervised, often used for adjusting parameters of RBF networks.

2.1.1. Values of the Centers of the Gaussian Functions Randomly Selected

a) The central value of each radial basis function of a network is randomly chosen from the training data, and the justification of this procedure is explained in [9].

b) The value of width (standard deviation) of each radial basis function network is given by expression (3), where N_c is the number of functions used in the hidden layer and D_{max} is the maximum distance between selected pairs of values of the centers in question.

$$\sigma = \frac{D_{max}}{\sqrt{2N_c}} \qquad (3)$$

A common width value ($\sigma_{kj} = \sigma$) for all network functions can be used when the input data have a uniform distribution, unusual condition in most practical applications. A more feasible procedure is expressed by (4), where c_{ij} are values in the neighborhood of the corresponding c_{kj} center associated with the radial function in question, a suggested value for p is 2 [11].

$$\sigma_{kj} = \frac{1}{p}\sqrt{\sum_{m=1}^{p}\left\|c_{kj} - c_{ij}\right\|^2} \tag{4}$$

Another approach [13] uses the data related to the Euclidean distance between the center value of each RBF and its immediate vicinity, which information is multiplied by a constant K as expressed in (5) in order to establish an overlap appropriate between adjacent network functions in question, whose purpose is to improve the interpolation of the resulting model.

$$\sigma_{kj} = K.\min\left(\left\|c_{kj} - c_{ij}\right\|\right) \tag{5}$$

[2] suggested the use of a scaling factor K_e as expressed in (6) to jointly adjust the values of the deviations of RBF, whose values can be estimated by one of three expressions mentioned above, allowing a better modeling capability of the resulting network.

$$\sigma_{kj} = K_e.\sigma_{kj} \tag{6}$$

c) The values of the weights of the output layer are calculated using the pseudo-inverse method (7), where G^+ is the corresponding values associated with $G_j(\boldsymbol{x})$ of RBF functions and Y_d are the values of the output desired in the neural network. When there are no problems of matrix inversion, the conventional least squares method is used.

$$w_j = G^+.Y_d \tag{7}$$

2.1.2. Values of the Centers Determined by Methods of Data Clustering

a) The central value of each radial basis function is determined by clustering techniques, processing the network data and providing values of the centers of clusters in question. The K-means algorithm is usually employed in the context of RBF networks [12], and alternatively more elaborate methods can be applied, for example, via the self-organizing map or other techniques.

b) The values of the widths of the RBF can be adjusted by information of distances related to centers of the functions and data in the neighboring regions, similarly to the procedures mentioned above. Other approaches processing the values of standard deviations associated with the RBF information covariance matrices resulting from elliptical data clustering algorithms, such as the algorithm of Kessel-Gustafson [12], for example, but that tends to be computationally expensive due to the complexity of the algorithm involved.

c) The values of the weights of the hidden layer are calculated by the methods of least squares.

3. METHODOLOGY

This section details the approach suggested in this paper to adjust the parameters of RBF networks, compiled from combinations of unsupervised procedures and with and a new way to scale parameters.

3.1. Procedure for Setting Parameters in RBF Neural Networks

1- The number of nodes is chosen for the hidden layer of the RBF neural network in question;

2- Network training data are processed by a clustering algorithm Fuzzy-Cmeans [3] in order to optimize the clusters of the corresponding data. The information of the clusters will be used to select the parameters of the activation functions, and the number of clusters defines the number of nodes in the network hidden layer;

3- Centers (c_j) of the Gaussian functions are the core values of the clusters determined by the Fuzzy-Cmeans algorithm;

4- The widths of the functions are initially estimated with (8), calculating the standard deviation (σ_j*) on the range of data (x_j) of each variable associated with the cluster identified by the algorithm;

$$\sigma_j^* = \sqrt{\frac{\sum_{j=1}^{m} x_j^2 - \frac{1}{m}\left(\sum_{j=1}^{m} x_j\right)^2}{m-1}} \tag{8}$$

5- A new way to scale parameters associated to the initial values of widths of the radial functions and intervals of data clusters, is proposed via equation (9). The purpose is to establish appropriate values of widths to improve the characteristic of generalization of network. In the case of a null value resulting from the application of equation (8) or (9), is repeated the value of the deviation of the nearest cluster;

$$\sigma_j' = \frac{1}{2}\frac{\left|\max(\mathbf{x}_j) - \min(\mathbf{x}_j)\right|}{\sigma_j^*} \tag{9}$$

6- A common multiplicative factor K_m (10), with values between 1 and 10, serves as the additional adjustment of the parameters (widths) in question;

$$\sigma_j = K_m \sigma_j' \tag{10}$$

7- The weights $\mathbf{w}_j$ of the output layer are then calculated by the least squares method (11), where "^{T}" denotes transposed matrix and "$^{-1}$" inverse matrix, using the regressor $\mathbf{F_r}$, output values of Gaussian functions (y_1 through y_p), and the corresponding expected pattern for the output information ($\mathbf{Y}$) of the network;

$$\mathbf{F}_r = [\mathbf{y}_1\mathbf{y}_2\ \ldots\ \mathbf{y}_p\mathbf{1}];$$

$$\mathbf{w}_j = [\mathbf{F}_r^T\mathbf{F}_r]^{-1}\mathbf{F}_r^T\mathbf{Y}. \tag{11}$$

8- Verify the precision achieved by the neural network. If necessary, vary the value of the multiplying factor of step 6, or increase the number of network nodes in step 1, and repeat the procedure.

4. EXAMPLES OF APPLICATIONS

Initially some simple examples are shown in order to illustrate the procedure outlined in this paper, and afterwards more elaborate applications will be presented. Comparisons with results obtained with standard procedures for setting parameters will be provided.

4.1. Example 1 – Approximation of Function

This example is didactic, serving to illustrate the application of the approach and illustrate the calculations. The RBF network in question is applied to model the nonlinear function expressed by (12).A data set of twenty-one samples represents the original function: $\boldsymbol{x_1}$ = [0; 0.05; 0.1; 0.15; 0.2; 0.25; 0.3; 0.35; 0.4; 0.45; 0.5; 0.55; 0.6; 0.65; 0.7; 0.75; 0.8; 0.85; 0.9; 0.95; 1]; $\boldsymbol{y}$ = [0; 0.0025; 0.01; 0.0225; 0.04; 0.0625; 0.09; 0.1225; 0.16; 0.2025; 0.25; 0.3025; 0.36; 0.4225; 0.49; 0.5625; 0.64; 0.7225; 0.81; 0.9025; 1].

$$y = x_1^2; \qquad x_1 \in [0, 1]. \tag{12}$$

Three experiments were conducted and are described below: i) Initially it was considered the unsupervised procedure with the values of the centers of the radial functions randomly selected; ii) The same example was run using a clustering algorithm as shown in previous works; iii) The proposed algorithm was then considered.

4.1.1. Unsupervised procedure (with randomly centers)

It was assumed a network with three nodes in the input layer, and the values of the centers were randomly chosen from the vector data $\boldsymbol{x_1}$. Table 1 contains the values of the centers of Gaussians functions.

Using the expression (3) follows the common value of the widths: $\sigma = \sigma_{11} = \sigma_{12} = \sigma_{13} = 0.26$. With the expression (7) or (11), the values of the weights of the output layer of the network are calculated. The simulation of the network model and the comparison with the data of the original function resulted in the sum of the squared error (SSE = $\sum[\boldsymbol{y} - \mathbf{Y}]^2/2$) of $7.0*10^{-2}$.

Adjusting now the values of the widths individually by expression (5), the values are obtained for $K = 1$.Now the SSE was modified to $2.8*10^{-2}$, resulting in improved accuracy of the corresponding neural model.

Now adopting a scaling factor given by (5) or (6) for $K = K_e = 3$, whose data was tested to achieve better accuracy in modeling, follow the values of the widths of the functions of the network and weight values of the output layer. The SSE was modified to $7.7*10^{-5}$, resulting in better accuracy of the corresponding neural model.

c_{11}	0.2		
c_{12}	0.6		
c_{13}	0.85		
σ_{11}	0.26	0.4	1.2
σ_{12}	0.26	0.25	0.75
σ_{13}	0.26	0.25	0.75
w_0	0.3299	1.3901	7.1102
w_1	-0.2448	-1.408	-8.8043
w_2	-0.2815	0.0585	4.1214
w_3	0.6421	-0.3285	-2.6859
SSE	$7.0*10^{-2}$	$2.8*10^{-2}$	$7.7*10^{-5}$

Tab I *Values of centers, widths and weights of the radial functions (Case 1).*

4.1.2. Clustering algorithm

Figure 2 illustrates the clusters obtained via Fuzzy C-Means on data from the original system. The resulting clusters are represented by three colors, and their centers are indicated by circles in black. The values of the centers of the clusters are (0.1728; 0.0435), (0.5707; 0.3354) and (0.8945; 0.8063), respectively, for the vectors $\boldsymbol{x}_1$ and $\boldsymbol{y}$ of the nonlinear function. In Table 2there are the corresponding values of the centers of the clusters associated with the vector data $\boldsymbol{x}$. Using the procedure defined by equation (5) with $K = 3$, are also in Table 2 the values of the widths of the radial functions and the values of the weights of the output layer of the network. Now the SSE (related to the error estimates between the RBF and the original data of the system) was modified to $4.4*10^{-5}$, resulting in improved accuracy of the corresponding neural model compared to previous network.

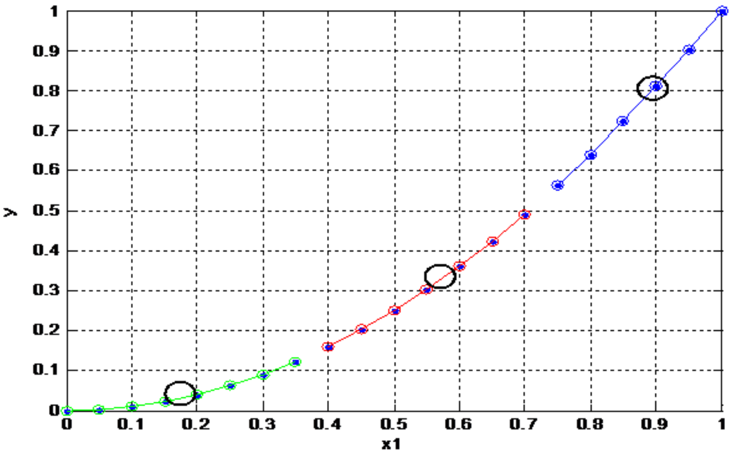
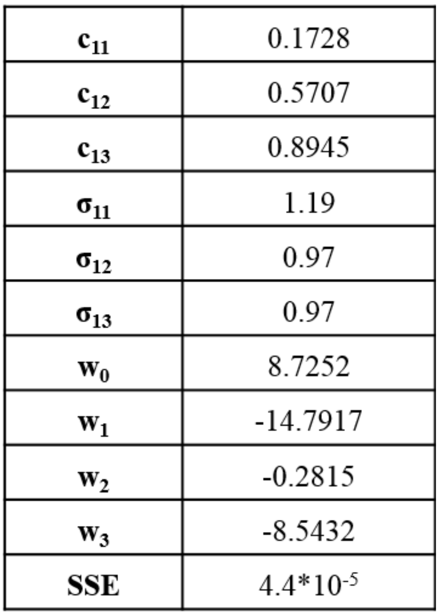


Fig. 2 *Data from equation (12) and the respective clusters.*



c_{11}	0.1728
c_{12}	0.5707
c_{13}	0.8945
σ_{11}	1.19
σ_{12}	0.97
σ_{13}	0.97
w_0	8.7252
w_1	-14.7917
w_2	-0.2815
w_3	-8.5432
SSE	$4.4*10^{-5}$

Tab II *Values of centers, widths and weights of the radial functions with K = 3 (Clustering algorithm).*

4.1.3. Proposed algorithm

For the same data clusters (Figure 2), derived from the application of Fuzzy-Cmeans algorithm, there are the intervals associated to the vector data $\boldsymbol{x_1}$:[0; 0,05; 0,1; 0,15; 0,2; 0,25; 0,3; 0,35], [0,4; 0,45; 0,5; 0,55; 0,6; 0,65; 0,7] and[0,75; 0,8; 0,85; 0,9; 0,95; 1].Calculating by (8) the variances associated with each cluster, we have $\sigma_{11}^{*} = 0.1225$, $\sigma_{12}^{*} = 0.1080$ and $\sigma_{13}^{*} = 0.935$.

Using (9), we have the values appropriately scaled: $\sigma_1{}^{'}$=0.5*(0.35− 0)/0.1225 = 1.4286; $\sigma_2{}^{'}$=0.5*(0.7−0.4)/0.108=1.3889; $\sigma_3{}^{'}$=0.5*(1−0.75)/0.0935=1.3369.For the same value of the scale factor $K = K_e = K_m = 3$ (10) used in the previous neural models. In Table 3 are the final values of the widths of the radial functions.

$\mathbf{c_{11}}$	0.1728
$\mathbf{c_{12}}$	0.5707
$\mathbf{c_{13}}$	0.8945
$\boldsymbol{\sigma_{11}}$	4.2857
$\boldsymbol{\sigma_{12}}$	4.1667
$\boldsymbol{\sigma_{13}}$	4.0107
$\mathbf{w_0}$	33.8119
$\mathbf{w_1}$	−369.0504
$\mathbf{w_2}$	672.7098
$\mathbf{w_3}$	−339.8387
SSE	$8.6*10^{-8}$

Tab III *Values of centers, widths and weights of the radial functions with K = 3 (Proposed algorithm).*

With the values $\mathbf{y}_1$, $\mathbf{y}_2$, and $\mathbf{y}_3$calculated for the three Gaussian functions corresponding to the vector $\boldsymbol{x}_1$ from the input data of the system in question, using (11) with the vector $\boldsymbol{y}$(data expected for the network output) the weights of the RBF are calculated by (11).

Figure3 shows the data of the original function (“.”) and the estimated data (“+”) for the neural network with the parameters determined with the approach suggested in this paper. The SSE was modified to $8.6*10^{-8}$, resulting in better accuracy of the corresponding neural model.

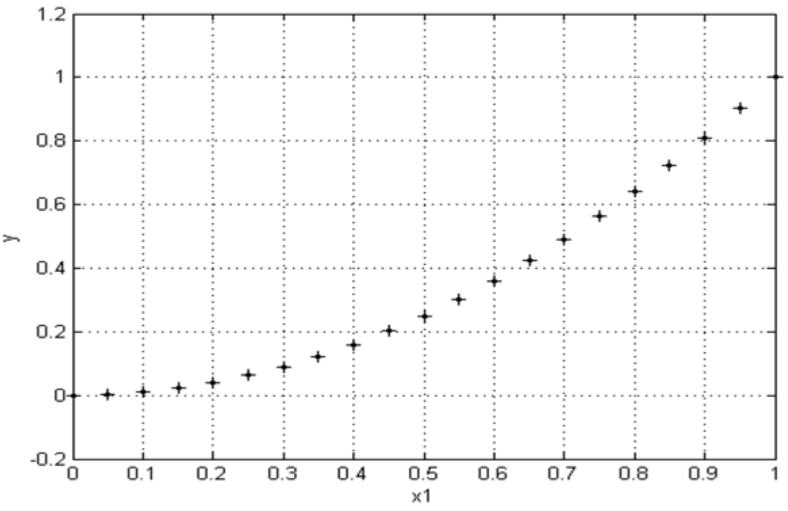


Fig. 3 *Data from equation (12) and the values estimated with the corresponding RBF.*

4.2. Example 2 – Application in Prediction of Time Series

Figure 4 contains data related to prices of a particular commodity, with records of prices for seventy-three months. A RBF neural network will be used as model for the corresponding time series. The network model was obtained with data of the first sixty-two months, and the data of the next eleven months was used to verify the generalizability of the model.

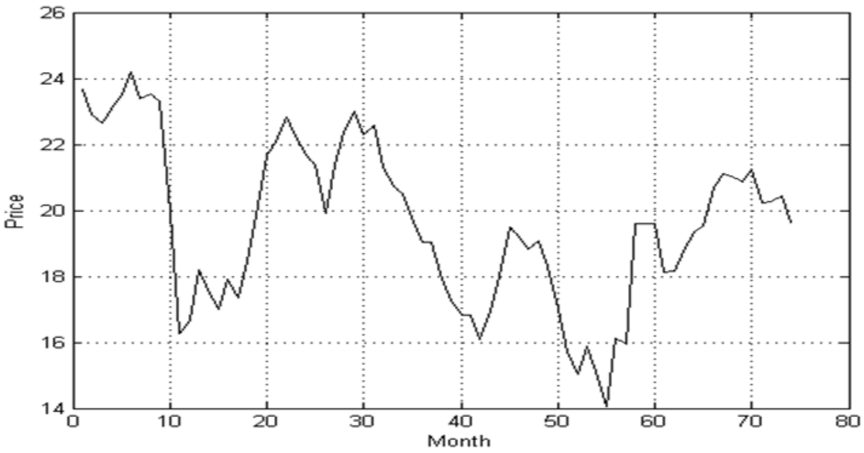


Fig. 4 *Time series data.*

A current price value (*k*) and the previous value (*k*-1) were used in order to estimate the price of the following month (*k*+1). The basic structure of RBF neural network will be similar to Figure 5, where the modeling variables are: x_1 = Previous Price; x_2 = Current Price;*y* = Following Month Price.

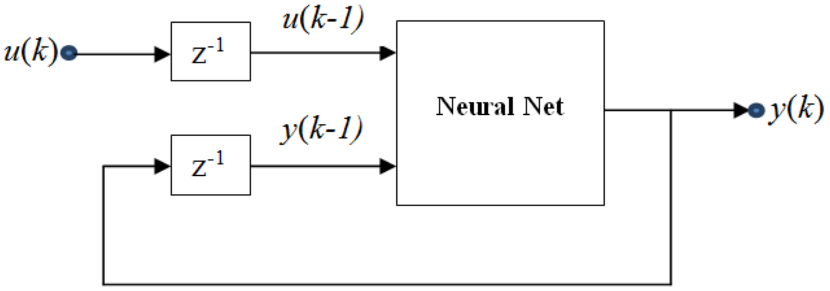


Fig. 5 *Example of neural network used for system modeling.*

The following experiments described below were considered.

4.2.1. Initially was considered a RBF network obtained via supervised training.

Using a neural network toolbox (MATLAB), a model (*newrb*) was generated with a good accuracy for the training data. But was noted that with a higher accuracy in the resulting model via training data, was obtained a larger error in the predictions with the test data. For comparison, the mean absolute percentage error (MAPE) was used between the values given for the data test and output data estimated by RBF:$\sum \left| (Y_{test} - Y_{estimated})/Y_{test} \right| /N$, where N is the number of samples. With the model obtained the value of error was: MAPE = 3.8%.

4.2.2. Now using the approach proposed in this paper.

For the hidden layer network six nodes were adopted. The six clusters of data processed by the Fuzzy-Cmeans algorithm are illustrated in Figure 6, where the values of the centers of the Gaussian are shown in Table 4.

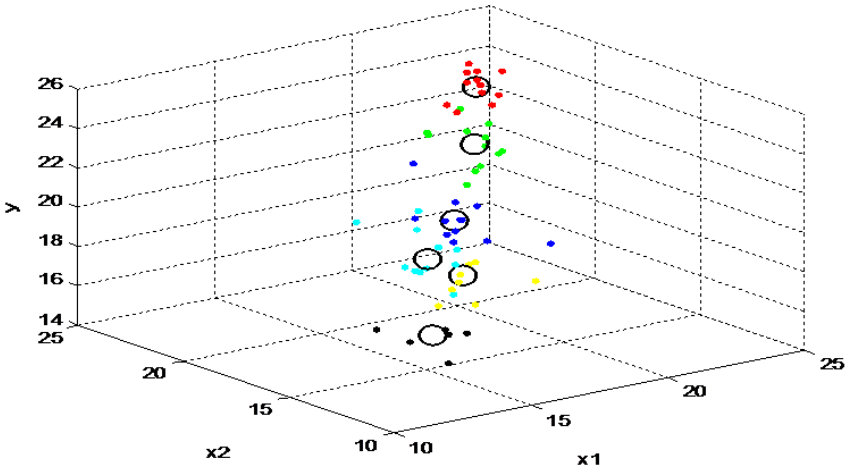


Fig. 6 *Time series data and respective clusters.*

c_{11}	23.1373	σ_{11}	18.6332	w_0	-5.5924
c_{12}	19.2555	σ_{12}	19.5991	w_1	-3.2244
c_{13}	21.7906	σ_{13}	16.7311	w_2	-55.8650
c_{14}	15.5351	σ_{14}	14.8883	w_3	122.6187
c_{15}	18.2458	σ_{15}	15.4140	w_4	1.5879
c_{16}	16.8230	σ_{16}	15.1891	w_5	-282.6299
c_{21}	23.1892	σ_{21}	17.0218	w_6	246.7825
c_{22}	19.2246	σ_{22}	18.3433		
c_{23}	21.5552	σ_{23}	18.1947		
c_{24}	15.3847	σ_{24}	13.7824		
c_{25}	17.4568	σ_{25}	15.3732	SSE	35.66
c_{26}	17.3302	σ_{26}	16.7565	MAPE	2.46%

Tab. IV *Values of the centers of the radial functions with K_m = 10 related to Example 4.*

Applying the approach proposed in this paper and using a scale factor with K_m=10 (value tested to achieve better accuracy in the neural model), we have the values of the widths and the values of the weights of the output layer.

Figure 7 shows the training data (black line) and the values estimated (blue line) with the RBF network, where the SSE was 35.66.

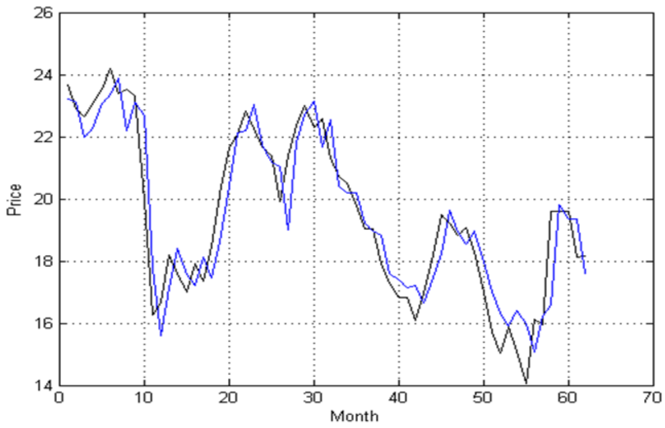


Fig. 7 *Time series data and values estimated with the neural network.*

Figure 8 shows the test data (black line) and the prices estimated (blue line) with the RBF network for eleven consecutive months (*N* = 11), where the MAPE was 2.46% (with some improvement in the accuracy of the model in relation to the previous neural network).

Other tests were considered through changes in the structure of the neural networks adopted, such as the use of more nodes in the hidden layers, larger numbers of input variables, more previous samples to model dynamic systems and time series. Applications in pattern recognition were also tested. These results will be presented in later publications.

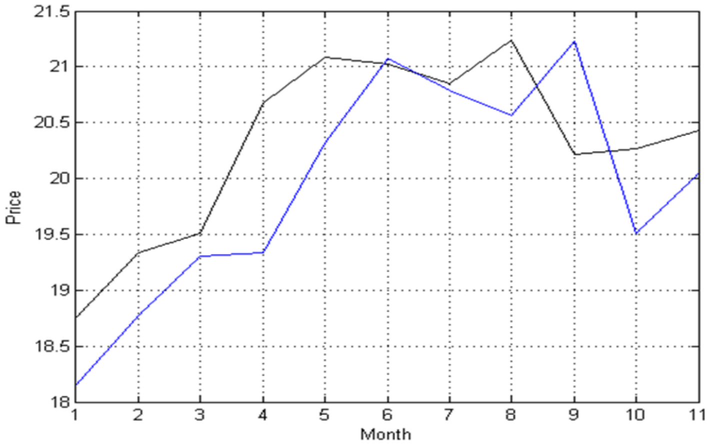


Fig. 8 *Time series data and the values estimated by the neural network.*

5. Conclusion

The procedure proposed in this paper, for adjusting of the parameters in RBF neural networks, is efficient and easy to use in different contexts of applications. The results obtained in examples of approximation of functions, modeling of systems with nonlinear characteristics and time series forecasting suggest promising potential for various applications in areas such as pattern recognition, control systems, signal processing, among others. The accuracy obtained in the approximations and estimations used in the examples suggest that the approach is an interesting alternative for the strategies of setting parameters for RBF neural networks.

The following suggestions are proposed for future investigations:

-The use in the first step of the procedure proposed in this paper of an approach to determine the number of minimal clusters for a particular application, for example, with the use of a criterion such as Fuzzy Silhouette [4], or others cited in the literature;

- The use of the procedure proposed in this paper as an initial parameter adjustment for a RBF neural network, which later (after the last stage of the procedure), will be refined by training techniques based on the gradient descent method, genetic algorithms, etc.

REFERENCES

[1] BABU G.S., SURESH S. Sequential Projection-Based Metacognitive Learning in a Radial basis Function Network for Classification Problems. *IEEE Transactions on Neural Networks and Learning Systems.* 2013, 24(2), pp. 194-206, doi: 10.1109/TNNLS.2012.2226748.

[2] BENOUDJIT, N., VERLEYSEN, M. On the kernel widths in Radial Basis Function Networks. *Neural Processing Letters*. 2003, 18(2), pp. 139-154, doi: 10.1023/A:1026289910256.

[3] BEZDEK J.C. A convergence theorem for the fuzzy isodata clustering algorithms. *IEEE Trans. On Pattern Analysis And Machine Intelligence*. 1980, 2(1), pp. 1-8, doi:10.1109/TPAMI.1980.4766964.

[4] CAMPELLO RJGB, HRUSCHKA ER. A fuzzy extension of the silhouette with criterion for cluster analysis. *Fuzzy Sets and Systems.* 2006, 157(21) pp. 2858-2875, doi:10.1016/j.fss.2006.07.006.

[5] GUILLÉN A, ROJAS I, GONZÁLES J, POMARES H, HERRERA L.J, VALENZUELA O, ROJAS F. Output value-based initialization for radial basis function neural networks. *Neural Processing Letters.* 2007, 25 (12):209–225, doi: 10.1007/s11063-007-9039-8.

[6] HAYKIN S. *Neural Networks: A comprehensive Foundation.* Prentice Hall, 2nd ed. 1999, Englewood Cliffs.

[7] JAFARNEJADSANI H, PIEPER J, EHLERS J. Adaptive Control of a Variable-Speed Variable-Pitch Wind Turbine Using Radial Basis Function Neural Network. *IEEE Transactions on Control Systems Technology*. 2013, 21(6):2264-2272, doi: 10.1109/TCST.2012.2237518.

[8] KEBRIAEI H, ARAABI BN, KIAN A.R. Short-Term Load Forecasting with a New Nonsymmetric Penalty Function. *IEEE Transactions on Power Systems*. 2011, 26(4):1817-1825, doi: 10.1109/TPWRS.2011.2142330.

[9] LOWE, D. Adaptive radial basis function nonlinearities, and the problem of generalization. *First IEEE International Conference on Artificial Neural Networks*, 1989, p. 171-175, London.

[10] MAHDI, R.N., ROUCHKA, E.C. Reduced Hyper BF Networks: Regularization by Explicit Complexity Reduction and Scaled Rprop-Based Training. *IEEE Transactions on Neural Networks.* 2011, 22(5):673-686, doi: 10.1109/TNN.2011.2109736.

[11] MOODY, J., DARKEN, C. J. Fast learning in networks of locally-tuned processing units. *Neural Computation,* 1989, n. 1, p. 281-295, doi:10.1162/neco.1989.1.2.281.

[12] NELLES, O. *Nonlinear System Identification.* Springer-Verlag, Berlin, 2001, doi: 10.1007/978-3-662-04323-3.

[13] SAHA, A., KEELER, J. D. Algorithms for better representation and faster learning in Radial basis Function Networks. *Advances in Neural Information Processing Systems 2,* edited by DavisS. Touretzky, 1989, p. 482-489.

[14] SHI Y, CHAN C.H. Comparison of Interpolating Functions and Interpolating Points in Full-Wave Multilevel Green's Function Interpolation Method. *IEEE Transactions on Antennas and Propagation.* 2010, 58(8):2691-2699, doi: 10.1109/TAP.2010.2050448.

[15] SU S.F, CHUANG C.C, TAO C.W, JENG J.T, HSIAO C.C. Radial basis Function Networks with Linear Interval Regression Weights for Symbolic Interval Data. *IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics.* 2012, 2(1):69-80, doi: 10.1109/TSMCB.2011.2161468.

[16] SUN YF, LIANG YC, ZHANG WL, LEE HP, LIN WZ, CAO L.J. Optimal partition algorithm of the RBF neural network and its application to financial time series forecasting. *Neural Comput. & Applic.* 2005, 14:36-44, doi: 10.1007/s00521-004-0439-7.

[17] ZHANG L, LI K, HE H.B. A New Discrete-Continuous Algorithm for Radial Basis Function Networks Construction. *IEEE Transactions On Neural Networks And Learning Systems*. 2013, 24 (11):1785-1798, doi:10.1109/TNNLS.2013.2264292.