"Mircea cel Batran" Naval Academy Scientific Bulletin, Volume XVI – 2013 – Issue 1
Published by "Mircea cel Batran" Naval Academy Press, Constanta, Romania

TRAINING OF RBF NEURAL NETWORKS: A COMPARATIVE OVERVIEW

Florin POPESCU[1]
Florin ENACHE[2]
[1] Lt. Col. Eng., Ph.D., Military Technical Academy
[2] Eng., Ph.D., Military Technical Academy

Abstract: *According to literature, it is well-known that the training algorithm of RBF neural networks depends a lot by the specific way to obtain the positioning of RBF centers over the input data space, and to fit the neural weights to the output layer, respectively. Having as starting point a real pattern recognition task belonging to video imagery to solve, this paper presents a comparative analysis of some standard and advanced approaches used to assure a high-quality training process of RBF neural networks.*

Key-words: *connexionist models, RBF neural network, pattern recognition*

1. INTRODUCTION

According to connexionist models theory, it is well-known that the *back-propagation* (BP) algorithm used to train the feedforward neural networks represents in fact, an optimization method belonging to the stochastic approximations class [1]. Another interesting way for a neural network design and training has as starting point the problem of curve approximation into R^n space and thus, the obtaining of *n*-D surface achieving the best matching with the input pattern set, respectively. Consequently, the capacity to assure the high-generalization of this new type of neural network can be successfully used for interpolation of the data from the available database [2]. Its hidden neural layer (see Fig.1) will also have the basic goal to generate a function set for an appropriate description of each input pattern, this representation space being made by so-called *radial basis functions* (RBF). In addition, the application area of the neural networks based on radial functions (i.e., RBF neural networks) is very large, starting with interpolation problems in *n*-dimensional space, regressions, prediction of the real time series and ending with the pattern recognition tasks, nonlinear systems identification etc.

Generally, a RBF neural network is usually trained to map a vector $x_k \in R^n$ into a vector $y_k \in R^p$, where the pairs $(x_k, y_k)_{k=\overline{1,M}}$, form the training set. If this mapping is viewed as a function in the input space R^n, learning can be seen as a function approximation problem. From this point of view, learning is equivalent to finding a surface in a multidimensional space that provides the best fit for the training data. Generalization is therefore synonymous with interpolation between the data points along the constrained surface generated by the fitting procedure as the optimum approximation to this mapping [2].

According to [3], the mathematical support of RBF neural networks design is assured by the *Cover theorem* (1965) referring to vectors/pattern separability: "*a complex classification problem can be resolved more easily in a space with more dimensions than in one with reduced dimension".* In addition, the fundamentals of RBF neural networks theory can be also recovered in Broomhead and Lowe scientific papers [4].

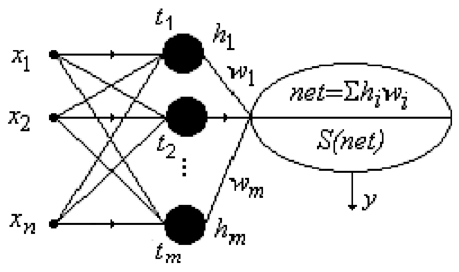

Figure 1. n-D standard RBF neural network topology

It is well-known that, the performance of a RBF neural network depends on the number and positions of the radial basis functions, their shapes, and the effective algorithm used for learning the input-output mapping. Consequently, the design and training process of a RBF neural network supposes virtually, two important steps [5], namely:

1) the positioning (or selection) of the centers $\{t_i\}_{i=\overline{1,m}}$ assigned to each neuron (i.e., radial basis function) from the hidden layer;

2) the fitting of the neural weights $\{w_i\}_{i=\overline{1,m}}$ from the hidden neural layer to the output neural layer (in this case, *p* denotes the number of the neurons on the output layer etc.).

Referring now at the RBF center selection technique, the following three basic classes of strategies are indicated in literature: strategies selecting radial basis function centers *randomly* from the training data [1], [4]; strategies employing *supervised* procedures for selecting radial basis function centers [2], [3], [5], [6] and finally, strategies employing *unsupervised* procedures for selecting radial basis function centers [6], [7], [8]. In addition, in order to solve some major drawbacks (e.g., difficulties due to network overfitting involving calculus time increasing, bad conditioning problems due to linear dependence caused by center proximity etc.) of the above mentioned techniques, other new hybrid approaches belonging to *AI paradigms* (e.g., a *genetic positioning* procedure of the RBF centers etc.) are also described in theory [8], [9].

This paper is aimed to present a comparative overview of some standard and advanced methods used to assure a high-quality training process for a RBF neural network. Consequently, in the first part of the paper, a comprehensive theoretical review of the tested RBF training procedures is indicated. In the next part of the paper, a shortly description of the modality to design the real database belonging to video imagery, is also described. In the last part of the paper, some suggestive experimental results for the paper goal are presented. Finally, the most important conclusions are also discussed.

2. TRAINING ALGORITHMS OF RBF NEURAL NETWORKS

As before mentioned, the most significant step for a high-quality RBF neural network training process is represented by the *RBF center selection* technique. Despite the existence of a variety of RBF center positioning methods indicated in theory, RBF neural networks are frequently trained in practice using the following consecrated methods:

• *random selection of the RBF centers*

In this case, the RBF centers are randomly positioning over the input data space. In addition, it is helpful to use a normalized gaussian function as radial function because its shape is independent by the center spreading. In order to assure the fitting of the neural weights to the output layer for example, the *singular value method* (SVM) proposed by Haykin can be used [1].

• *supervised selection of the RBF centers*

In this case, all parameters which described the RBF neural network connectivity are achieved using a supervised training procedure. For example, the well-known method is based on *error correction* technique using a *descendent gradient* algorithm [3]. According to [1], using the *regularization method* (RM) proposed by Haykin, the interesting parameters of RBF neural network are given by the following equations:

- RBF centers selection:

$$\begin{cases} t_i(n+1) = t_i(n) - \eta \cdot \dfrac{\partial E(n)}{\partial t_i(n)}, \quad i = \overline{1,m} \\ \dfrac{\partial E(n)}{\partial t_i(n)} = 2w_i(n)\displaystyle\sum_{j=1}^{s} e_j(n)G'(\left\|x_j - t_i(n)\right\|^2)\Sigma_i^{-1}\left[x_j - t_i(n)\right] \end{cases}, \tag{1}$$

where $E(\cdot)$ represents the error-function and $G(\cdot)$ is a radial function by Green type;

- neural weights fitting:

$$\begin{cases} w_i(n+1) = w_i(n) - \eta \cdot \dfrac{\partial E(n)}{\partial w_i(n)}, i = \overline{1,m}, \\ \dfrac{\partial E(n)}{\partial w_i(n)} = \displaystyle\sum_{j=1}^{s} e_j(n)G(\left\|x_j - t_i(n)\right\|^2) \end{cases}. \tag{2}$$

More details about Haykin's regularization method can be found in [5].

• *unsupervised selection of the RBF centers*

In this case, the selection process supposes a self-organizing of the center positions over the input data space, and the neural weights to the output layer are calculated using a proper supervised training rule etc.

The positioning of the RBF centers are made only in the input data space zones which contain significant pattern clusters [1]. In order to achieve this task for example, some well-known (standard or neural) *clustering techniques* can be used, like: grouping algorithms (*k*-means, ISODATA etc.), Kohonen algorithm used in case of SOM networks, unsupervised competitive clustering algorithm etc. In addition, some fuzzy versions of the above mentioned clustering techniques can also be used [1], [8].

According to [6], the *unsupervised competitive clustering algorithm* (UCC) proposed by Brown supposes two important stages, namely:

1) the training of the hidden neural layer using a specific grouping method usually, *k*-means or ISODATA algorithms. Finally, the dispersions $\{\sigma_i\}_{i=\overline{1,m}}$ assigned to each hidden neuron are calculated using the following equation:

$$\sigma_i^2 = \frac{1}{p_i}\sum_{x_i \in A_i}(x_i - t_i)^T(x_i - t_i), i = \overline{1,m}, \tag{3}$$

where $\{A_i\}_{i=\overline{1,p}}$ is the partition of the input data space made by used clustering technique and p_i is the number of patterns from A_i cluster, respectively;

2) the training of the output neural layer using the *orthogonal least square algorithm* (OLS) described in [5]. Consequently, the output neural weights matrix W_{out} is obtained as solution of the following normalized equation:

$$W_{out} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T d, \tag{4}$$

where **H** denotes the RBF functions matrix, and *d* represents the vector of the desired outputs etc.

• *GARBF concept*

Generally, although the standard clustering algorithms had been applied to many practical clustering problems successfully, it has been shown that these algorithms may fail to converge to a global minimum under certain conditions that represents certainly, an important disadvantage. Since genetic algorithms are powerful global searching tools, and are most appropriate to solve complex nonlinear problems (where location of the global solution is a difficult task), then they can be applied with a high-level of performance to find the proper number of the clusters from the input data space [1], [9].

The main idea related to the mixing mode between *genetic algorithms* (GA) and *neural network* (NN) theory has as starting point the encoding of the information about the neural network architecture in the genome of genetic algorithm. Consequently, it can be observed that this general design (or optimization) procedure of the neural networks is one as soon as directly, and the most important problem of GANN systems consists in fact, in the finding of the specific encoding way of the neural network etc.

According to [9], an algorithm which assures a *genetic positioning* of the RBF centers over input data space (i.e., a GARBF system) contains the following important stages, namely:

1) if the training dataset has the form $\{\mathbf{X}_k, d_k\}_{k=\overline{1,P}}$, $\mathbf{X}_k \in \mathrm{R}^n$ and *c* is the number of the classes from input space, then using an *adaptive clustering* algorithm (e.g., ISODATA) the major tendencies from inside of each data cluster are determined. Therefore, based on this clustering method use, inside of each main data cluster is bounded into $m_j \left(m = \sum_{j=1}^{c} m_j \right)$ new subclusters;

2) the starting chromosome population is made using a random choice of m_j vectors $\mathbf{X}_k$ from each class (one vector for each bounded subcluster) and finally, a vector linear concatenation. Therefore, each chromosom will have assigned *m* vectors $\{t_i\}_{i=\overline{1,m}}$ which are extracted from the training dataset. Also, in order to provide a suitable representation, it was used a real coding technique;

3) because in this moment RBF setting parameters $\{t_i, \sigma_i\}_{i=\overline{1,m}}$ are known, the neural weigths to output layer $\{w_i\}_{i=\overline{1,m}}$ can be easily calculated using for example, OLS algorithm etc.

More specific details about the GA structure used in this case can be found in [1] and [9].

Finally, it is important to know that, the above described training methods of a RBF neural network will be comparative analyzed in the experimental part of the paper. In addition, more theoretical details about these techniques can be also found in [1], [5], [8] and [9].

3. DATABASE DESIGN

In order to add more consistency to the experimental part of the paper, the comparative analysis of the above described training techniques was made using a *real* pattern recognition (PR) task belonging to the video imagery (i.e., the PR task was to classify 5 input classes representing CCD images of some well-known military aircrafts).

The video database was obtained using a (digital) photographical survey of *five* military aircraft models (i.e., F117, Mirage 2000, Mig 29, F16, and Tornado) scaled each at 1:48 ratio (see Fig.2). In addition, the survey was taken using a 5^0 increment into azimuthal plane (see Fig.3) using an angular range of $\left[0^0, 180^0\right]$, justified by the geometrical aircraft shape symmetry. Consequently, a number of 37 video images/class was achieved etc.

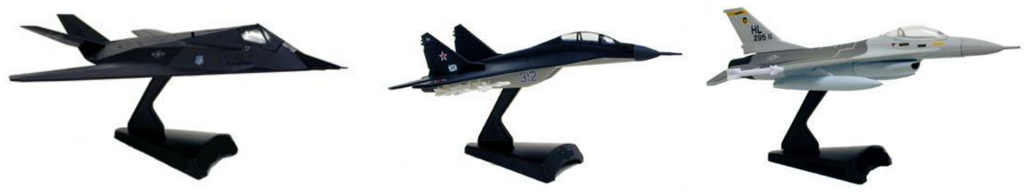

Figure 2. Some aircraft models used in video database design

In order to achieve the feature extraction stage, the Fourier invariants were calculated. Consequently, the most significant 11 Fourier invariants were retained for the next classification stage. In addition, in order to assure the feature selection stage, the standard Sammon projection algorithm was also used (implementing a projection by $R^{11} \rightarrow R^n$ type) etc.

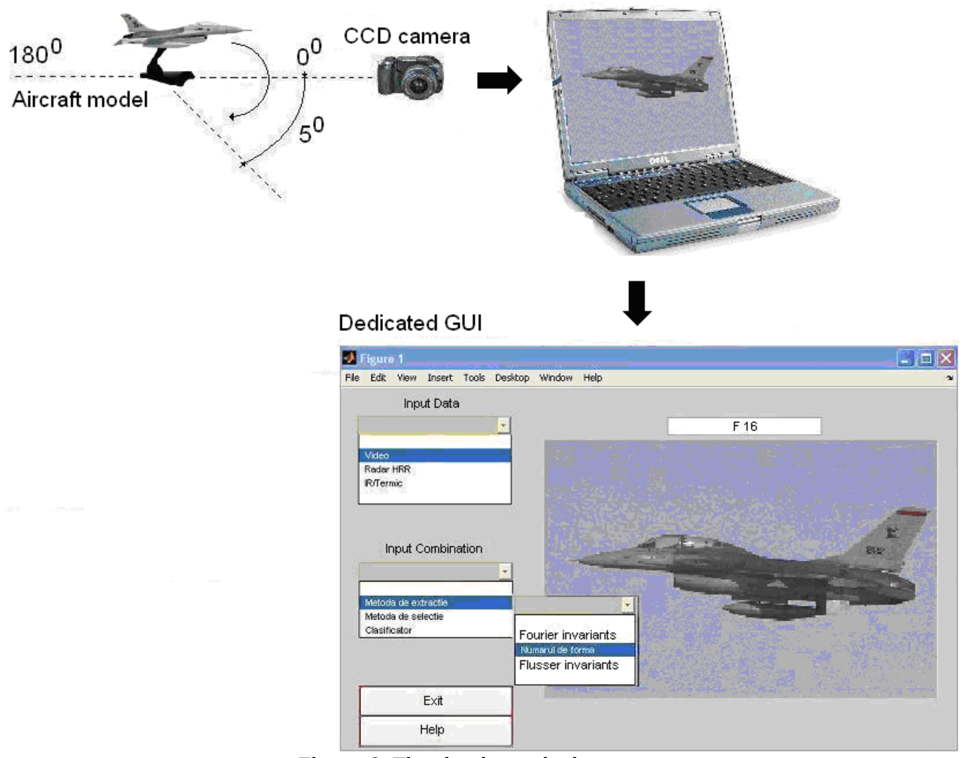

Figure 3. The database design setup

More details about the concrete way to achieve the video database can be found in [1] and [10].

4. EXPERIMENTAL RESULTS

The main goal of the experimental part of the paper was to comparative analyse (as performance level) the following training techniques of a RBF neural network, namely: a random selection of RBF centers [1], regularization method [5], UCC algorithm [6] and a GARBF system [9], respectively (see Fig.4).

In order to quantify the PR performance level, the *classification rate* (CR) as the most important performance parameter was computed (it is known that CR represents in [%], the ration between the number of correct classified patterns and total number of available input patterns etc.). In addition, other relevant parameters were also computed, like: training time of the RBF neural network, RBF neural network connectivity (see Table 1) etc.

Consequently, using the available video database, the experimental results achieved by simulations are synthetically illustrated in Table 1. In addition, in case of GARBF system, a minimal 2D projection of the RBF centers mapping over the input data space is also shown in Fig.5.

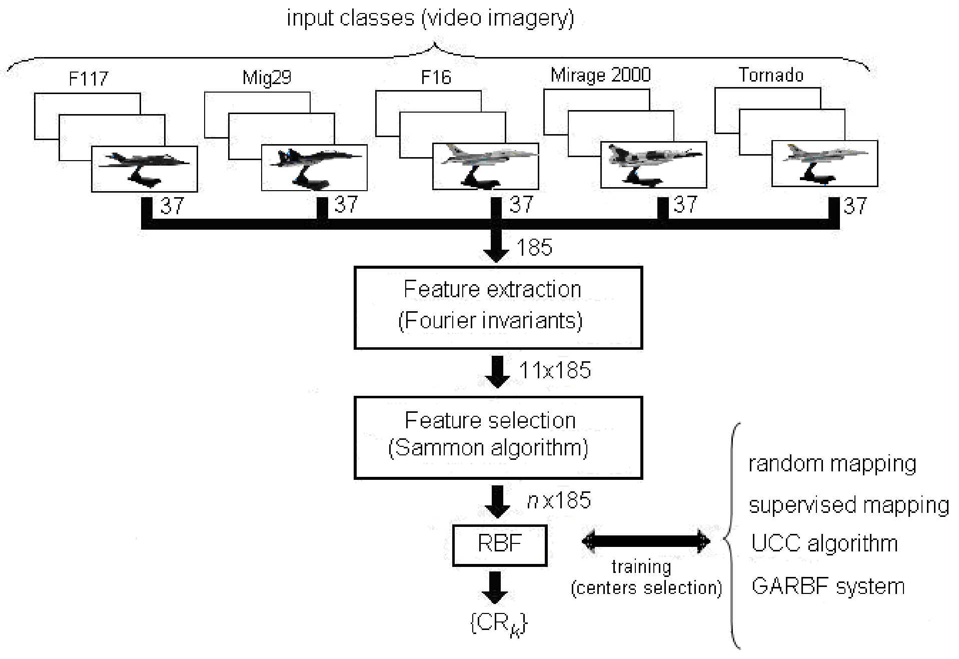
input classes (video imagery)
F117
Mig29
F16
Mirage 2000
Tornado
37
37
37
37
37
185
Feature extraction (Fourier invariants)
11x185
Feature selection (Sammon algorithm)
nx185
RBF
{CR_k}
training (centers selection)
random mapping
supervised mapping
UCC algorithm
GARBF system

Figure 4. The experimental setup

Table 1

Center selection	Classification performances (CR and others training parameters)	
random mapping	90% n=7; m=8; 0.1 s; nepochs=10^4; $\varepsilon=10^{-4}$; $\sigma=1$	
supervised mapping	93.5% n=7; m=10; 0.15 s; nepochs=10^4; $\varepsilon=10^{-4}$; $\sigma=1$	
UCC algorithm	96.5% n=7; m=11; 0.21 s; nepochs=10^4; $\varepsilon=10^{-4}$; $\sigma=0.8$	
GARBF system	98.5% n=6; m=14; c=5; 0.67 s nepochs=10^4 $\varepsilon=10^{-4}$; $\sigma=0.8$	**GA modules**
		155 s maxpop=50; maxgen=100 p_c=0.8; $\varepsilon_0=10^{-2}$
		121 s maxpop=50; maxgen=100 p_c=0.85; $\varepsilon_0=10^{-3}$; k=0.75

Experimental results

Having as starting point the experimental results reported in Table 1, a first preliminary remark is that the best classification results (as it is expected) are achieved in case of GARBF system concept use. In addition, in this study case, a 5.2% average increasing of CR related to the other tested approaches was also obtained etc.

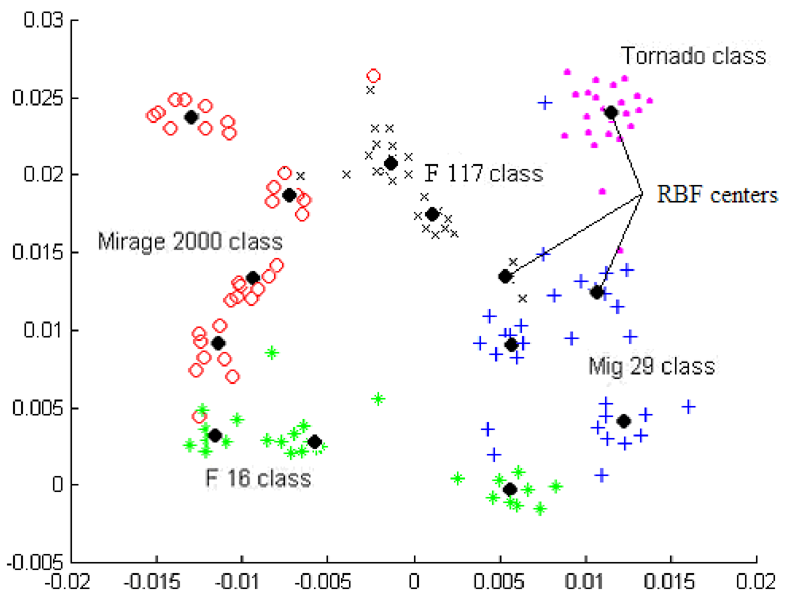

Figure 5. RBF centers mapping in case of GARBF system use

As it can be seen from previous figure, a second basic remark is that, the genetic selection of the RBF network centers leads to a very good positioning of these over input data space (i.e., each significant data cluster had allocated at least a RBF center etc.), even though it was used only a 2D minimal representation.

All computer simulations designed in this experimental part of the paper were implemented using *nnet* and *gaot* MATLAB toolboxes. In addition, more details about the effective implementation can be found in [11].

5. CONCLUSIONS

This paper presented as primary goal, a comparative analysis of some standard and advanced methods used to assure a high-quality training process for a RBF neural network. Based on a real database belonging to video imagery, the achieved experimental results confirmed the superiority as performance level, of the unsupervised techniques in assuring of the RBF centers mapping related to other standard approaches. In addition, the best PR results were obtained in case of advanced GARBF system concept implementation (more than 98%), although all tested approaches provided very good CRs (i.e., more than 90%).

On the other hand, the required computing resources to implement all tested training techniques of a RBF neural network are reasonable and thus, using dedicated processing hardware tools (e.g., DSP, FPGA technology etc.), their practical implementation can also become an acceptable task.

In summary, this comparative overview demonstrated the indubitable superiority as performance level, of the advanced training approaches based on AI paradigms use. In addition, this analysis can represent a good starting point into investigation and design of new techniques for RBF neural network training.

REFERENCES

[1] I.C. Vizitiu, *Neuro-fuzzy-genetic architectures. Theory and applications*, MTA Publishing House, Bucharest, 2011

[2] L. Jain and A.M. Fanelli, *Recent advances in artificial neural networks. Design and applications*, CRC Press, Boca Raton, 2000

[3] M.M. Gupta, L. Jin and N. Homma, *Static and dynamic neural networks*, IEEE Press, John Wiley&Sons, New Jersey, 2003

[4] D.S. Broomhead and D. Lowe, *Multivariable functional interpolation and adaptive networks*, Complex Systems, vol. 2, pp. 321-355, 1988

[5] Y.H. Hu and J.N. Hwang (editors), *Handbook of neural network signal processing*, CRC Press, Boca Raton, 2002

[6] B.D. Ripley, *Pattern recognition and neural networks*, Cambridge University Press, 2004

[7] N.B. Karayiannis and M.R. Gips, *The construction and training of reformulated RBF neural networks*, IEEE Transaction on Neural Networks, no. 4, pp. 835-844, 2003

[8] R. Howlett and L.C. Jain, *RBF networks: new advances in design*, Physica-Verlag, 1st Edition, Heidelberg, 2001

[9] I.C. Vizitiu, *Advanced ATR system using improved neural recognition and decision fusion techniques*, Proceedings of the International Conference ICMT, Brno, Czeck Republic, pp. 47-52, 2013

[10] F. Popescu, I.C. Vizitiu and A. Stoica, *High-quality ATR system using an improved neural recognition chain*, Proceedings of the IEEE International Conference Communications, Bucharest, Romania, pp. 217-220, 2010

[11] H. Demuth and M. Beale, *Neural network toolbox (for use with MATLAB)*, version 4, MathWorks Inc., 2000