

An Immunological Approach to Initialize Centers of Radial Basis Function Neural Networks

Leandro Nunes de Castro & Fernando J. Von Zuben
{lnunes,vonzuben}@dca.fee.unicamp.br

School of Electrical and Computer Engineering (FEEC) – UNICAMP

Abstract

The appropriate operation of a radial basis function (RBF) neural network depends mainly upon an adequate choice of the number and positions of its basis function centers. The simplest approach to train an RBF network is to assume fixed radial basis functions defining the activation of the hidden units, followed by the application of a regression procedure to determine the linear output weights. The main drawback of this strategy is the lack of an efficient algorithm to determine the amount and positions of the RBF centers. In this paper, an immunological approach makes use of the training data in order to initialize the radial basis functions. The approach to be proposed is inspired by the vertebrate immune system, and tries to compress the information contained in the data set while positioning the prototype vectors into representative regions of the input space. The algorithm is compared to random and k -means center selection, and results are reported concerning regression and classification problems.

1. Introduction

RBF neural networks are powerful function approximators for multivariate nonlinear continuous mappings. They have a simple architecture and the learning algorithm corresponds to the solution of a linear regression problem, resulting in a fast training process. The RBF network behavior strongly depends upon the number and position of the basis functions at the hidden layer. Traditional methods to determine the centers are: randomly choose input vectors from the training data set [1]; vectors obtained from unsupervised clustering algorithms, such as k -means, applied to the input data [2]; or vectors obtained through a supervised learning scheme [3].

Over the last few years the immune system has been used as a rich source of inspiration to develop new computational tools for solving complex engineering problems, among which we can stress computer and network security [4, 5, 6], optimization problems [7, 8], and robotic control [9]. Recently, de Castro & Von Zuben [7] proposed a data compression algorithm to solve classification problems described by a set of unlabeled data. In this paper, we will show that, with a few refinements, this algorithm can be directly applied to the problem of defining RBF network centers.

2. RBF neural network

An RBF neural network can be regarded as a feedforward network composed of three layers of neurons with entirely different roles (Figure 1) [11]. The input layer is made up of sensory units that connect the network to its environment. The second layer, the single hidden layer, applies a nonlinear transformation from the input space into the hidden space. The nonlinear hidden units are locally tuned and their responses are outputs of radial basis functions. The output layer is linear, supplying each network response as a linear combination of the hidden responses [12, 3].

For a p -dimensional input vector $\mathbf{x} = (x_1, x_2, \dots, x_p)$, where $\mathbf{x} \in \mathbf{X} \subset \mathfrak{R}^p$, the RBF network outputs can be computed by the following expression:

$$y_i = \mathbf{w}_i^T \mathbf{g} = \sum_{j=1}^m w_{ij} g_j, \quad i=1, \dots, o, \quad (1)$$

where $\mathbf{w}_i = [w_{i1}, \dots, w_{im}]^T$, $i=1, \dots, o$, are the network weight vectors for each output neuron i , $\mathbf{g} = [g_1, g_2, \dots, g_m]^T$ is the vector of basis functions, and o is the number of network output units. Given a set of prototype vectors $\mathbf{c}_j \in \mathfrak{R}^p$ and standard deviations σ_j , $j=1, \dots, m$, the output of each radial basis function is

$$g_j = h(\|\mathbf{x} - \mathbf{c}_j\|, \sigma_j), \quad j=1, \dots, m, \quad (2)$$

where $h_j(\cdot)$ is the *basis function* and $\|\cdot\|$ is a norm, usually the Euclidean norm, defined on the input space.

In the context of an interpolation problem, a mapping function $y: \mathfrak{R}^p \rightarrow \mathfrak{R}$ satisfying Equation (1), for $o=1$, has to be determined. Consider a set of N data points $\{\mathbf{x}_i \in \mathfrak{R}^p \mid i=1, \dots, N\}$. If the desired output values are known for all these N data points, i.e. $\{d_i \in \mathfrak{R} \mid i=1, \dots, N\}$, then each basis function may be centered on one of these data points. So, there are as many centers (prototype vectors) as data points: $m=N$ [13]. In matrix notation,

$$\mathbf{H} \mathbf{w} = \mathbf{d}, \quad (3)$$

where the N -by-1 vectors \mathbf{d} and \mathbf{w} represent the desired response vector and the output weight vector, respectively. \mathbf{H} is an N -by- N matrix, called interpolation matrix. The solution to the problem stated in Equation (3) is given by

$$\mathbf{w} = \mathbf{H}^{-1} \mathbf{d} \quad (4)$$

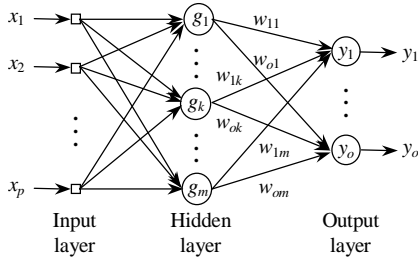


Figure 1: Radial Basis Function (RBF) network.

Micchelli's theorem [14] stated that the only prerequisite for the existence of \mathbf{H}^{-1} is that the N data points are *distinct*, regardless the values of N and p .

According to Broomhead & Lowe [11], the strict interpolation problem described above may sometimes not be a good strategy for the training of RBF networks because of poor generalization to previously unseen data. In addition, if N is large, and/or if there is a great amount of redundant data (linearly dependent vectors), the likelihood of obtaining an ill-conditioned matrix \mathbf{H} will be higher. The constraint of having as many RBFs as data points makes the problem overdetermined. To overcome these computational difficulties, the complexity of the network has to be reduced, requiring an approximation to a regularized solution [15]. This approach involves the search for a suboptimal solution in a lower dimensional space. A new set of m_1 basis functions ($m_1 < N$), assumed to be linearly independent, has to be defined. The set of centers $\{\mathbf{c}_j | j = 1, \dots, m_1\}$ must be determined, such that in the case of $m_1 = N$, $\mathbf{d}_i = \mathbf{x}_i$, $\forall i$. If we disregard the regularization parameter, the solution \mathbf{w}^* to the least-squares data-fitting problem, for $m_1 < N$, is simply given by

$$\mathbf{w}^* = \mathbf{H}^+ \mathbf{d} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{d}, \quad (5)$$

where \mathbf{H}^+ is the pseudo-inverse of the matrix \mathbf{H} [11].

Haykin [12] argued that experience with this method showed it is relatively insensitive to the choice of regularization parameters, as far as an appropriate choice of the RBF centers is performed.

So, the approach used here to train an RBF neural network is to assume fixed radial basis functions as the activation of the hidden units. The locations of the centers might be chosen somehow, usually at random, or from the training data set as will be proposed in this paper, using an immune-inspired approach. For the RBFs, we will employ a Gaussian function whose standard deviation is fixed according to the spread of the centers

$$h(\|\mathbf{x} - \mathbf{c}_j\|, \sigma_j) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_j\|^2}{\sigma_j^2}\right), \quad (6)$$

where $j = 1, \dots, m_1$ is the number of centers (hidden units), $\sigma_j = d_{\max} / \sqrt{2m_1}$ is the standard deviation (the same for all basis functions), d_{\max} is the maximum distance between the chosen centers, \mathbf{x} is the input vector, and \mathbf{c}_j is the j -th center location.

This choice for σ_j guarantees that the individual radial basis functions are not too peak or too flat; conditions that should be avoided [12].

So, the only parameters to be learnt are the weights in the network output. A straightforward procedure for doing so is to use the pseudo-inverse method presented in Equation (5).

3. A brief review of immunology

Among the many types of immune principles, two were used to determine the location of the centers: (1) the clonal selection principle, and (2) the immune network theory. An interested reader might refer to the book of Janeway Jr. & Travers [16] as a good introduction to immunology, and to the technical report by de Castro & Von Zuben [17] for an introduction to the artificial immune systems and their applications.

The clonal selection principle explains how the adaptive immune system (IS) responds to the invasion of a pathogenic microorganism, called antigen (Ag). When an antigen invades our bodies, a subset of the immune cells recognizes these antigens, through a complementary match, and is selected for reproduction. The cellular reproduction is asexual (mitosis or cloning). During reproduction, the clones suffer a hypermutation process that alters their shapes with relation to the parents, creating the possibility of improving their ability to recognize the selective antigen. Those new cells, whose affinity with the antigen is improved, are rescued to become part of a memory set that will act in future responses. Another subset of the clones becomes plasma cells, characterized by a high rate of antibody secretion. Antibodies (Ab) are molecules (cell receptors) that bind to antigens for their posterior elimination. A schematic overview of the clonal selection principle is depicted in Figure 2.

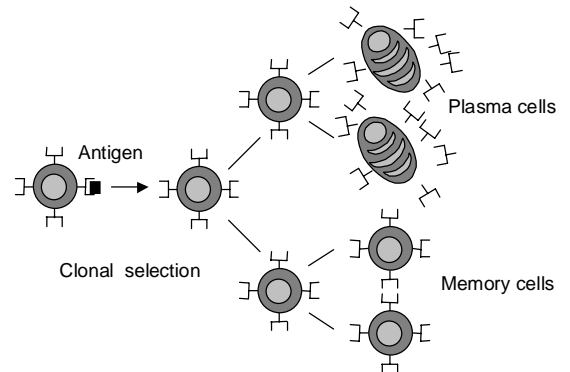


Figure 2: The clonal selection principle. An antigen is recognized by an immune cell, resulting in its proliferation and differentiation into memory and plasma cells. A memory cell is a long living cell presenting high affinity with the antigenic stimulus, while a plasma cell secretes antibodies in a high rate.

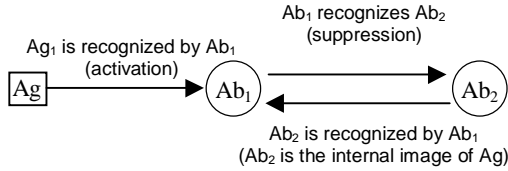


Figure 3: The immune network idea. An antigen (Ag) and an antibody (Ab₂) are recognized by Ab₁. The Ag recognition promotes a network activation, while the Ab₂ recognition promotes a network suppression. Ab₂ is considered the internal image of the antigens.

The immune network theory, as originally proposed by Jerne [18], hypothesized mainly a novel viewpoint of self/nonself discrimination, i.e., how the immune system differentiates between our own cells and pathogenic substances (antigens). The immune system was formally defined as an enormous and complex network of cells and molecules that recognize each other even in the absence of antigens. The relevant events in the immune system are not only the molecules, but also their interactions. The immune cells can respond either positively or negatively to the recognition signal. A positive response would result in cell *proliferation*, *activation* and antibody secretion, while a negative response would lead to *suppression*. Figure 3 depicts the immune network idea.

The central characteristic of the immune network theory is the definition of the individual's molecular identity (*internal images*), which emerges from a network organization followed by the learning of the molecular composition of the environment where the system develops. The network approach is particularly interesting for the development of computational tools because it potentially provides a precise account of emergent properties such as learning and memory, self-tolerance, size control and diversity of cell populations. In general terms, the structure of most network models can be represented as

$$\text{RPV} = \text{INC} - \text{DUC} + \text{RSC} \quad (7)$$

where RPV is the rate of population variation, INC is the increase of network cells, DUC is the death of unstimulated cells, and RSC is the reproduction of stimulated cells that includes Ab-Ab recognition and Ag-Ab stimulation.

4. RBF center selection: an immunological approach (ICS)

In this paper, we will use an immune-inspired learning algorithm [7], called aiNet, disregarding the steps responsible for constructing the network architecture, to define a data clustering algorithm to select radial basis function centers, resulting in the ICS algorithm. In the ICS algorithm, each input pattern \mathbf{x}_i , $i = 1, \dots, N$, corresponds to an antigenic stimulus, while

each candidate center, \mathbf{z}_j , $j = 1, \dots, m_1$, corresponds to an antibody.

Basically, the ICS algorithm works as follows:

1. A set of candidate centers is initialized at random, where the initial number of candidates and their positions is not crucial to the performance;
2. The clonal selection principle will control which candidates will be selected and how they will be updated;
3. The immune network theory will identify and eliminate (suppress) self-recognizing individuals, controlling the number of candidate centers.

Note that the clonal selection principle will be responsible for how the centers will represent the training data set. On the other hand, the immune network theory will avoid data redundancy, such that Micchelli's theorem applied to Equation (4) is satisfied (see Section 2). In fact, as the resulting number m_1 of prototype vectors $\{\mathbf{c}_j | j = 1, \dots, m_1\}$, will be much smaller than the number of training patterns, $m_1 \ll N$, the extended result of Broomhead & Lowe [11] will be used to define the network output weights, as described by Equation (5).

If we assume that the set of N patterns $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, $\mathbf{x}_i \in \mathcal{R}^p$, $i = 1, \dots, N$, to be used as the inputs of an RBF neural network is a set of unlabeled data, the data compression problem results in the determination of a new set $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_{m_1}\}$ composed of m_1 patterns ($\mathbf{z}_j \in \mathcal{R}^p$, $j = 1, \dots, m_1$), where $m_1 \ll N$. \mathbf{Z} is not necessarily a subset of \mathbf{X} .

The elements composing the \mathbf{Z} set will serve as *internal images* of the \mathbf{X} set, and will be responsible for mapping existing clusters in the data set into RBF centers. Figure 4 presents hypothetical elements of \mathbf{X} (stars) and the respective elements of \mathbf{Z} (labeled circles) generated by the ICS algorithm. Notice that the number of elements in \mathbf{Z} is much smaller than the number of elements in \mathbf{X} ($m_1 \ll N$).

The \mathbf{Z} set is initialized randomly with a fixed number of elements that will compete with each other for pattern recognition, and those successful will proliferate, i.e., generate copies subject to mutation, while those who fail recognition will be eliminated. In addition, if two elements from the \mathbf{Z} set recognize each other (based upon a given distance metric, assumed to be the Euclidean distance in this work), a suppressive step will occur. In our model, suppression is performed by eliminating the self-recognizing elements from \mathbf{Z} , given a suppression threshold σ_s . Every pair \mathbf{x}_i - \mathbf{z}_j (representing an Ag-Ab pair), will be related to each other in a metric-space S through the affinity d_{ij} of their interactions (dissimilarity), which reflects the probability of starting a clonal response, according to the clonal selection principle discussed in the previous section. Similarly, an affinity s_{ij} will be assigned to each pair \mathbf{z}_i - \mathbf{z}_j , reflecting their interactions (similarity).

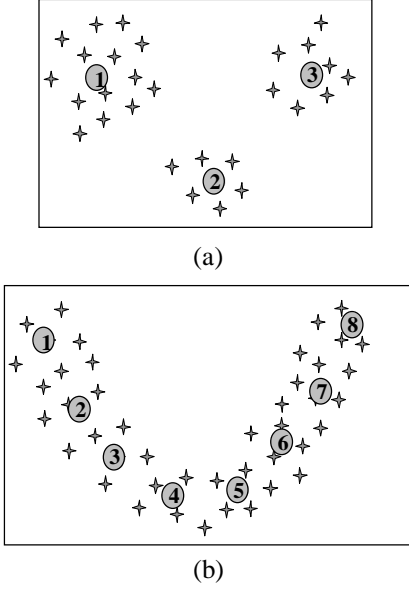


Figure 4: Illustration of the performance of the data compression algorithm (ICS) for pattern classification and approximation, respectively. The stars represent elements of \mathbf{X} , and the numbered circles the resulting labeled elements of \mathbf{Z} . Notice that each element of \mathbf{Z} represents a group (cluster) of elements belonging to \mathbf{X} . (a) 3 elements composing the set \mathbf{Z} of RBF centers are defined by the ICS approach. (b) The elements from set \mathbf{Z} will determine the number and positions of RBF centers such that the function can be appropriately approximated.

The following notation will be adopted:

\mathbf{X} : matrix with the original data set as its rows ($\mathbf{X} \in \mathfrak{R}^{N \times p}$);

\mathbf{T} : temporary matrix containing a number N_t of candidate centers as its rows ($\mathbf{T} \in \mathfrak{R}^{N_t \times p}$);

\mathbf{C} : matrix with the m_1 centers (memory cells), as its rows, taken from the rows of \mathbf{T} ($\mathbf{C} \in \mathfrak{R}^{m_1 \times p}$);

N_c : no. of clones generated by each stimulated center;

\mathbf{D} : dissimilarity matrix with elements $d_{ij}(\mathbf{x}-\mathbf{z})$;

\mathbf{S} : similarity matrix with elements $s_{ij}(\mathbf{z}-\mathbf{z})$;

n : n highest affinity centers selected for reproduction and mutation;

ζ : percentage of the matured centers to be selected;

σ_d : natural death threshold; and

σ_s : suppression threshold.

Each RBF candidate center is considered to be an element, or cell, of the immune network model. The learning algorithm aims at building a memory set, arranged as the rows of matrix \mathbf{C} , that recognizes and represents the data structural organization. The more specific the centers (cells), the less parsimonious the network¹ (low compression rate), whilst the more generalist the centers, the more parsimonious the network with relation to the number of centers

¹ Note that *network* in this section refers to the immune network model, and might not be confused with the RBF neural network.

(improved compression). The suppression threshold (σ_s) controls the specificity level of the centers, the clustering arrangement and the network plasticity. As a suggestion, the user must first set a small value for σ_s (e.g., $\sigma_s \leq 10^{-3}$) and continuously fine-tune the network performance.

The ICS learning algorithm works as follows:

1. At each iteration step, do:
 - 1.1 For each input pattern i , do:
 - 1.1.1 Determine its affinity, d_{ij} , to all the network centers according to a distance metric;
 - 1.1.2 Select the n highest affinity network centers;
 - 1.1.3 Reproduce (clone) these n selected centers. The higher the center affinity, the larger the number of clones (N_c);
 - 1.1.4 Apply Equation (9) to these N_c centers;
 - 1.1.5 Determine \mathbf{D} for these improved centers;
 - 1.1.6 Re-select $\zeta\%$ of the highest affinity centers and create a partial \mathbf{C}_p memory centers matrix;
 - 1.1.7 Eliminate those centers whose affinity is inferior to threshold σ_d , yielding a reduction in the size of the \mathbf{C}_p matrix;
 - 1.1.8 Calculate the network $\mathbf{x}_i-\mathbf{z}_j$ affinity, s_{ij} ;
 - 1.1.9 Eliminate those candidates with $s_{ij} < \sigma_s$ (clonal suppression);
 - 1.1.10 Concatenate \mathbf{C} and \mathbf{C}_p , ($\mathbf{C} \leftarrow [\mathbf{C}; \mathbf{C}_p]$);
 - 1.2 Determine \mathbf{S} , and eliminate those centers whose $s_{ij} < \sigma_s$ (network suppression);
 - 1.3 Replace $r\%$ of the worst individuals;
2. Test the stopping criterion.

In steps 1.1.1, 1.1.5 and 1.1.8 we adopted the Euclidean distance as a metric of similarity and dissimilarity. Steps 1.1.1 to 1.1.7 describe the clonal selection and hypermutation processes. Steps 1.1.8 to 1.1.10 and 1.2 to 1.3 simulate the immune network activity. The affinity of the cells with the given input pattern \mathbf{x}_i can be improved by the following expression (biased mutation):

$$\mathbf{c}_j = \mathbf{c}_j - \alpha_j (\mathbf{c}_j - \mathbf{x}_i), \quad \forall j, \quad (8)$$

where, \mathbf{c}_j is the j -th clone (j -th row of matrix \mathbf{C}) and α_j is the *learning rate*, or *mutation rate*. The α_j value is set according to the $\mathbf{x}_i-\mathbf{z}_j$ affinity, the higher the affinity, the smaller the α_j . Equation (9) proposes a biased search, where the $\mathbf{x}-\mathbf{z}$ complementarity is increased proportionally to α . By doing so, we guide our search to locally optimize the network cells (*greedy search*) in order to improve their pattern recognition capability along the iterations.

As can be seen from this algorithm, a clonal immune response is elicited to each presented input pattern. Notice also the existence of two suppressive steps in this algorithm (1.1.9 and 1.2): the *clonal suppression* is responsible for eliminating intra-clonal self-recognizing centers, while the *network suppression* searches for similarities between different sets of network clones. After the learning phase, the network centers represent internal images of the input patterns

(or groups of patterns) presented to it. As a complement to the general network structure presented in Equation (8), our model suppresses self-recognizing centers (steps 1.1.9 and 1.2).

The ICS output is taken to be the matrix of memory cells' coordinates (C) that represent the final centers to be adopted by the RBF network. The stopping criterion of the ICS algorithm is given by a pre-defined number of iterations.

5. Performance evaluation

In order to evaluate the performance of the proposed immune-inspired algorithm for RBF center selection (ICS), it was applied to a regression and a classification problem. This results were compared to those of the random initialization technique, presented in Section 2, and to the k -means method. The k -means method classifies an input vector by assigning it the label most frequently represented among the k nearest samples [2].

The stopping criterion, SC , for the ICS method is a fixed number of generations, and CR represents the percentile compression rate obtained by the proposed strategy, i.e. the ratio between the resulting amount m_1 of RBF centers and the total amount N of training data. Note that the ICS method is stochastic in nature, so each time the algorithm is run, a slightly different network architecture (amount and center positions) might arise, but with a similar performance. Typical results will be presented here.

In the first problem tested, REG, it was generated 315 data samples for one period of the $\sin(x)$ function with noise uniformly distributed over the interval $[-0.7, +0.7]$ (see Figure 5). Notice that the ICS algorithm is trained only with the input data, disregarding the desired outputs (unsupervised learning).

Figure 5 also presents the performance of the three algorithms applied to this problem. The resulting number of RBF centers for the ICS method was 8, corresponding to a compression rate of 97.46%.

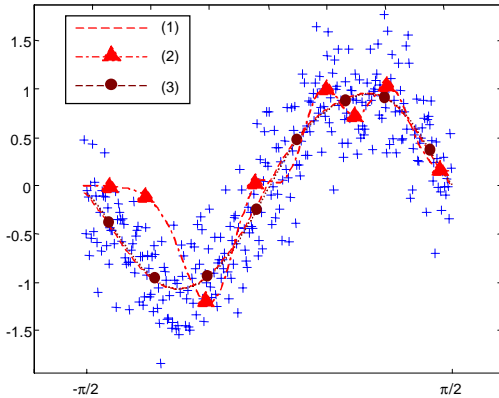


Figure 5: Regression problem (REG), $m_1 = 8$. Crosses: training data; (1): ICS; (2) random initialization of centers; (3) k -means selection of centers.

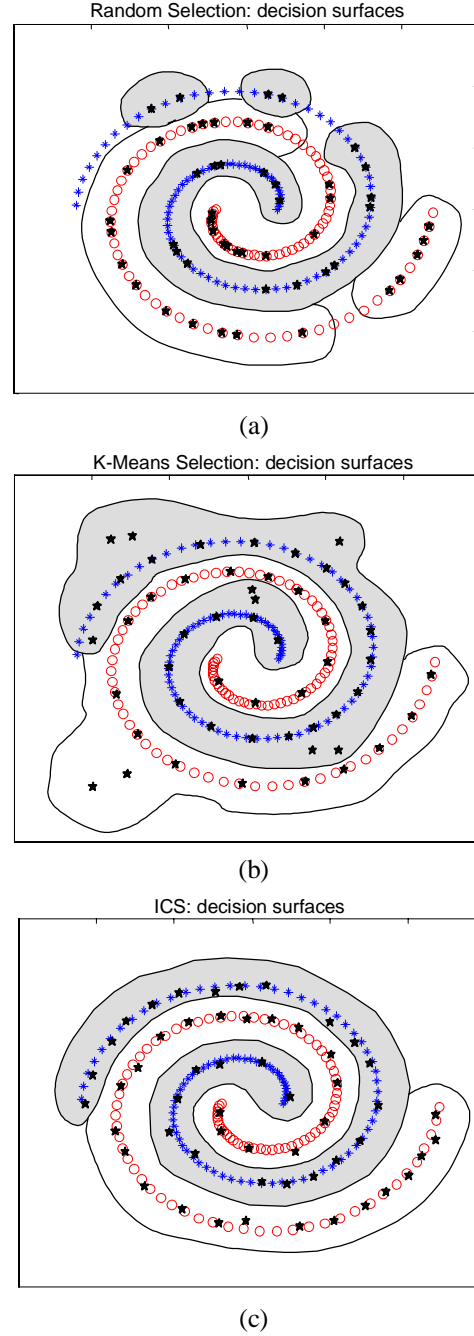


Figure 6: Center positions and decision surfaces for the SPIR problem and the three algorithms tested, $m_1 = 54$. (a) Random initialization. (b) k -means initialization. (c) ICS.

The same amount, $m_1 = 8$, of RBF centers was adopted by the random and k -means methods of center initialization for the REG problem. In Figure 5 we can notice that the results obtained by the ICS and k -means strategies are practically the same, with the difference that the ICS algorithm can automatically determine the number of RBF centers.

Figure 6 depicts the center positions and decision surfaces determined by the three algorithms applied to the SPIR problem, that is composed of 2 non-linearly separable spirals with 95 samples each. The ICS

approach produced a network with 54 centers (this number of centers was then applied to the other methods), corresponding to a compression rate of 71.58%. The random initialization sometimes results in wrong decision surfaces (Figure 6(a)), while the ICS method always lead to a correct classification for the parameters adopted. The ICS and k -means methods were able to classify the data with an error rate of 0.0%, while the random initialization presented an error rate of 24.7%. Note that the k -means clustering algorithm may position centers in regions of no data (Figure 6(b)). The training parameters for the ICS learning algorithm were: $\sigma_s = 0.5$, $\sigma_d = 0.01$, $n = 4$, $\xi = 20\%$ and $SC = 5$, for both problems.

6. Concluding remarks

This paper presented the development and evaluation of a new paradigm to define the number and position of radial basis function neural network centers. The proposed learning scheme makes use of an unsupervised learning approach for the creation of the prototype vectors, based only on the input data set.

An immunologically inspired technique for data compression is introduced. The strategy is plastic in nature, i.e., automatically determining the number of prototype vectors, and positioning them into locations of the input space which are crucial to the implementation of the input-output mapping. This strategy allows the representation of the input space with different resolution levels by distributing the prototypes according to the density distribution of the data set in the input space.

The performance of the proposed technique was compared to that of the random and k -means center selection procedures. Experiments demonstrated that a random initialization of centers might lead to misclassification and biased approximation. The k -means unsupervised selection can waste network resources by creating prototypes in insignificant regions of the input space while ignoring regions that are important for the input-output mapping [12].

The method presented has the advantage that it allows the construction of a reduced set of radial basis function centers, satisfying the Micchelli's condition for the application of the simplest training algorithm for RBF: the pseudo-inverse method.

Similar to most unsupervised learning approaches, the main drawback of the proposed strategy is the existence of user-defined parameters.

Acknowledgments

Leandro Nunes de Castro would like to thank FAPESP (Proc. n. 98/11333-9) and Fernando Von Zuben would like to thank FAPESP (Proc. n. 98/09939-6) and CNPq (Proc. n. 300910/96-7) for their financial support.

References

- [1] R. P. Lippmann, "Pattern Classification Using Neural Networks", *IEEE Communications Magazine*, November, 47-63, 1989.
- [2] J. Moody and C. Darken, "Fast Learning in Networks of Locally-Tuned Processing Units", *Neural Computation*, 1:281-294, 1989.
- [3] N. B. Karayiannis and G. W. Mi, "Growing Radial Basis Neural Networks: Merging Supervised and Unsupervised Learning with Network Growth Techniques", *IEEE Trans. on Neural Networks*, 8(6):1492-1506, 1997.
- [4] D. Dasgupta and S. Forrest, "Artificial Immune Systems in Industrial Applications", *Proc. of the IPMM'99*, 1999.
- [5] S. Forrest, A. S. Perelson, L. Allen and R. Cherkuri, "Self-Nonself Discrimination in a Computer", *Proc. of the IEEE Symposium on Research in Security and Privacy*, pp. 202-212, 1994.
- [6] J. O. Kephart, "A Biologically Inspired Immune System for Computers", In (Eds.) R. A. Brooks & P. Maes, *Artificial Life IV Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*, MIT Press, pp. 130-139, 1994.
- [7] L. N. de Castro and F. J. Von Zuben, "The Clonal Selection Algorithm with Engineering Applications", *GECCO'00 - Workshop Proceeding*, pp. 36-37, 2000a.
- [8] P. Hajela, & J. S. Yoo, "Immune Network Modelling in Design Optimization", In *New Ideas in Optimization*, (Eds.) D. Corne, M. Dorigo & F. Glover, McGraw Hill, London, pp. 203-215, 1999.
- [9] A. Ishiguro, Y. Watanabe and T. Kondo, "A Robot with a Decentralized Consensus-Making Mechanism Based on the Immune System", In *Proc. ISADS'97*, pp. 231-237, 1997.
- [10] L. N. de Castro and F. J. Von Zuben, "An Evolutionary Immune Network for Data Clustering", *Proc. of the IEEE Brazilian Symposium on Neural Networks*, pp. 84-89, 2000b.
- [11] D. S. Broomhead and D. Lowe, "Multivariable Functional Interpolation and Adaptive Networks", *Complex Systems*, 2:321-355, 1988.
- [12] S. Haykin *Neural Networks - A Comprehensive Foundation*, Prentice Hall, 2nd Ed., 1999.
- [13] M. J. D. Powell, "Radial Basis Functions for Multivariable Interpolation: A Review", in *IMA Conf. Algorithms for the Appr. of Functions and Data*, J. C. Mason & M. G. Cox (eds.), Oxford, U.K.: Oxford Univ. Press, 143-167, 1987.
- [14] C. A. Micchelli, "Interpolation of Scattered Data: Distance Matrices and Conditionally Positive Definite Functions", *Const. Approx.*, 2:11-22, 1986.
- [15] T. Poggio and F. Girosi, "Networks for Approximation and Learning", *Proceedings of the IEEE*, 78(9):1481-1497, 1990.
- [16] C. A. Janeway Jr. and P. Travers, *Immunobiology The Immune System in Health and Disease*, Garland Publishing Inc., N.Y., 2nd ed., 1994.
- [17] L. N. de Castro, L. N. and F. J. Von Zuben, "Artificial Immune Systems: Part I - Basic Theory and Applications", *Technical Report - RT DCA 01/99*, 95 p., URL: <http://www.dca.fee.unicamp.br/~lnunes>, 1999.
- [18] N. K. Jerne, "Towards a Network Theory of the Immune System", *Ann. Immunol. (Inst. Pasteur)* 125C, pp. 373-389, 1974.