Neurocomputing 75 (2012) 123–134

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Parameter estimation of q-Gaussian Radial Basis Functions Neural Networks with a Hybrid Algorithm for binary classification ☆

Francisco Fernández-Navarro [a,*], César Hervás-Martínez [a], Pedro A. Gutiérrez [a], Jose M. Peña-Barragán [b], Francisca López-Granados [b]

[a] Department of Computer Science and Numerical Analysis, University of Córdoba, Campus de Rabanales, Albert Einstein building, 3rd floor, 14074 - Córdoba, Spain
[b] Institute for Sustainable Agriculture, CSIC, P.O. Box 4084, 14080 Córdoba, Spain

ARTICLE INFO

Available online 10 August 2011

Keywords:
q-Gaussian Radial Basis Functions Neural Networks
Hybrid Algorithms
Classification
Remote sensing
Weed discrimination and control

ABSTRACT

A classification problem is a decision-making task that many researchers have studied. A number of techniques have been proposed to perform binary classification. Neural networks are one of the artificial intelligence techniques that has had the most successful results when applied to this problem. Our proposal is the use of q-Gaussian Radial Basis Function Neural Networks (q-Gaussian RBFNNs). This basis function includes a supplementary degree of freedom in order to adapt the model to the distribution of data. A Hybrid Algorithm (HA) is used to search for a suitable architecture for the q-Gaussian RBFNN. The use of this type of more flexible kernel could greatly improve the discriminative power of RBFNNs. In order to test performance, the RBFNN with the q-Gaussian basis functions is compared to RBFNNs with Gaussian, Cauchy and Inverse Multiquadratic RBFs, and to other recent neural networks approaches. An experimental study is presented on 11 binary-classification datasets taken from the UCI repository. Moreover, aerial imagery taken in mid-May, mid-June and mid-July was used to evaluate the potential of the methodology proposed for discriminating *Ridolfia segetum* patches (one of the most dominant and harmful weeds in sunflower crops) in two naturally infested fields in southern Spain.

1. Introduction

Radial Basis Function Neural Networks (RBFNNs) have received an extensive attention in the literature, and their applications vary from face recognition [1] to time series predictions [2] or signal processing [3], and as they are the object of continuous research, there is also abundant literature on extensions and improvements in RBFNN learning and modeling [4–6].

Different basis functions like Standard Gaussian (SRBF), thin-plate Spline (TPSRBF) [7], Multiquadratic (MRBF) [8], Inverse Multiquadratic (IMRBF) [8], and Cauchy (CRBF) [9] have been proposed for hidden layer nodes. Even so, the standard Gaussian function is the one most commonly selected (SRBF). Compared to other types of Artificial Neural Networks (ANNs), the RBFNNs require less computational time for learning and also have a more compact topology [10]. RBFs have been applied in the area of ANNs, and are used as a replacement for the sigmoidal unit in Multilayer Perceptron Neural Networks (MLPNNs) [11] or for multiplicative units in Product Unit Neural Networks (PUNNs) [12–14].

This paper presents a novel RBF based on q-Gaussian distribution which parametrizes standard Gaussian distribution by replacing exponential expressions with q-exponential expressions [15], while maximizing Tsallis entropy [16] under certain constraints [17,18]. This novel basis function incorporates a real parameter q (besides the centers and width of the RBF) which can relax or contract the shape of the kernel. It matches the shape of the kernel and the distribution of the distances better, since the modification of the q parameter allows the representation of different basis functions (CRBF, SRBF, IMRBF, …). This basis function has been recently analysed in predictive microbiology [19].

On the other hand, the neural network learning scheme refers to the procedure in which the algorithm estimates neural network parameters from a set of previously labeled examples. Depending on how examples are provided, learning can be performed in an incremental mode (online/sequential method), or in a batch mode (offline method). In practical classification applications, even though there is initially a large data set, more new training data becomes available from time to time and will arrive sequentially. To handle the training of the network for the new data, batch learning schemes require the network to be retrained all over again, resulting in a long

☆ This paper is a very significant extension of a contribution that appeared in the 5th International Conference on Hybrid Artificial Intelligence Systems (HAIS 2010, San Sebastian, Spain).

* Corresponding author. Tel.: +34 957 21 83 49; fax: +34 957 21 83 60.
E-mail address: i22fenaf@uco.es (F. Fernández-Navarro).

0925-2312/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2011.03.056

training time. A learning scheme would be quite useful in real applications to handle this sequential data without retraining all the data all over again [20–22].

A new learning algorithm called the Extreme Learning Machine (ELM) has recently been proposed for single hidden layer feed forward neural networks [23,24]. This novel procedure, unlike conventional implementations of gradient-based learning algorithms, chooses randomly hidden nodes and analytically determines the output weights of the network [25]. This algorithm provides good generalization performances at extremely fast learning speeds and in theory the universal approximator property has proved to hold true [26]. However, ELM may need a higher number of hidden nodes due to the random determination of input weights and hidden biases. Several algorithms based on the ELM method (hybrid proposals which use the differential evolution algorithms [27] and a convex optimization method [28,29]) have been developed to achieve good generalization performances with more compact networks. Originally, the ELM algorithm worked with a batch learning mode, although recently an online version has been proposed [30].

Classical neural network training assumes a fixed architecture that is difficult to establish beforehand. Evolutionary algorithms (EAs), which are stochastic search algorithms that execute a global search in the input space preventing a fall to local optimum [31,32], have demonstrated great accuracy in designing a nearly optimal architecture. This fact, together with the complexity of the error surface associated with an ANN, justifies the use of an EA to design the structure and to adjust the weight of these models [4,14,33]. For this reason, a Hybrid Algorithm (HA) is employed in this study to estimate the parameters of the RBFs: the number of hidden nodes, the centers, the width and the value of the q parameter of q-Gaussian RBFNNs.

In this way, the procedure is hybridized by two artificial intelligence paradigms: Neural Networks and Evolutionary Computation. The Hybrid Artificial Intelligent System (HAIS) denotes a software system which employs a combination of methods and techniques from artificial intelligence subfields [34,35]. Therefore, the proposal of this paper can be considered a new alternative approach within the scope of HAIS.

In order to analyze the performance and robustness of the methodology proposed, it is applied to both 11 datasets taken from the UCI repository and also to an interesting agronomical problem that involves discriminating *Ridolfia segetum* patches in sunflower fields, using multispectral imagery. *R. segetum* is a very frequent annual, umbelliferous weed that is abundant in clay soils in Andalusia (Spain). Its life cycle coincides with that of the sunflower, which enhances its competitiveness. It results in an average crop yield reduction of about 32% when there are two *R. segetum* plants per m^{-2}. In order to reduce herbicide applications by applying Site-Specific Weed Management (SSWM) strategies in the fields infested with *R. segetum*, approaches based on the combination of Evolutionary Product-Unit Neural Networks (EPUNNs) and Logistic Regression (LR) were previously applied to remotely sensed images [36,37].

The results are compared to other RBFNNs obtained with the same HA (SRBF, CRBF and IMRBF), to a sequential ELM classifier (Online Sequential ELM for RBF (OS-ELM-RBF) [30]), and to batch neural network classifiers (original ELM using Sigmoidal and RBF (ELM-SIG and ELM-RBF) and Evolutionary ELM for RBF (EELM-RBF) [27]). The q-Gaussian method is found to obtain better results than SRBF in almost all the datasets considered. A measure of statistical significance is used to determine differences in mean ranking, which indicates that q-Gaussian reaches the state-of-the-art.

This paper is organized as follows: a brief analysis of RBFNN models and their application to binary-classification problems is given in Section 2; a methodology to estimate the RBF parameters based on HAs is presented in Section 3; Section 4 explains the experiments carried out; and finally, Section 5 summarizes the conclusions of our work.

2. Probabilistic radial basis function neural networks

RBFNNs [38] are well-suited for function approximation and pattern recognition due to their simple topological structure and ability to reveal how learning proceeds in an explicit manner. A RBF is a function which has been built into a distance criterion with respect to a center. Let the number of nodes in the input and hidden layer be p and m, respectively. For any sample $\mathbf{x}=(x_1,x_2,\ldots,x_p)$, the output of the RBFNN is $f_{\text{RBFNN}}(\mathbf{x})$. The model of a RBFNN can be described using the following equation:

$$f_{\text{RBFNN}}(\mathbf{x})=\beta_0+\sum_{i=1}^{m}\beta_i\cdot\phi_i(d_i(\mathbf{x})), \tag{1}$$

where $\phi_i(d_i(\mathbf{x}))$ is non-linear mapping from the input layer to the hidden layer, $\boldsymbol{\beta}=(\beta_0,\beta_1,\beta_2,\ldots,\beta_m)$ is the vector including the weights of the connections between the hidden layer and the output layer, and β_0 is the bias. The function $d_i(\mathbf{x})$ can be defined as:

$$d_i(\mathbf{x})=\frac{\|\mathbf{x}-\mathbf{c}_i\|^2}{\theta_i^2}, \tag{2}$$

where θ_i is the scalar parameter that defines the width for the i-th radial unit, $\|\cdot\|$ represents the Euclidean norm and $\mathbf{c}_i=[c_1,c_2,\ldots,c_p]$ is the i-th center of the RBFNN. The SRBF is the Gaussian function, which is given by:

$$\phi_{\text{SRBF}_i}(d_i(\mathbf{x}))=e^{-d_i(\mathbf{x})}. \tag{3}$$

The radial basis function $\phi_i(d_i(\mathbf{x}))$ can take different forms, including the Cauchy RBF (CRBF) defined by:

$$\phi_{\text{CRBF}_i}(d_i(\mathbf{x}))=\frac{1}{1+d_i(\mathbf{x})}, \tag{4}$$

and the Inverse Multiquadratic RBF (IMRBF), given by:

$$\phi_{\text{IMRBF}_i}(d_i(\mathbf{x}))=\frac{1}{(1+d_i(\mathbf{x}))^{1/2}}. \tag{5}$$

Fig. 1a illustrates the influence of the choice of the RBF in hidden unit activation. One can observe that the Gaussian function presents higher activation close to the radial unit center than the other two RBFs. The CRBF has been successfully applied to image retrieval [39] and Computerized Tomography [40], whereas the IMRBF has been used in applications related to real-time signal-processing [8], among other scientific and engineering applications.

This paper analyses the use of the q-Gaussian function as the RBF. Based on the idea of q-Gaussian distribution [16,17], the q-Gaussian RBF is defined by transforming the exponential expression of the standard RBF to a q-exponential expression. The q-exponential function, which reduces the exponential function in the $q\to1$ limit, is defined as follows:

$$e_q^x\equiv(1+(1-q)x)^{1/(1-q)}=\frac{1}{(1-(q-1))^{1/(q-1)}}. \tag{6}$$

Therefore, the q-Gaussian RBF can be defined as:

$$\phi_{\text{q-GaussianRBF}_i}(d_i(\mathbf{x}))=(1-(1-q)d_i(\mathbf{x}))^{1/(1-q)}, \tag{7}$$

if $(1-(1-q)d_i(\mathbf{x}))\geq0$, and 0, otherwise. The q-Gaussian RBF can reproduce different RBFs for different values of the real q parameter. As an example, when the parameter q is close to 2, the q-Gaussian is the CRBF, for $q=3$, the activation of a radial unit with an IMRBF for $d_i(\mathbf{x})$ turns out to be equal to the activation of a radial unit with a q-Gaussian RBF for $d_i(\mathbf{x})/2$ and, finally, when the value of q converges to 1, the q-Gaussian converges to the

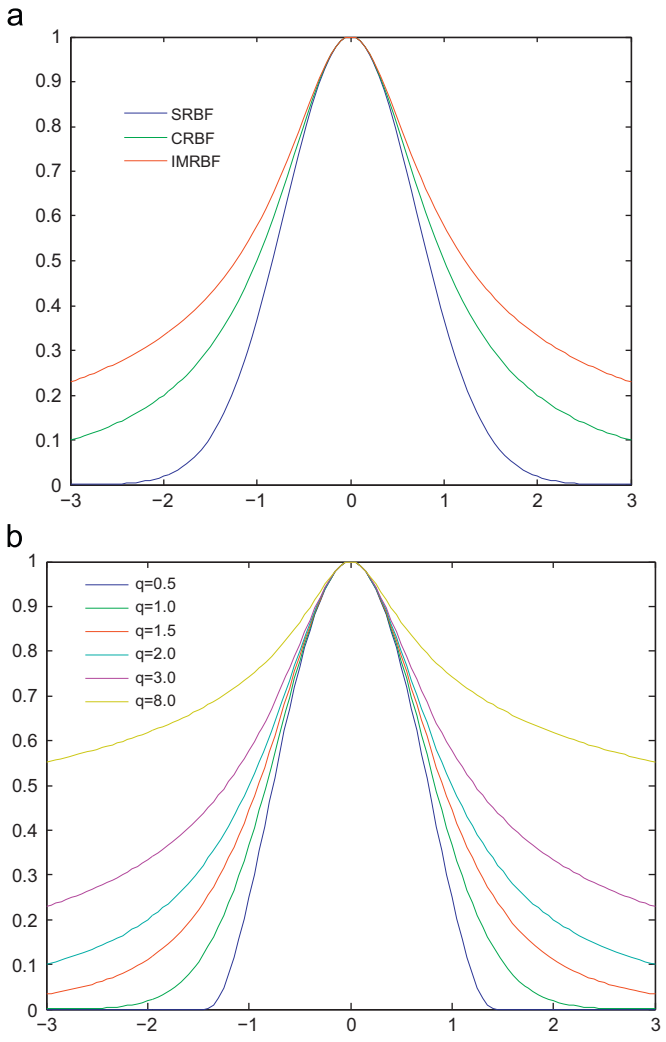


Fig. 1. Radial unit activation in one-dimensional space with $c=0$ and $\theta=1$ for different RBFs: (a) SBRF, CRBF and IMRBF and (b) q-Gaussian with different values of q.

Gaussian function (SRBF). Fig. 1b presents the radial unit activation for the q-Gaussian RBF for different values of q. As can be seen in Fig. 1b, a slight increase in the q value results in a smooth modification in the shape of the RBF.

In this work, the outputs of the neural networks are interpreted from the point of view of probability through the use of the softmax activation function. The softmax activation function is given by:

$$g(\mathbf{x}) = \frac{e^{f_{\mathrm{RBFNN}}(\mathbf{x})}}{1+e^{f_{\mathrm{RBFNN}}(\mathbf{x})}}, \tag{8}$$

where $g(\mathbf{x})$ is the probability that a pattern $\mathbf{x}$ has of belonging to the class of interest. The probability a pattern $\mathbf{x}$ has of not belonging to the class is $1-g(\mathbf{x})$.

The error surface associated with the model is very convoluted. Thus, the parameters of the RBFNNs are estimated by means of a HA (detailed in Section 3). The HA was developed to optimize the error function given by the negative log-likelihood for N observations, which is defined for a classifier g:

$$l(g) = \frac{1}{N}\sum_{n=1}^{N}[-y_n f_{\mathrm{RBFNN}}(\mathbf{x}_n) + \log \exp f_{\mathrm{RBFNN}}(\mathbf{x}_n)], \tag{9}$$

where y_n is equal to 1 if the pattern $\mathbf{x}_n$ belongs to the first class and equal to 0 otherwise, and f_{RBFNN} is defined in Eq. (1).

Finally, a RBFNN that allows linear combinations of SRBF, CRBF and IMRBF in the same RBFNN could have been considered, because the use of linear combinations of different kinds of kernel functions is commonplace in RBFNN design. However, this option presents two drawbacks with respect to using q-Gaussian RBFNN: (1) our proposal is more flexible because infinite types of RBF can be obtained by modifying the value of the q parameter (not only well-known RBFs); (2) having a continuous parameter, which can vary the shape of the RBF, allow the mutations within the evolutionary algorithm to be less abrupt which improves the convergence of the evolutionary algorithm.

3. Hybrid Algorithm

The proposed Hybrid Algorithm (HA) is composed of two stages. In the first stage, an evolutionary algorithm (EA) is used as a global stochastic search algorithm which generates candidate RBFNNs. In the second stage, the *iRprop+* algorithm performs a local optimization procedure on the best RBFNN individual from the previous generation. For the sake of simplicity and to reduce the computational cost, the local improvement procedure (*iRprop+*) is applied only at the end of the HA, and only to the best solution. This implies that the HA renounces merge exploration (global search) with exploitation (local search), as occurs commonly in evolutionary algorithms. Fig. 2 describes the procedure to estimate the parameters of the radial units.

The basic framework of the EA is the following: the search begins with an initial population of RBFNNs and, in each iteration, the population is updated using a population-update algorithm which evolves both its structure and weights. The population is subject to operations of replication and mutation. The main characteristics of the algorithm are the following:

1. *Representation of the individuals*: The algorithm evolves architectures and connection weights simultaneously, each individual being a fully specified RBFNN. RBFNNs are represented using an object-oriented approach and the algorithm deals directly with the RBFNN phenotype. Each connection is specified by a binary value indicating if the connection exists, and a real value representing its weights.
2. *Error and fitness functions*: $l(g)$ (see Eq. (9)) is considered as the error function of an individual g in the population. The fitness

1: **Hybrid Algorithm**:
2: Generate a random population of size N
3: **repeat**
4: Calculate the fitness of every individual in the population
5: Rank the individuals with respect to their fitness
6: The best individual is copied into the new population
7: The best 10% of population individuals are replicated and they substitute the worst 10% of individuals
8: Apply parametric mutation to the best (p_m)% of individuals
9: Apply structural mutation to the remaining $(100-p_m)$% of individuals
10: **until** the stopping criterion is fulfilled
11: Apply *iRprop+* to the best solution obtained by the EA in the last generation.

Fig. 2. Hybrid Algorithm (HA) framework.

measure needed to evaluate the individuals is a strictly decreasing transformation of the error function $l(g)$ given by $A(g)=1/(1+l(g))$, where $0<A(g)\leq 1$.

3. *Initialization of the population*: The initial population is generated trying to obtain RBFNNs with the maximum possible fitness. First, 5000 random RBFNNs are generated. The centers of the radial units are first defined by the k-means algorithm. k-means are run for all the different values of k, where $k\in[M_{min},M_{max}]$. M_{min} and M_{max} are parameters of the HA defined as the minimum and maximum number of hidden nodes allowed for any RBFNN model. We assign a random value $m\in[M_{min},M_{max}]$ per RBFNN, and consider the centers obtained through the corresponding k-means. The widths of the RBFNNs are initialized at the geometric mean of the distance to the two nearest neighbors and the q parameter at values in the interval [0.75,1.25], since when $q\rightarrow 1$, the q-Gaussian reduces the SRBF. A random value in the $[-I,I]$ interval is assigned for the weights between the hidden layer and the output layer. The individuals obtained are evaluated using the fitness function, and the initial population is finally obtained by selecting the best 500 RBFNNs.
4. *Parametric and structural mutations*: Parametric mutation consists of a simulated annealing algorithm [12]. Structural mutation implies a modification in the structure of the RBFNNs and allows the exploration of different regions in the search space, helping to maintain the diversity of the population. There are four different structural mutations: hidden node addition, hidden node deletion, connection addition and connection deletion. These four mutations are applied to each network sequentially. More information about the genetic operators proposed can be seen in [41,42]. It is important to highlight how the structural and parametric mutations of q are performed:
   - Structural mutation: If the structural mutator adds a new node in the RBFNN, the q parameter is assigned a γ value, where $\gamma\in[0.75,1.25]$, since when $q\rightarrow 1$, the q-Gaussian reproduces the SRBF.
   - Parametric mutation: The q parameter is updated by adding a ε value, where $\varepsilon\in[-0.25,0.25]$, since the modification of the q-Gaussian is very sensitive to the q variation (Fig. 1b).
5. *iRprop+ local optimizer*: The local optimization algorithm used in our paper is the *iRprop+* [43] optimization method. In the proposed methodology, after the EA is run, then the local optimization algorithm is applied to the best solution obtained by the EA in the previous generation. We have adapted the *iRprop+* local improvement procedure to the softmax activation function (8) and the cross-entropy error function (9). In this case, the gradient vector is given by the following equation:

$$\nabla l(\boldsymbol{\beta},\mathbf{c},\boldsymbol{r},\mathbf{q})=\left(\frac{\partial l}{\partial \boldsymbol{\beta}},\frac{\partial l}{\partial \mathbf{c}},\frac{\partial l}{\partial \boldsymbol{r}},\frac{\partial l}{\partial \boldsymbol{q}}\right). \tag{10}$$

Let η be any of the parameters of $\boldsymbol{\beta},\mathbf{c},\boldsymbol{r}$ and $\mathbf{q}$ where $\boldsymbol{\beta}$ is the vector with the connection weights of hidden to output node, $\boldsymbol{c}$ is the vector with the centers values of the q-Gaussian RBFs, $\boldsymbol{r}$ includes the values of the width of each q-Gaussian and $\boldsymbol{q}$ stores the q values of each q-Gaussian RBF, being therefore:

$$\frac{\partial l}{\partial \eta}=\frac{1}{N}\sum_{n=1}^{N} y_n \frac{1}{g(\mathbf{x}_n,\boldsymbol{\theta})}\frac{\partial g(\mathbf{x}_n,\boldsymbol{\theta})}{\partial \eta},$$

$$\frac{\partial g(\mathbf{x},\boldsymbol{\theta})}{\partial \eta}=(g-g^2)\frac{\partial f}{\partial \eta},$$

where $g(\mathbf{x}_n,\boldsymbol{\theta})=g$ and $f(\mathbf{x}_n,\boldsymbol{\theta})=f$. Finally, we have the following expressions for the output layer:

$$\frac{\partial f}{\partial \beta_0}=1;\quad \frac{\partial f}{\partial \beta_s}=\phi_s(d_s(\mathbf{x}))$$

and for the hidden layer:

$$\frac{\partial f}{\partial c_{st}}=\beta_s\frac{2\phi_s(d_s(\mathbf{x}))^{q_s}(x_{st}-c_{st})}{r_s^2},$$

$$\frac{\partial f}{\partial r_s}=\beta_s\frac{2\phi_s(d_s(\mathbf{x}))^{q_s}(1-\phi_s(d_s(\mathbf{x}))^{(1-q_s)}}{r_s(1-q_s)},$$

$$\frac{\partial f}{\partial q_s}=\frac{\beta_s}{(1-q_s)^2}(\phi_s(d_s(\mathbf{x}))^{q_s}(1-\phi_s(d_s(\mathbf{x}))^{(1-q_s)}) +\phi_s(d_s(\mathbf{x}))\ln(\phi_s(d_s(\mathbf{x}))^{(1-q_s)})),$$

where $\phi_s(d_s(\mathbf{x}))$ represents the output of the s-th q-Gaussian RBF, $s=1,2,\ldots,m$ and $t=1,2,\ldots,p$. In the case of the remaining basis function considered (SRBF, CRBF and IMRBF) the gradient vector is given by the following equation:

$$\nabla l(\boldsymbol{\beta},\mathbf{c},\boldsymbol{r})=\left(\frac{\partial l}{\partial \boldsymbol{\beta}},\frac{\partial l}{\partial \mathbf{c}},\frac{\partial l}{\partial \boldsymbol{r}}\right). \tag{11}$$

Taking into account that the only difference from other neural networks considered with respect to the q-Gaussian RBFNN is the basis function employed in hidden layer, the only derivatives that differ with respect to those shown in the case of the q-Gaussian, are the derivatives of the output function with respect to the $\boldsymbol{c}$ and $\boldsymbol{r}$ parameters. In the case of SRBF, we have the following expressions for the hidden layer:

$$\frac{\partial f}{\partial c_{st}}=\beta_s\phi_s(d_s(\mathbf{x}))\frac{(x_t-c_{st})}{r_s^2},$$

$$\frac{\partial f}{\partial r_s}=\beta_s\phi_s(d_s(\mathbf{x}))\frac{\sum_{j=1}^{p}(x_j-c_{sj})^2}{r_s^3},$$

where $\phi_s(d_s(\mathbf{x}))$ represents the output of the s-th SRBF. For the IMRBF, we obtain the following equations:

$$\frac{\partial f}{\partial c_{st}}=\beta_s\frac{\phi_s(d_s(\mathbf{x}))^3(x_t-c_{st})}{r_s^2},$$

$$\frac{\partial f}{\partial r_s}=\beta_s\frac{\phi_s(d_s(\mathbf{x}))(1-\phi_s(d_s(\mathbf{x}))^2)}{r_s},$$

where $\phi_s(d_s(\mathbf{x}))$ represents the output of the s-th IMRBF. Finally, the derivatives of the hidden layer for the CRBF are obtained as:

$$\frac{\partial f}{\partial c_{st}}=\beta_s\frac{2\phi_s(d_s(\mathbf{x}))^2(x_t-c_{st})}{r_s^2},$$

$$\frac{\partial f}{\partial r_s}=\beta_s\frac{2(1-\phi_s(d_s(\mathbf{x}))^2)}{r_s},$$

where $\phi_s(d_s(\mathbf{x}))$ represents the output of the s-th CRBF.

4. Experiments

4.1. Description of the datasets and the experimental design

The methodology proposed is applied to 11 datasets taken from the UCI repository [44]. Two additional datasets have been included that correspond to remote-sensing agricultural problems concerning weed patch determination in sunflower crops, described in Section 4.2.

Table 1 summarizes the properties of the selected datasets. It shows, for each dataset, the number of instances (Size), number of Real (R), Binary (B) and Nominal (N) input variables, total number of inputs (#In.), and per-class distribution of the instances (Distribution). There is a simple linear rescaling of the input variables in the interval $[-2,2]$, X_i^* being the transformed variables.

Table 1
Summary description of the datasets.

Dataset	Size	*R*	*B*	*N*	#In.	Distribution
Labor	57	8	3	5	29	(30, 27)
Promoters	106	–	–	57	114	(53, 53)
Hepatitis	155	6	13	–	19	(32, 123)
Sonar	208	60	–	–	60	(98, 110)
Heart	270	13	–	–	13	(150, 120)
Breast-C	286	4	3	2	15	(201, 85)
Heart-C	302	6	3	4	26	(164, 138)
Liver	345	6	–	–	6	(145, 200)
Vote	435	–	16	–	16	(267, 168)
Card	690	6	4	5	51	(307, 308)
German	1000	6	3	11	61	(700, 300)
Mtb-May	1600	4	–	–	4	(800, 800)
Mtb-June	1600	4	–	–	4	(800, 800)
Mtb-July	1600	4	–	–	4	(800, 800)
SC-May	1600	4	–	–	4	(800, 800)
SC-June	1600	4	–	–	4	(800, 800)
SC-July	1600	4	–	–	4	(800, 800)

All nominal variables are transformed to binary variables. Breast-C, Breast-Cancer; Heart-C, Heart-disease; Mtb, Matabueyes; SC, Santa Cruz.

The proposed model (q-Gaussian RBFNN) is compared to:

- Other RBFNNs obtained applying the same HA (detailed in Section 3):
  - The Standard Radial Basis Function (SRBF), where the transfer function is Gaussian.
  - The Cauchy Radial Basis Function (CRBF).
  - The Inverse Multiquadratic Radial Basis Function (IMRBF).
- Sequential Neural Network Classifiers:
  - Online Sequential Extreme Learning Machine for Radial Basis Function (OS-ELM-RBF) [30]. This algorithm is an online sequential learning algorithm for Single Layer Feed-forward Networks (SLFNs) with RBF hidden nodes. In OS-ELM-RBF, the parameters of hidden nodes (the centers and radius of RBF nodes) are randomly selected and the output weights are analytically determined based on data that arrives sequentially.
- Batch Neural Networks Classifiers:
  - Original Extreme Learning Machine (ELM) [23]. We have employed two different basis functions for the ELM algorithm: Sigmoidal (ELM-SIG) and RBF (ELM-RBF). In the ELM-RBF algorithm, the centers have been taken randomly from the data points and the widths randomly drawn between percentile 20 and percentile 80 of the distance distribution in the input space, as suggested in [45].
  - The Evolutionary Extreme Learning Machine for Radial Basis Function (EELM-RBF) [27] improves the original ELM by using a Differential Evolution (DE) algorithm. DE was proposed by Storn and Price [46] and it is known as one of the most efficient evolutionary algorithms. The EELM-RBF uses DE to select the input weights between input and hidden layers (centers and radius) and the Moore-Penrose generalized inverse to analytically determine the output weights between hidden and output layers.

The experimental design was conducted using 10-fold cross validation, with 10 repetitions per fold, except for the "Matabueyes" and "Santa Cruz" datasets. The experimental design followed for the Matabueyes and Santa Cruz datasets is described in Section 4.2. For these datasets, we did not perform a 10-fold cross validation in order to compare our results with those obtained in [36].

The evaluation of different models in the generalization set has been performed using three different measures: Correctly Classified Rate (C_G) or accuracy, Root Mean Squared Error ($RMSE_G$) and Area Under the ROC Curve (AUC_G) because they have been identified as three of the most commonly used metrics for determining the performance of a classifier [47]. C represents threshold metrics, while AUC is a probability metric and $RMSE$ is a rank metric.

All the parameters used in the evolutionary algorithm, except for the maximum and minimum number of RBFs in the hidden layer and the number of generations, have the same values in all the problems analysed below. For the selection of these parameters, a grid search algorithm was applied with ten-fold cross validation in the same way as for SVM, using the following ranges: $[M_{min},M_{max}] \in \{[2,5],[4,7]\}$ and $\#Gen \in \{20,40,100,400\}$. In the case of the compared methods (OS-ELM-RBF, ELM-SIG, ELM-RBF, EELM-RBF), the most critical parameter is the number of nodes in the hidden layer. In the experiments, the m value for these methods was experimentally determined by a cross validation procedure applied to the training set, using the values $\{10,20,\ldots,100\}$. With respect to EELM-RBF, in order to achieve good performance results, the population of the EELM-RBF is set at 500 individuals to obtain better diversity in the population. Similarly, the maximum number of generations is set at 100.

The connections between the hidden and output layer are initialized in the $[-5,5]$ interval (i.e. $[-I,I]=[-5,5]$). The size of the population is $N=500$. For the structural mutation, the number of nodes that can be added or removed is within the [1,2] interval, and the number of connections to add or delete in the hidden and the output layer during structural mutations is within the [1,7] interval.

4.2. Description and experimental design of the "Matabueyes" and "Santa Cruz" datasets

The original dataset was taken from [36]. In [36], the authors evaluate the potential of Evolutionary Product-Unit Neural Networks (EPUNNs), Logistic Regression (LR) and a combination of both (Logistic Regression using Initial Covariates and Product Units, LRIPU) to discriminate *R. segetum* patches in the sunflower crops in two naturally infested fields. The hybridization of LR and EPUNNs was proposed in two recent works [48,49]. This study was conducted in two fields in Andalusia, southern Spain, called Matabueyes and Santa Cruz that were naturally infested by *R. segetum*.

Conventional-color (CC) and color infrared (CIR) aerial imagery of the areas under study were taken in mid-May, mid-June and mid-July. Then, the photographs were digitalized and re-sampled to a pixel size representing 40 cm $\times$ 40 cm ground area.

Input variables included the digital values of all the bands in each available image: CC images responded to blue (B), green (G) and red (R) broad bands of the electromagnetic spectrum, and CIR images to green, red and near infra-red bands. The authors geo-referenced a total of 1600 pixels in each phenological stage, where 800 pixels corresponded to *R. segetum*, 400 pixels corresponded to the bare soil class and 400 corresponded to sunflowers. The objective is to differentiate between *R. segetum* and all other pixels, because it is not necessary to distinguish between *soil* and *sunflower*. More details can be found in [36].

The experimental design was conducted using a stratified holdout cross validation procedure, where the size of the training set was approximately $3n/4$, and that of the generalization set was $n/4$, n being the size of the full dataset. Consequently, each dataset mentioned above was randomly split into two datasets. A 1120 instance dataset was used for model training and the remaining 480 instances formed the generalization dataset.

4.3. Comparison to other radial basis functions and to other recent neural networks approaches

An analysis was performed on the performance of all the traditional RBFs (CRBF, IMRBF and SRBF) obtained by the HA and

on other recent neural networks approaches when compared to the q-Gaussian model. This was essential because the proposed model can reproduce different RBFs by changing an additional q parameter. Consequently, the q-Gaussian model had to achieve better performance than all these RBFs in order to justify the additional q parameter.

As mentioned above, the agronomical problem datasets were already evaluated with product-unit neural networks and with their hybridization with logistic regression models.

Tables 2–4 show the mean and the standard deviation of the correct classification rate, Root Mean Squared Error and Area Under the ROC Curve in the generalization set (C_G, AUC_G and $RMSE_G$) in a total of 100 executions for the datasets taken from the UCI repository and 30 executions for the remote sensing agricultural problems. The analysis of the results leads us to conclude that the q-Gaussian model obtained the best results for seven datasets using C_G as the test variable. Furthermore, the q-Gaussian model got the best results for ten datasets in AUC_G and for seven datasets in $RMSE_G$, obtaining the best mean performance and the best mean ranking of all the metrics used in the experiments.

In this case, a performance analysis of the results using a parametric statistical treatment could lead to mistaken conclusions, since a previous evaluation of the C_G, AUC_G and $RMSE_G$ value resulted in rejecting the normality and the equality of the variance hypothesis. Furthermore, DemŠar [50] suggested that

Table 2
Comparison of the proposed basis functions to other basis functions: mean and standard deviation of the generalization accuracy results ($C_G(\%)$), mean C_G over all the datasets ($\overline{C}_G(\%)$), and mean C_G ranking ($\overline{R_C}$).

	Method ($C_G(\%)$)							
	OS-ELM-RBF	ELM-SIG	ELM-RBF	EELM-RBF	SRBF	CRBF	IMRBF	q-Gaussian
Labor	89.30 ± 10.34	78.66 ± 17.99	90.00 ± 12.06	89.33 ± 12.04	91.33 ± 12.09	$\mathbf{95.00 \pm 11.24}$	91.66 ± 8.78	*93.33 ± 11.65*
Promoters	78.30 ± 15.35	61.78 ± 13.97	81.03 ± 13.16	*82.34 ± 10.53*	75.54 ± 13.56	80.18 ± 6.66	81.09 ± 8.69	$\mathbf{84.00 \pm 6.15}$
Hepatitis	83.75 ± 7.44	82.36 ± 7.19	82.97 ± 8.54	83.59 ± 7.78	$\mathbf{86.33 \pm 8.09}$	83.16 ± 7.15	85.12 ± 7.52	*85.30 ± 7.54*
Sonar	73.20 ± 3.87	69.95 ± 9.75	75.47 ± 4.93	*77.30 ± 6.42*	$\mathbf{78.38 \pm 9.03}$	74.09 ± 10.20	76.02 ± 11.16	76.04 ± 13.56
Heart	81.80 ± 6.09	81.95 ± 5.09	82.59 ± 4.67	82.93 ± 6.70	81.85 ± 8.97	83.70 ± 8.76	$\mathbf{84.81 \pm 8.45}$	*84.07 ± 7.20*
Breast-C	72.09 ± 6.55	70.89 ± 7.78	71.51 ± 7.38	72.34 ± 5.31	72.04 ± 6.39	71.35 ± 8.00	$\mathbf{73.10 \pm 6.39}$	*73.06 ± 6.77*
Heart-C	84.98 ± 4.70	83.21 ± 4.68	85.10 ± 4.21	84.31 ± 3.50	85.44 ± 3.83	85.45 ± 5.59	*85.77 ± 3.05*	$\mathbf{85.79 \pm 5.20}$
Liver	$\mathbf{71.40 \pm 6.69}$	70.06 ± 9.40	69.29 ± 7.91	69.08 ± 7.12	68.41 ± 5.15	65.23 ± 8.23	65.52 ± 6.31	*71.30 ± 6.50*
Vote	96.02 ± 2.17	95.78 ± 2.40	95.98 ± 2.52	$\mathbf{96.42 \pm 3.53}$	*96.32 ± 3.97*	95.39 ± 3.59	94.94 ± 2.36	96.08 ± 3.45
Card	85.95 ± 3.77	85.94 ± 3.99	86.28 ± 4.02	*86.58 ± 3.89*	86.08 ± 3.14	86.52 ± 3.55	85.94 ± 3.80	$\mathbf{87.87 \pm 0.37}$
German	$\mathbf{75.46 \pm 3.90}$	73.20 ± 3.77	74.93 ± 3.46	74.32 ± 3.90	74.80 ± 3.82	74.90 ± 3.17	74.40 ± 2.50	*75.25 ± 2.98*
Mtb-May	63.89 ± 2.13	61.30 ± 1.89	64.32 ± 1.87	66.89 ± 2.40	*69.84 ± 0.98*	69.78 ± 1.10	68.93 ± 1.12	$\mathbf{69.95 \pm 1.56}$
Mtb-June	96.59 ± 0.39	95.70 ± 0.12	97.12 ± 0.57	97.30 ± 0.66	$\mathbf{99.27 \pm 0.10}$	98.45 ± 0.27	98.90 ± 0.30	*99.15 ± 0.05*
Mtb-July	73.34 ± 2.32	*75.71 ± 3.49*	75.34 ± 2.12	75.42 ± 3.12	75.24 ± 1.97	73.59 ± 1.85	73.21 ± 1.58	$\mathbf{76.63 \pm 2.14}$
SC-May	74.32 ± 1.63	76.12 ± 1.70	76.24 ± 1.19	75.45 ± 1.88	77.85 ± 1.30	*77.92 ± 1.40*	77.14 ± 1.39	$\mathbf{78.25 \pm 1.17}$
SC-June	94.98 ± 1.42	95.10 ± 2.12	95.41 ± 1.89	95.08 ± 1.92	*97.70 ± 0.26*	97.40 ± 0.22	97.26 ± 0.29	$\mathbf{97.87 \pm 0.31}$
SC-July	81.32 ± 1.23	80.12 ± 2.12	79.98 ± 2.58	80.40 ± 1.98	*83.53 ± 0.55*	83.38 ± 0.62	$\mathbf{83.57 \pm 0.60}$	83.47 ± 0.73
$\overline{C}_G(\%)$	80.98	78.69	81.38	81.71	*82.35*	82.08	82.19	**83.37**
$\overline{R_C}$	5.64	6.79	5.11	4.52	*3.64*	4.41	4.08	**1.76**

The best result is in bold face and the second best result in italics.

Table 3
Comparison of the proposed basis functions to other basis functions: mean and standard deviation of the area under the ROC curve results over the generalization set (AUC_G), mean AUC_G over all the datasets ($\overline{AUC}_G$), and mean AUC_G ranking ($\overline{R_{AUC}}$).

	Method (AUC_G)							
	OS-ELM-RBF	ELM-SIG	ELM-RBF	EELM-RBF	SRBF	CRBF	IMRBF	q-Gaussian
Labor	0.90 ± 0.10	0.83 ± 0.09	0.91 ± 0.09	0.90 ± 0.09	0.92 ± 0.10	*0.95 ± 0.09*	0.91 ± 0.07	$\mathbf{0.97 \pm 0.10}$
Promoters	0.84 ± 0.17	0.70 ± 0.17	0.86 ± 0.12	*0.87 ± 0.11*	0.83 ± 0.15	0.86 ± 0.14	0.85 ± 0.08	$\mathbf{0.90 \pm 0.09}$
Hepatitis	0.80 ± 0.07	0.81 ± 0.06	0.80 ± 0.07	0.81 ± 0.07	$\mathbf{0.86 \pm 0.10}$	0.81 ± 0.08	*0.84 ± 0.08*	0.82 ± 0.11
Sonar	0.78 ± 0.09	0.75 ± 0.12	0.85 ± 0.10	$\mathbf{0.88 \pm 0.10}$	$\mathbf{0.88 \pm 0.12}$	0.83 ± 0.09	0.82 ± 0.05	*0.86 ± 0.08*
Heart	0.82 ± 0.07	0.83 ± 0.04	0.83 ± 0.03	0.85 ± 0.09	*0.87 ± 0.06*	0.82 ± 0.06	0.85 ± 0.07	$\mathbf{0.89 \pm 0.06}$
Breast-C	0.68 ± 0.08	0.62 ± 0.09	0.63 ± 0.09	0.69 ± 0.09	0.67 ± 0.06	0.66 ± 0.05	*0.70 ± 0.08*	$\mathbf{0.71 \pm 0.09}$
Heart-C	0.86 ± 0.09	0.87 ± 0.08	0.88 ± 0.07	0.86 ± 0.08	0.89 ± 0.05	0.88 ± 0.04	*0.90 ± 0.05*	$\mathbf{0.92 \pm 0.04}$
Liver	$\mathbf{0.72 \pm 0.12}$	0.68 ± 0.09	0.68 ± 0.10	0.69 ± 0.13	0.70 ± 0.10	0.65 ± 0.13	0.68 ± 0.10	*0.71 ± 0.09*
Vote	$\mathbf{0.99 \pm 0.02}$	*0.98 ± 0.04*	0.97 ± 0.01	0.97 ± 0.02	$\mathbf{0.99 \pm 0.01}$	0.97 ± 0.02	0.97 ± 0.01	*0.98 ± 0.01*
Card	0.82 ± 0.06	0.84 ± 0.05	0.89 ± 0.05	$\mathbf{0.94 \pm 0.07}$	0.88 ± 0.05	0.87 ± 0.08	0.85 ± 0.05	*0.93 ± 0.03*
German	$\mathbf{0.78 \pm 0.05}$	0.75 ± 0.06	*0.76 ± 0.02*	0.75 ± 0.06	0.74 ± 0.06	0.76 ± 0.06	*0.77 ± 0.06*	$\mathbf{0.78 \pm 0.04}$
Mtb-May	$0.72 \pm 19 \times 10^{-3}$	$0.70 \pm 18 \times 10^{-3}$	$0.72 \pm 11 \times 10^{-3}$	$0.73 \pm 10 \times 10^{-3}$	$\mathbf{0.79 \pm 1 \times 10^{-3}}$	*0.77 ± 2 × 10⁻³*	$0.76 \pm 8 \times 10^{-3}$	*0.77 ± 7 × 10⁻³*
Mtb-June	$0.96 \pm 21 \times 10^{-3}$	$0.95 \pm 16 \times 10^{-3}$	$0.96 \pm 15 \times 10^{-3}$	$0.96 \pm 14 \times 10^{-3}$	$0.97 \pm 1 \times 10^{-3}$	*0.98 ± 3 × 10⁻³*	$0.97 \pm 2 \times 10^{-3}$	$\mathbf{0.99 \pm 1 \times 10^{-3}}$
Mtb-July	$0.76 \pm 22 \times 10^{-3}$	$0.79 \pm 14 \times 10^{-3}$	$0.80 \pm 15 \times 10^{-3}$	*0.81 ± 15 × 10⁻³*	$0.80 \pm 3 \times 10^{-3}$	$0.78 \pm 5 \times 10^{-3}$	$0.77 \pm 6 \times 10^{-3}$	$\mathbf{0.84 \pm 1 \times 10^{-3}}$
SC- May	$0.77 \pm 21 \times 10^{-3}$	$0.78 \pm 16 \times 10^{-3}$	$0.80 \pm 11 \times 10^{-3}$	$0.76 \pm 13 \times 10^{-3}$	$0.80 \pm 2 \times 10^{-3}$	$0.78 \pm 4 \times 10^{-3}$	*0.81 ± 1 × 10⁻³*	$\mathbf{0.85 \pm 7 \times 10^{-3}}$
SC-June	$0.95 \pm 11 \times 10^{-3}$	$0.95 \pm 21 \times 10^{-3}$	$0.96 \pm 18 \times 10^{-3}$	$0.96 \pm 14 \times 10^{-3}$	$\mathbf{0.99 \pm 1 \times 10^{-3}}$	*0.98 ± 2 × 10⁻³*	*0.98 ± 2 × 10⁻³*	$\mathbf{0.99 \pm 1 \times 10^{-3}}$
SC-July	$0.81 \pm 21 \times 10^{-3}$	$0.78 \pm 15 \times 10^{-3}$	$0.79 \pm 23 \times 10^{-3}$	$0.79 \pm 29 \times 10^{-3}$	$0.85 \pm 5 \times 10^{-3}$	*0.89 ± 4 × 10⁻³*	$\mathbf{0.90 \pm 2 \times 10^{-3}}$	$0.88 \pm 4 \times 10^{-3}$
$\overline{AUC}_G$	0.82	0.80	0.82	0.83	*0.85*	0.83	0.84	**0.88**
$\overline{R_{AUC}}$	5.76	6.67	5.17	4.70	*3.29*	4.64	3.97	**1.76**

The best result is in bold face and the second best result in italics.

Table 4
Comparison of the proposed basis functions to other basis functions: mean and standard deviation of the generalization Root Mean Squared Error results ($RMSE_G$), mean $RMSE_G$ over all the datasets, and mean $RMSE_G$ ranking ($\overline{R_{RMSE}}$).

	Method ($RMSE_G$)							
	OS-ELM-RBF	ELM-SIG	ELM-RBF	EELM-RBF	SRBF	CRBF	IMRBF	q-Gaussian
Labor	0.32 ± 0.21	0.34 ± 0.23	0.27 ± 0.23	0.29 ± 0.19	0.22 ± 0.21	$\mathbf{0.18 \pm 0.19}$	0.24 ± 0.17	*0.19 ± 0.17*
Promoters	0.42 ± 0.27	0.52 ± 0.32	0.40 ± 0.22	0.41 ± 0.22	0.43 ± 0.14	*0.39 ± 0.11*	0.41 ± 0.09	$\mathbf{0.35 \pm 0.10}$
Hepatitis	0.37 ± 0.07	0.41 ± 0.07	0.40 ± 0.08	0.39 ± 0.06	$\mathbf{0.28 \pm 0.03}$	0.36 ± 0.07	*0.31 ± 0.05*	0.34 ± 0.06
Sonar	0.41 ± 0.04	0.47 ± 0.05	0.40 ± 0.06	$\mathbf{0.39 \pm 0.07}$	0.41 ± 0.08	0.44 ± 0.10	0.43 ± 0.09	$\mathbf{0.39 \pm 0.08}$
Heart	0.40 ± 0.05	0.39 ± 0.07	0.42 ± 0.05	0.40 ± 0.05	$\mathbf{0.33 \pm 0.05}$	0.38 ± 0.08	0.38 ± 0.09	*0.35 ± 0.07*
Breast-C	0.44 ± 0.07	0.48 ± 0.06	0.49 ± 0.06	0.42 ± 0.06	0.47 ± 0.03	0.49 ± 0.04	0.43 ± 0.03	$\mathbf{0.41 \pm 0.03}$
Heart-C	0.47 ± 0.07	0.43 ± 0.08	*0.35 ± 0.07*	0.36 ± 0.06	0.37 ± 0.05	0.40 ± 0.05	0.35 ± 0.03	$\mathbf{0.33 \pm 0.04}$
Liver	$\mathbf{0.45 \pm 0.03}$	0.49 ± 0.07	0.47 ± 0.05	0.49 ± 0.03	*0.47 ± 0.01*	0.50 ± 0.03	0.49 ± 0.02	0.48 ± 0.02
Vote	0.17 ± 0.09	0.21 ± 0.06	0.18 ± 0.05	*0.17 ± 0.06*	0.19 ± 0.06	0.22 ± 0.07	0.23 ± 0.07	$\mathbf{0.15 \pm 0.07}$
Card	0.31 ± 0.05	0.31 ± 0.03	0.30 ± 0.04	$\mathbf{0.28 \pm 0.05}$	0.33 ± 0.04	0.35 ± 0.05	0.35 ± 0.06	*0.29 ± 0.03*
German	$\mathbf{0.39 \pm 0.05}$	0.45 ± 0.02	0.43 ± 0.03	*0.40 ± 0.03*	0.46 ± 0.02	0.44 ± 0.03	0.46 ± 0.03	0.41 ± 0.02
Mtb-May	$0.56 \pm 12 \times 10^{-3}$	$0.54 \pm 13 \times 10^{-3}$	$0.54 \pm 12 \times 10^{-3}$	$0.50 \pm 12 \times 10^{-3}$	$\mathbf{0.40 \pm 3 \times 10^{-3}}$	$0.47 \pm 5 \times 10^{-3}$	$0.49 \pm 9 \times 10^{-3}$	*$0.44 \pm 3 \times 10^{-3}$*
Mtb-June	$0.26 \pm 12 \times 10^{-3}$	$0.26 \pm 15 \times 10^{-3}$	$0.24 \pm 10 \times 10^{-3}$	$0.22 \pm 17 \times 10^{-3}$	*$0.12 \pm 2 \times 10^{-3}$*	$0.20 \pm 6 \times 10^{-3}$	$0.15 \pm 3 \times 10^{-3}$	$\mathbf{0.08 \pm 5 \times 10^{-3}}$
Mtb-July	$0.50 \pm 19 \times 10^{-3}$	$\mathbf{0.39 \pm 17 \times 10^{-3}}$	$0.48 \pm 14 \times 10^{-3}$	$0.47 \pm 11 \times 10^{-3}$	$0.49 \pm 1 \times 10^{-3}$	$0.47 \pm 1 \times 10^{-3}$	$0.52 \pm 1 \times 10^{-3}$	*$0.41 \pm 1 \times 10^{-3}$*
SC-May	$0.61 \pm 10 \times 10^{-3}$	$0.53 \pm 14 \times 10^{-3}$	$0.55 \pm 9 \times 10^{-3}$	$0.56 \pm 10 \times 10^{-3}$	$0.46 \pm 5 \times 10^{-3}$	$0.48 \pm 3 \times 10^{-3}$	*$0.41 \pm 6 \times 10^{-3}$*	$\mathbf{0.39 \pm 5 \times 10^{-3}}$
SC-June	$0.27 \pm 11 \times 10^{-3}$	$0.25 \pm 12 \times 10^{-3}$	$0.24 \pm 15 \times 10^{-3}$	$0.24 \pm 15 \times 10^{-3}$	$\mathbf{0.10 \pm 4 \times 10^{-3}}$	$0.17 \pm 3 \times 10^{-3}$	$0.15 \pm 3 \times 10^{-3}$	$\mathbf{0.13 \pm 7 \times 10^{-3}}$
SC-July	$0.43 \pm 9 \times 10^{-3}$	$0.47 \pm 11 \times 10^{-3}$	$0.48 \pm 10 \times 10^{-3}$	$0.51 \pm 10 \times 10^{-3}$	$0.39 \pm 2 \times 10^{-3}$	$0.37 \pm 2 \times 10^{-3}$	$\mathbf{0.32 \pm 5 \times 10^{-3}}$	*$0.35 \pm 3 \times 10^{-3}$*
$\overline{RMSE_G}$	0.39	0.40	0.39	0.38	*0.34*	0.34	0.35	**0.31**
$\overline{R_{RMSE}}$	5.55	6.20	5.0	4.50	*3.79*	4.70	4.38	**1.85**

The best result is in bold face and the second best result in italics.

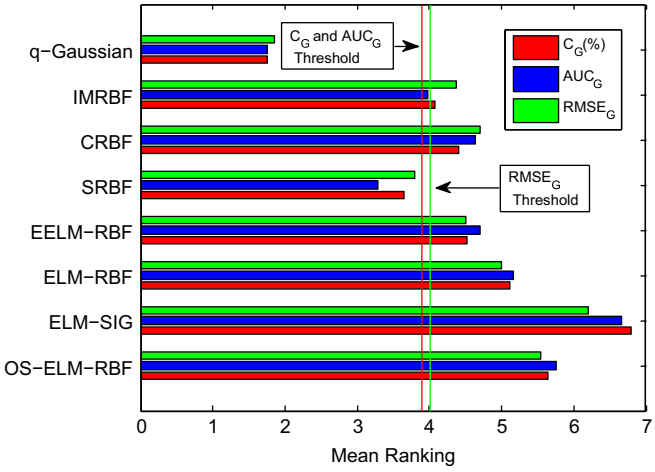


Fig. 3. Bonferroni–Dunn graphic for $\alpha = 0.05$.

the independence condition was not truly verified in ten-fold cross validation (a portion of samples was used either for training or testing in different partitions).

Therefore, in order to determine the statistical significance of the rank differences observed for each method in the different datasets, we have carried out three non-parametric Friedman tests [51] with the ranking of C_G, AUC_G and $RMSE_G$ of the best models as the test variables. These tests show that the effect of the method used for classification is statistically significant at a significance level of 5% for C_G, AUC_G and $RMSE_G$, as the confidence interval is $C_0 = (0, F_{0.05} = 2.09)$ and the F-distribution statistical values are $F^* = 9.21 \notin C_0$ for C_G, $F^* = 9.92 \notin C_0$ for AUC_G and $F^* = 6.28 \notin C_0$ for $RMSE_G$. Consequently, we reject the null-hypothesis stating that all algorithms perform equally in mean ranking of C_G, AUC_G and $RMSE_G$.

In our case, the Friedman test shows significant differences in the results. Due to these differences, three post-hoc statistical analyses were required. These analyses have been performed choosing the best performing model, q-Gaussian, as the control method, for comparison with the rest of the models.

Table 5
Adjusted p-values using $C_G(\%)$, AUC_G and $RMSE_G$ as the test variables (q-Gaussian is the control method).

Model	Unadjusted p	$p_{Bonferroni}$	p_{Holm}	$p_{Hochberg}$
$C_G(\%)$ variable test				
ELM-Sig	2.14×10^{-9}	1.50×10^{-8}	1.50×10^{-8}	1.50×10^{-8}
OS-ELM-RBF	3.82×10^{-6}	2.67×10^{-5}	2.29×10^{-5}	2.29×10^{-5}
ELM-RBF	6.58×10^{-5}	4.60×10^{-4}	3.29×10^{-4}	3.29×10^{-4}
EELM-RBF	9.99×10^{-4}	69.9×10^{-4}	39.9×10^{-4}	39.9×10^{-4}
CRBF	0.00162	0.01140	0.00488	0.00488
IMRBF	0.00568	0.03977	0.01136	0.01136
SRBF	0.02506	0.17543	0.02506	0.02506
AUC_G variable test				
ELM- Sig	5.03×10^{-9}	3.52×10^{-8}	3.52×10^{-8}	3.52×10^{-8}
OS-ELM-RBF	1.92×10^{-6}	1.34×10^{-5}	1.15×10^{-5}	1.15×10^{-5}
ELM-RBF	4.89×10^{-5}	3.42×10^{-4}	2.44×10^{-4}	2.44×10^{-4}
EELM-RBF	4.64×10^{-4}	32.4×10^{-4}	18.5×10^{-4}	18.0×10^{-4}
CRBF	6.02×10^{-4}	42.1×10^{-4}	18.5×10^{-4}	18.0×10^{-4}
IMRBF	0.00865	0.04056	0.01730	0.01730
SRBF	0.04870	0.11092	0.04570	0.04570
$RMSE_G$ variable test				
ELM-Sig	2.20×10^{-7}	1.54×10^{-6}	1.54×10^{-6}	1.54×10^{-6}
OS-ELM-RBF	1.02×10^{-5}	7.20×10^{-5}	6.17×10^{-5}	6.17×10^{-5}
ELM-RBF	1.79×10^{-4}	12.5×10^{-4}	8.99×10^{-4}	8.99×10^{-4}
CRBF	6.84×10^{-4}	47.9×10^{-4}	27.3×10^{-4}	27.3×10^{-4}
EELM-RBF	0.00162	0.01140	0.00488	0.00488
IMRBF	0.00260	0.01825	0.00521	0.00521
SRBF	0.02086	0.14603	0.02086	0.02086

The Bonferroni–Dunn test considers that the performance of any two classifiers is deemed to be significantly different if their mean ranks differ by at least the critical difference (CD):

$$CD = q\sqrt{\frac{K(K+1)}{6D}}, \tag{12}$$

where K and D are the number of classifiers and datasets, and q can be computed as suggested in [52]. Fig. 3 illustrates the application of the Bonferroni–Dunn test for each test variable. This graph is a bar chart where the bars have a height proportional

to the average rank obtained for each algorithm, following the procedure of Friedman. Adding the ranking value of the lowest bar (associated with the q-Gaussian model) to the Critical Difference (CD) value obtains a vertical line (denoted as "Threshold"), which is displayed in the graphic. The bars exceeding this line are those associated with the algorithms whose performance is significantly worse than the control algorithm. From the results of these tests, it can be concluded that the q-Gaussian model produced a significantly better C_G, AUC_G and $RMSE_G$ ranking than all the other RBFs and methodologies considered except for the SRBF model.

More powerful tests are applied, such as Holm's and Hochberg's [52] tests, to compare the control model (q-Gaussian RBFNN) with the rest of the models. Holm and Hochberg tests are detailed in [50,52]. Table 5 shows all the adjusted p-values for each comparison, using C_G, AUC_G and $RMSE_G$ as the test variables and q-Gaussian RBFNN as the test method. The adjusted p-values represent the lowest level of significance of a hypothesis that results in a rejection [53]. In this manner, we can find out whether two algorithms are significantly different and also obtain a metric of how different they are. Holm's and Hochberg's tests indicate that the control model (q-Gaussian) outperforms all the remaining models in C_G, AUC_G and $RMSE_G$, which justifies the proposal.

4.4. Analysis of the best q-Gaussian models obtained for the agricultural remote sensing problems

As mentioned above, these datasets were formerly evaluated with Evolutionary Product-Unit Neural Networks (EPUNNs) and with their hybridization with Logistic Regression models (LRIPU). Like the methodology proposed, all these models predict a probability, thus differentiating the pixels of the image (for concluding *R. segetum* presence or absence).

Taking into account only the best model per dataset, q-Gaussian RBF outperformed both EPUNN and LRIPU in five of the six datasets considered (Table 7).

This section studies in detail the best q-Gaussian RBFNN obtained for the biclass Matabueyes and Santa Cruz datasets in mid-June. The mathematical expressions of the different models are presented in Table 6, all of them being relatively simple.

We now justify the use of q-Gaussian RBF models instead of standard Gaussian RBF models for agricultural remote sensing problems.

Table 7
Comparison of the proposed basis functions to EPUNN and LRIPU.

	Best C_G(%)		
	EPUNN	LRIPU	q-Gaussian
Mtb-May	70.6	70.6	**72.50**
Mtb-June	98.7	99.2	**99.38**
Mtb-July	79.8	79.0	**80.00**
SC-May	78.4	77.5	**79.77**
SC-June	98.4	**98.7**	98.42
SC-July	83.1	84.3	**85.30**

The best result is in bold face.

Table 6
Probability expression of the best q-Gaussian RBF model for Matabueyes and Santa Cruz problems.

Best q-Gaussian RBF Matabueyes model

$$P = 1/(1+\exp(-5.92+7.89RBF_1+10.96RBF_2-6.47RBF_3+3.70RBF_4+1.45RBF_5))$$
$$RBF_1=(1-(1-0.17)d_1)^{1/(1-0.17)};\ RBF_2=(1-(1-0.52)d_2)^{1/(1-0.52)}$$
$$RBF_3=(1-(1+0.54)d_3)^{1/(1+0.54)};\ RBF_4=(1-(1+1.16)d_4)^{1/(1-1.16)}$$
$$RBF_5=(1-(1+0.69)d_5)^{1/(1+0.69)}$$
$$d_1=\left(\frac{\sqrt{(R^\star+1.92)^2+(G^\star-0.27)^2}}{1.25}\right)^2;\ d_2=\left(\frac{\sqrt{(R^\star+1.45)^2+(G^\star+0.07)^2}}{0.85}\right)^2$$
$$d_3=\left(\frac{\sqrt{(R^\star+0.38)^2+(G^\star+0.94)^2+(B^\star+1.16)^2}}{1.31}\right)^2;\ d_4=\left(\frac{\sqrt{(R^\star+1.70)^2+(B^\star-0.08)^2}}{2.06}\right)^2$$
$$d_5=\left(\frac{\sqrt{(B^\star-0.44)^2}}{0.74}\right)^2$$
$$CCR_T=98.30\%,\ CCR_G=99.38\%$$
$$AUC_T=0.99,\ AUC_G=0.99$$
$$RMSE_T=0.10,\ RMSE_G=0.07$$

Best q-Gaussian RBF Santa Cruz model

$$P = 1/(1+\exp(-4.03+6.23RBF_1+4.86RBF_2+6.87RBF_3-7.34RBF_4+12.06RBF_5-9.94RBF_6))$$
$$RBF_1=(1-(1+0.09)d_1)^{1/(1+0.09)};RBF_2=(1-(1-0.11)d_2)^{1/(1-0.11)}$$
$$RBF_3=(1-(1+0.44)d_3)^{1/(1+0.44)};RBF_4=(1-(1+0.20)d_4)^{1/(1+0.20)}$$
$$RBF_5=(1-(1-0.11)d_5)^{1/(1-0.11)};RBF_6=(1-(1+1.04)d_6)^{1/(1+1.04)}$$
$$d_1=\left(\frac{\sqrt{(R^\star+1.20)^2+(NIR^\star-1.95)^2}}{2.37}\right)^2;\ d_2=\left(\frac{\sqrt{(R^\star+1.04)^2+(NIR^\star-1.28)^2}}{1.22}\right)^2$$
$$d_3=\left(d\frac{\sqrt{(R^\star-0.33)^2+(NIR^\star-1.91)^2}}{0.77}\right)^2;\ d_4=\left(\frac{\sqrt{(R^\star-0.60)^2+(NIR^\star-0.19)^2}}{1.88}\right)^2$$
$$d_5=\left(\frac{\sqrt{(R^\star+1.24)^2+(B^\star-0.41)^2}}{1.75}\right)^2;\ d_6=\left(\frac{\sqrt{(R^\star-0.10)^2+(G^\star+1.05)^2}}{2.28}\right)^2$$
$$CCR_T=98.26\%,\ CCR_G=98.42\%$$
$$AUC_T=0.99,\ AUC_G=0.99$$
$$RMSE_T=0.10,\ RMSE_G=0.12$$

$$R^\star,G^\star,B^\star,NIR^\star\in[-2,2];\ (1-(1-q_i)d_i)\geq 0$$

Two and three dimensional graphic representations have been included of the non-linear relationship between the value associated with each q-Gaussian RBF and the corresponding input variables for the Matabueyes and Santa Cruz mid-June datasets, in Figs. 4 and 5. It is important to keep in mind that the input variables are scaled ($X_i \in [-2,2]$).

As shown in Figs. 4 and 5, the Hybrid Algorithm (HA) generates models with q values other than 1 (where q-Gaussian RBFNN is reduced to the standard Gaussian RBFNN), indicating the need for a great number of basis functions with q values under 1. These types of basis functions present a very selective response, with high activation for patterns close to the centroid and very small activation for distant patterns, since this type of basis function has almost no tail.

In order to use our models to derive agronomical information and evaluate the contribution of the bands studied to predict the probability of occurrence *R. segetum*, the response of the R and NIR bands is studied when the B and G bands are fixed to certain interpretable values (Fig. 6). An overhead view of the field in mid-June showed that the sunflower crop had established a closed canopy of greenish coloration, the *R. Segetum* patches were entirely flowering and yellow, growing over the sunflower plants and, in general, the soil was completely covered by the crop. The yellowish color of *R. segetum* was generated when the reflectance in the R and G regions was similar and even higher than in the B region. Furthermore, the sunflower generally showed a relative reflectance maximum in the G region due to the high amounts of chlorophyll content in the living vegetation and an absolute maximum in the NIR region due to its vigorous and dense canopy.

When the B and the G bands were fixed at low values (Fig. 6a and b), *R. segetum* was only absent in the region defined by NIR-band values which were much greater than R-band values. This scenario is typical of vigorous and healthy vegetation that, in our case, might be attributed to zones completely covered in a dense sunflower crop, free of weed infestation and scarcely affected by soil background. In the opposite scenario, formed by high values of B and G bands (Fig. 6c and d), the probability of weed occurrence was high when the R-band values were close to or greater than those of both the B-band and G-band. In this case, the low probability of weed occurrence can be assigned to two different situations: (1) the presence of non-infested sunflowers, when the NIR-band values are greater than those of the R-band and (2) the high background effect of bare soil in zones with low crop development, when the NIR-band values are lower than those of visible (B, G, R) bands. Another scenario studied was formed by values relatively greater in the G region than in other visible bands, which showed greenish areas in the fields studied (Fig. 6e and f). In this case, the probability distribution was partially similar to the previous plots except in the region delimited by low values of R and NIR bands, which might be

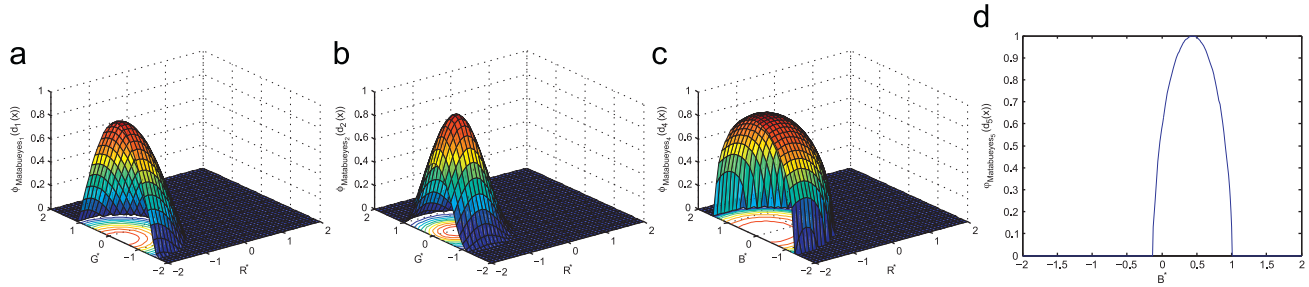


Fig. 4. Three and two dimensional graphical representation of the non-linear relationship between the value associated with the hidden nodes and the corresponding input variables, in the Matabueyes-mid June model given by the q-Gaussian RBFNN: (a): $\phi_{\text{Matabueyes}_1}$, (b): $\phi_{\text{Matabueyes}_2}$ (c): $\phi_{\text{Matabueyes}_4}$ (d): $\phi_{\text{Matabueyes}_5}$.

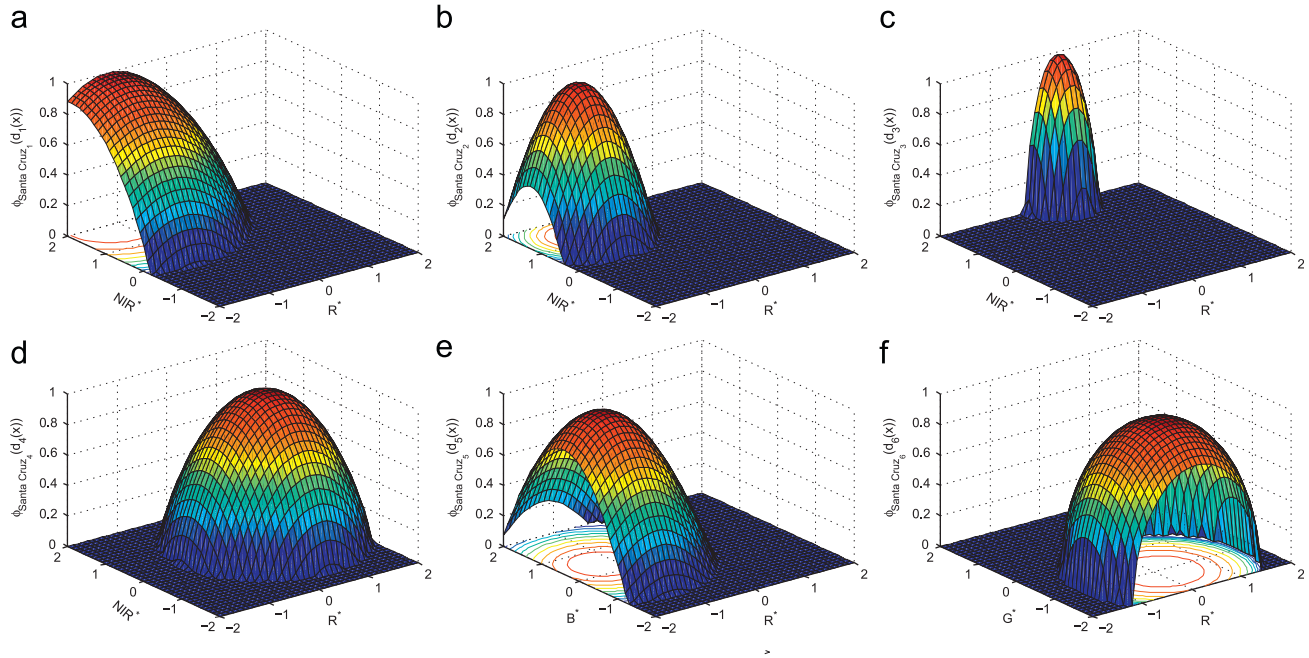


Fig. 5. Three-dimensional graphical representation of the non-linear relationship between the value associated with the hidden nodes and the corresponding input variables, in the Santa Cruz-mid June model given by the q-Gaussian RBFNN: (a): $\phi_{\text{SantaCruz}_1}$; (b): $\phi_{\text{SantaCruz}_2}$; (c): $\phi_{\text{SantaCruz}_3}$; (d): $\phi_{\text{SantaCruz}_4}$; (e): $\phi_{\text{SantaCruz}_5}$; (f): $\phi_{\text{SantaCruz}_6}$.

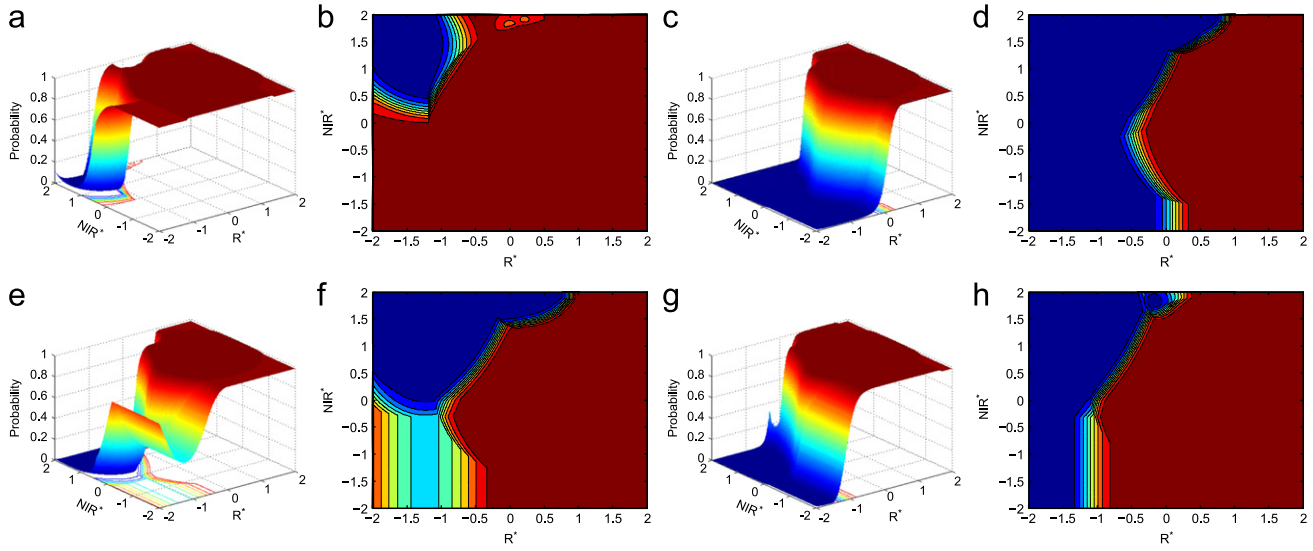


Fig. 6. Contribution of the R and NIR multispectral bands to the probability of *R. segetum* occurrence in the field Santa-Cruz in mid-June for certain fixed values of blue (B) and green (G) bands: (a,b) B = −2, G = −2; (c,d) B = 1, G = 1; (e,f) B = −1, G = 1; (g,h) B = 1, G = −1. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

attributed to the zones where the sunflower crop was in an advanced stage of development (initial desiccation), and then it could be confused with *R. segetum* patches. Although in mid-June both species showed consistent spectral differences, the coincidence of different phenological stages of the two plant species could obviously complicate their discrimination, as previously reported by [54]. Lastly, the scenario formed by B-band values relatively greater than G-band ones was also plotted (Fig. 6g and h), although it might be unusual in the data studied because, from an agronomical point of view, it could only be attributed to zones with bare soil or hardly covered by vegetation. In this assumption, it is noteworthy that the threshold for *R. segetum* occurrence was established at R-band values greater than G-band values, demonstrating that the influence of the B region was practically negligible, as also inferred from the analysis of the previous plots.

As a general conclusion, the probability of *R. segetum* presence depended mainly on the interaction between the R and NIR bands. A minor contribution was derived from the B and G bands, because only a reduced part of the figures changed when values of the two bands were interchangeable with each other. Previous reports also demonstrated that differences between the R and the NIR bands were very good at separating species since this spectral region was strongly affected by chlorophyll and pigment contents and changes in the phenological stages of the plants. Furthermore, the distinction between vegetation and bare soil is also usually based on both bands [55]. These results could provide information for mapping this weed in sunflower crops using multispectral bands from remote sensed imagery, which in the future may help contribute to efficient weed management strategies and timely site-specific herbicide application decisions.

5. Conclusions

A Hybrid Algorithm (HA) with specific operators has been developed to automatically find *q*-Gaussian Radial basis-Function Neural Networks to solve binary-classification problems. The HA optimizes all the parameters related to neural network architecture, i.e., the number of hidden neurons and their configuration. The evaluation of the model and the algorithm for the 17 datasets considered showed that *q*-Gaussian RBF obtained higher accuracy, Root Mean Squared Error and area under the ROC curve when compared to other RBFs and to other recent neural network approaches.

These models can provide information to program suitable wavelengths of further Compact Airborne Spectral Imager (CASI) images for Site-Specific Weed Management (SSWM). Moreover, comparing *q*-Gaussian RBFNNs to EPUNNs and LRIPU has shown that *q*-Gaussian RBFNNs models are very competitive models and has demonstrated how they are able to analyze multispectral imagery to predict *R. segetum* presence in the field under study, providing a useful tool for early SSWM.

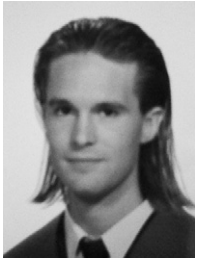
Acknowledgement

This work has been partially subsidized by the TIN 2008-06681-C06-03 project of the Spanish Inter-Ministerial Commission of Science and Technology (MICYT), FEDER funds and the P08-TIC-3745 project of the "Junta de Andalucía" (Spain). The research of Francisco Fernández-Navarro has been funded by the "Junta de Andalucia" Predoctoral Program, grant reference P08-TIC-3745.

References

[1] M.J. Er, S. Wu, J. Lu, H.L. Toh, Face recognition with radial basis function (rbf) neural networks, IEEE Transactions on Neural Networks 13 (3) (2002) 697–710.
[2] C. Harpham, C.W. Dawson, The effect of different basis functions on a radial basis function network for time series prediction: a comparative study, Neurocomputing 69 (16-18) (2006) 2161–2170.
[3] M. Li, G. Huang, P. Saratchandran, N. Sundararajan, Performance evaluation of gap-rbf network in channel equalization, Neural Processing Letters 22 (2) (2005) 223–233.
[4] M.D. Perez-Godoy, A. Fernández, A.J. Rivera, M.J. del Jesus, Analysis of an evolutionary RBFN design algorithm, CO2RBFN, for imbalanced data sets, Pattern Recognition Letters 31 (2010) 2375–2388.
[5] H. Wei, S. Amari, Dynamics of learning near singularities in radial basis function networks, Neural Networks 21 (7) (2008) 989–1005.
[6] S.A. Billings, H. Wei, M.A. Balikhin, Generalized multiscale radial basis function networks, Neural Networks 20 (10) (2007) 1081–1094.
[7] A. Punzi, A. Sommariva, M. Vianello, Meshless cubature over the disk using thin-plate splines, Journal of Computational and Applied Mathematics 221 (2) (2008) 430–436.

[8] A. Saranli, B. Baykal, Complexity reduction in radial basis function (RBF) networks by using radial B-spline functions, Neurocomputing 18 (1–3) (1998) 183–194.
[9] F. Fernández-Navarro, C. Hervás-Martínez, P.A. Gutiérrez, M. Cruz-Ramírez, M. Carbonero-Ruz, Evolutionary *q*-Gaussian radial basis functions for binary-classification, Fifth International Conference on Hybrid Artificial Intelligent Systems, Lecture Notes in Artificial Intelligence, vol. 6077, Springer, 2010, pp. 280–287.
[10] N. Gueorguieva, I. Valova, Building rbf neural network topology through potential functions, in: Artificial Neural Networks and Neural Information Processing – ICANN/ICONIP 2003, Lecture Notes in Computer Science, vol. 2714, 2003, pp. 179–180.
[11] R. Lippmann, Pattern classification using neural networks, IEEE Communications Magazine 27 (1989) 47–64.
[12] F.J. Martínez-Estudillo, C. Hervás-Martínez, P.A. Gutiérrez, A.C. Martínez-Estudillo, Evolutionary product-unit neural networks classifiers, Neurocomputing 72 (1–2) (2008) 548–561.
[13] P.A. Gutiérrez, F. López-Granados, J.M. Peña-Barragán, M. Jurado-Expósito, M.T. Gómez-Casero, C. Hervás, Mapping sunflower yield as affected by *Ridolfia segetum* patches and elevation by applying evolutionary product unit neural networks to remote sensed data, Computers and Electronics in Agriculture 60 (2) (2008) 122–132.
[14] P.A. Gutiérrez, C. Hervás-Martínez, M. Carbonero, J.C. Fernández, Combined projection and kernel basis functions for classification in evolutionary neural networks, Neurocomputing 72 (13–15) (2009) 2731–2742.
[15] R. Tinos, L. Murta, Selection of radial basis functions via genetic algorithms in pattern recognition problems, in: Tenth Brazilian Symposium on Neural Networks, 2008. SBRN '08, 2008, pp. 171–176.
[16] C. Tsallis, Possible generalization of Boltzmann–Gibbs statistics, Journal of Statistical Physics 52 (1–2) (1988) 479–487.
[17] C. Tsallis, R.S. Mendes, A.R. Plastino, The role of constraints within generalized nonextensive statistics, Physica A: Statistical Mechanics and its Applications 261 (3–4) (1998) 534–554.
[18] R. Díaz, E. Pariguan, On the Gaussian q-distribution, Journal of Mathematical Analysis and Applications 358 (1) (2009) 1–9.
[19] F. Fernández-Navarro, C. Hervás-Martínez, M. Cruz, P.A. Gutierrez, A. Valero, Evolutionary *q*-Gaussian radial basis function neural network to determine the microbial growth/no growth interface of *Staphylococcus aureus*, Applied Soft Computing 11 (3) (2011) 3012–3020.
[20] L. Yingwei, N. Sundararajan, P. Saratchandran, Performance evaluation of a sequential minimal radial basis function (rbf) neural network learning algorithm, IEEE Transactions on Neural Networks 9 (2) (1998) 308–318.
[21] G.-B. Huang, P. Saratchandran, N. Sundararajan, S. Member, S. Member, N. Sundararajan, An efficient sequential learning algorithm for growing and pruning rbf (gap-rbf) networks, IEEE Transaction on System, Man and Cybernetics, Part-B: Cybernetics 34 (2004) 2284–2292.
[22] S. Suresh, N. Sundararajan, P. Saratchandran, A sequential multi-category classifier using radial basis function networks, Neurocomputing 71 (2008) 1345–1358.
[23] G.B. Huang, Q. Zhu, C. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1–3) (2006) 489–501.
[24] G. Huang, X. Ding, H. Zhou, Optimization method based extreme learning machine for classification, Neurocomputing 74 (2010) 155–163.
[25] F. Fernández-Navarro, C. Hervás-Martínez, J. Sánchez-Monedero, P.A. Gutierrez, Melm-grbf: a modified version of the extreme learning machine for generalized radial basis function neural networks, Neurocomputing 74 (2011) 2502–2510.
[26] G.-B. Huang, L. Chen, C.K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Transactions on Neural Networks 17 (4) (2006) 879–892. URL ⟨http://doi.ieeecomputersociety.org/10.1109/TNN.2006.875977⟩.
[27] Q.-Y. Zhu, A. Qin, P. Suganthan, G.-B. Huang, Evolutionary extreme learning machine, Pattern Recognition 38 (10) (2005) 1759–1763.
[28] G.-B. Huang, L. Chen, Convex incremental extreme learning machine, Neurocomputing 70 (16–18) (2007) 3056–3062.
[29] Javier Sánchez-Monedero, Pedro Gutiérrez, F. Fernández-Navarro, C. Hervás-Martínez, Weighting efficient accuracy and minimum sensitivity for evolving multi-class classifiers. Neural Processing Letters (2011), 1–16. doi:10.1007/s11063-011-9186-9.
[30] N.-Y. Liang, G.-B. Huang, P. Saratchandran, N. Sundararajan, A fast and accurate online sequential learning algorithm for feedforward networks, IEEE Transactions on Neural Networks 17 (6) (2006) 1411–1423.
[31] P.J. Angeline, G.M. Sauders, J.B. Pollack, An evolutionary algorithm that constructs recurrent neural networks, IEEE Transactions on Neural Networks 5 (1994) 54–65.
[32] F. Fernández-Navarro, A. Valero, C. Hervás-Martínez, P. Gutíerrez, R. García-Gimeno, G. Zurera-Cosano, Development of a multi-classification neural network model to determine the microbial growth/no growth interface, International Journal of Food Microbiology 141 (2010) 203–212.
[33] C.M. Bishop, Improving the generalization properties of radial basis function neural networks, Neural Computation 3 (4) (1991) 579–581.
[34] A. Abraham, E. Corchado, J.M. Corchado, Hybrid learning machines, Neurocomputing 72 (13–15) (2009) 2729–2730.
[35] E. Corchado, A. Abraham, A.C.P.L.F. de Carvalho, Hybrid intelligent algorithms and applications, Information Sciences (2010) 2633–2634.
[36] P.A. Gutiérrez, F. López-Granados, J.M. Peña-Barragán, M. Jurado-Expósito, C. Hervás-Martínez, Logistic regression product-unit neural networks for mapping *Ridolfia segetum* infestations in sunflower crop using multitemporal remote sensed data, Computers and Electronics in Agriculture 62 (2) (2008) 293–306.
[37] P.A. Gutiérrez, C. Hervás-Martínez, J.C. Fernández, M. Jurado-Expósito, J.M. Peña-Barragán, F. López-Granados, Structural simplification of hybrid neuro-logistic regression models in multispectral analysis of remote sensed data, Neural Network World 19 (1) (2009) 3–20.
[38] D.S. Broomhead, D. Lowe, Multivariable functional interpolation and adaptive networks, Complex Systems 2 (1988) 321–355.
[39] K. Shkurko, X. Qi, A radial basis function and semantic learning space based composite learning approach to image retrieval, in: Proceedings on ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 1, 2007, pp. 945–948.
[40] J. Zhang, H. Li, A reconstruction approach to CT with cauchy RBFs network, in: Advances in Neural Networks – ISNN 2004, Lecture Notes in Computer Science, vol. 3174, 2004, pp. 234–236.
[41] F. Fernández-Navarro, C. Hervás-Martínez, C. García-Alonso, M. Torres-Jimenez, Determination of relative agrarian technical efficiency by a dynamic over-sampling procedure guided by minimum sensitivity, Expert Systems with Applications 38 (10) (2011) 12483–12490.
[42] F. Fernández-Navarro, C. Hervás-Martínez, P. Gutíerrez, A dynamic over-sampling procedure based on sensitivity for multi-class problems, Pattern Recognition 44 (2011) 1821–1833.
[43] C. Igel, M. Hüsken, Empirical evaluation of the improved Rprop learning algorithms, Neurocomputing 50 (6) (2003) 105–123.
[44] A. Asuncion, D. Newman, UCI machine learning repository, URL ⟨http://www.ics.uci.edu/∼mlearn/MLRepository.html⟩, 2007.
[45] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, A. Lendasse, OP-ELM: optimally pruned extreme learning machine, IEEE Transactions on Neural Networks 21 (1) (2010) 158–162.
[46] R. Storn, K. Price, Differential evolution. A fast and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization 11 (1997) 341–359.
[47] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, Information Processing and Management 45 (2009) 427–437.
[48] C. Hervás-Martínez, F. Martínez-Estudillo, Logistic regression using covariates obtained by product-unit neural network models, Pattern Recognition 40 (1) (2007) 52–64.
[49] C. Hervás-Martínez, F.J. Martínez-Estudillo, M. Carbonero-Ruz, Multilogistic regression by means of evolutionary product-unit neural networks, Neural Networks 21 (7) (2008) 951–961.
[50] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.
[51] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, Annals of Mathematical Statistics 11 (1) (1940) 86–92.
[52] Y. Hochberg, A. Tamhane, Multiple Comparison Procedures, John Wiley & Sons, 1987.
[53] S. Wright, Statistical comparisons of classifiers over multiple data sets, Biometrics 48 (1992) 1005–1013.
[54] J. Peña-Barragán, F. López-Granados, M. Jurado-Expósito, L. García-Torres, Spectral discrimination of *Ridolfia segetum* and sunflower as affected by phenological stage, Weed Research 46 (1) (2006) 10–21.
[55] F. López-Granados, Weed detection for site-specific weed management: mapping and real-time approaches, Weed Research 51 (1) (2011) 1–11.



Francisco Fernández Navarro was born in Córdoba, Spain, in 1984. He received the B.S. Degree in computer science from the University of Córdoba, Spain, in 2007. He is currently working toward the Ph.D. Degree at the department of Computer Science and Numerical Analysis. His current interests include radial basis function neural networks, evolutionary computation and Hybrid Algorithms.

César Hervás Martínez was born in Cuenca, Spain. He received the B.S. Degree in statistics and operating research from the Universidad Complutense, Spain, in 1978 and the Ph.D. Degree in mathematics from the University of Seville, Seville, Spain, in 1986. He is a Professor with the University of Córdoba in the department of computing and numerical analysis, in the area of computer science and artificial intelligence. His current research interests include neural networks, evolutionary computation, and the modeling of natural systems.

Pedro Antonio Gutiérrez was born in Córdoba, Spain, in 1982. He received the B.S. Degree in Computer Science from the University of Seville, Spain, in 2006 and the Ph.D. Degree in Computer Science and Artificial Intelligence from the University of Granada, Granada, Spain, in 2009. He is an Assistant Professor with the University of Córdoba in the department of Computer Science and Numerical Analysis. His current research interests include neural networks and their applications, evolutionary computation and Hybrid Algorithms.

Jose Manuel Peña Barragán was born in Córdoba, Spain. He received the B.S. Degree in Agronomic Engineering from the University of Córdoba, Spain, in 2002 and the Ph.D. Degree in Agricultural Sciences from the University of Córdoba, Córdoba, Spain, in 2006. He is a Research Scientist of the Spanish National Research Council (CSIC) in the Institute for Sustainable Agriculture (IAS), the unique research center within the CSIC mission in agricultural research. His current research interest is centerd on the reduction of agrochemical applications by means of precision agriculture strategies.

Francisca López Granados was born in Córdoba, Spain. He received the B.S. Degree in Biology from the University of Córdoba, Spain, in 1985 and the Ph.D. Degree in Biology from the University of Córdoba, Spain, in 1991. He is a Research Scientist of the Spanish National Research Council (CSIC). His current research interest is centerd on the reduction of agrochemical applications by means of precision agriculture strategies.