Reconstruction of Multivariate Signals Using Q-Gaussian Radial Basis Function Network

LEV Silva, JJ Duque, R Tinós, LO Murta Jr

University of São Paulo, Ribeirão Preto, São Paulo, Brazil

Abstract

Radial basis function Networks (RBFNs) have been successfully employed in different Machine Learning problems. The use of different radial basis functions in RBFN has been reported in the literature. Here, we discuss the use of the q-Gaussian function as a radial basis function employed in RBFNs. An interesting property of the q-Gaussian function is that it can continuously and smoothly reproduce different radial basis functions, like the Gaussian, the Inverse Multiquadratic, and the Cauchy functions, by changing a real parameter q. In addition, we discuss the mixed use of different shapes of radial basis functions in only one RBFN. For this purpose, a Genetic Algorithm is employed to select the number of hidden neurons, width of each RBF, and q parameter of the q-Gaussian associated with each radial unit.

Network training is the search for optimal values of the radius and the q-parameter of each radial basis Gaussian. The minimum and maximum numbers of basis function in the mid layer are defined a priori. The k-means clustering algorithm was employed to calculate each set of center positions of the q-Gaussians. In training stage with a multivariate signal with n variable, the network inputs are the n samples of each channel at once, except for the channel which part of the data is missing, which was used as desired output.

Results from testing dataset were precise for good and moderate quality signals. However, if channel which part is missing is very noisy, the reconstruction, in general, was not so good. This fact could be explained by the artificial network training that is strongly dependent on the desired output channel, getting to learn with certain efficiency even when some of the inputs are noisy.

1. Introduction

In medical multivarite signals settings, we frequently face the problem of misdetection of one or more component signal due to various causes, including electronic failure and noise corruption. Therefore, the data in such situation is incomplete. The method proposed here aims to recover the missing data parts using Radial Basis Function (RBF) Network.

RBF Networks are a class of Artificial Neural Networks where RBFs are used to compute the activation of artificial neurons. RBF Networks have been successfully employed in real function approximation and pattern recognition problems. In general, RBF Networks are associated with architectures with two layers, where the hidden layer employs RBFs to compute the activation of the neurons. Different RBFs have been used, like the Gaussian, the Inverse Multiquadratic, and the Cauchy functions [1]. In the output layer, the activations of the hidden units are combined in order to produce outputs. While there are weights in the output layer, they are not present in the hidden layer. Here we used q-Gaussian as function basis generalizing the three previous basis function [2].

Choosing the parameters of the radial basis units means to determine the number of hidden neurons, the type, widths, and centers of the RBFs. In several cases, the first three parameters are previously defined, and only the radial basis centers are optimized [3]. Besides the centers, the number of hidden neurons [4], [5] and the widths [6] can be still optimized. In general, all the radial units have the same type of RBF, e.g., the Gaussian function, which is chosen before the training.

A Genetic Algorithm (GA) is employed to select the number of hidden neurons, center, type, and width of each RBF associated with each hidden unit.

2. Methods

RBF are a class of real-valued functions where its output depends on the distance between the input pattern and a point **c**, defined as the center of the RBF. Moody and Darken proposed the use of RBFs in Artificial Neural Networks (ANNs) inspired by the selective response of some neurons [3]. ANNs where RBFs are used as activation functions are named RBF Networks. The architecture and the learning of RBF Networks are described in the next sections.

2.1. Architecture

RBF Networks can have any number of hidden layers and outputs with linear or nolinear activation. However, RBF Networks are generally associated with architectures with only one hidden layer without weights and with an output layer with linear activation. Such architecture is employed because it allows the separation of the training in two phases: when the radial units parameters are determined, the weights of the output layer generally can be easily computed.

The most common RBF is the Gaussian function, which is given by

$$\phi_j\big(d_j(\mathbf{x})\big) = \mathbf{e}^{-\mathbf{d_j(x)}} \tag{1}$$

Neurons with Gaussian RBF present a very selective response, with high activation for patterns close to the radial unit center and very small activation for distant patterns.

2.2. The q-Gaussian Function

In this section we describe some theoretical aspects concerning the q-Gaussian function. It is important to observe that the q-Gaussian is not an alternative to the classic Gaussian function but a parametric generalization of Gaussian function. The main use of the q-Gaussian function is as the probability distribution function that arises naturally when we consider central limit theorem from sum of random variables with global correlations [7]. The use of the q-Gaussian function as a radial basis function in RBF Networks is interesting because it allows to change the shape of the RBF according to the real parameter q [8].

The use of the q-Gaussian function as a radial basis function in RBF Networks is interesting because it allows to change the shape of the RBF according to the real parameter q [8]. The q-Gaussian RBF for the radial unit j can be defined as

$$\phi_j\big(d_j(\mathbf{x})\big) = \mathbf{e}_{\mathbf{q_j}}^{-\mathbf{d_j(x)}} \tag{2}$$

where q_j is a real valued parameter and the q-exponential function of $-d_j(\mathbf{x})$ [9] is given by

$$e_{q_j}^{-d_j(\mathbf{x})} \equiv \begin{cases} \frac{1}{\left(1+(q_j-1)d_j(\mathbf{x})\right)^{\frac{1}{\mathbf{q_j}-1}}} & 1+(q_j-1)d_j \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

An interesting property of the q-Gaussian function is that it can reproduce different RBFs for different values of the real parameter q. For large negative numbers, the function is concentrated around the center of the RBF. When the value of q increases, the tail of the function becomes larger.

In the next, Eq. 3 will be analysed for $q \to 1$, $q = 2$ and $q = 3$. For simplicity, the index j and the dependence on $\mathbf{x}$ will be ommitted in the following equations. For $q \to 1$, the limit of the q-Gaussian RBF can be computed

$$\lim_{q\to 1} e_q^{-d} = \lim_{q\to 1} \frac{1}{\big(1+(q-1)d\big)^{\frac{1}{q-1}}} \tag{4}$$

If we write $z = (q-1)d$, then

$$\lim_{q\to 1} e_q^{-d} = \lim_{z\to 0} \big(1+z\big)^{-\frac{d}{z}}$$

$$\lim_{q\to 1} e_q^{-d} = \lim_{z\to 0} \Big(\big(1+z\big)^{\frac{1}{z}}\Big)^{-d} \tag{5}$$

The limit of the function $(1+z)^{\frac{1}{z}}$ is well know and converges to e when $z \to 0$. Thus,

$$\lim_{q\to 1} e_q^{-d} = e^{-d} \tag{6}$$

In this way, we can observe that the q-Gaussian RBF (Eq. 3) reduces to the standard Gaussian RBF (Eq. 1) when $q \to 1$.

Replacing $q = 2$ in Eq. 3, we have

$$e_q^{-d} = \frac{1}{1+d} \tag{7}$$

i.e., the q-Gaussian RBF (Eq. 3) is equal to the Cauchy RBF for $q = 2$.

When $q = 3$, we have

$$e_q^{-d} = \frac{1}{\big(1+2d\big)^{1/2}} \tag{8}$$

i.e., the activation of a radial unit with an Inverse Multiquadratic RBF for d is equal to the activation of a radial unit with a q-Gaussian RBF (Eq. 3) for $d/2$.

Figure 1 presents the radial unit activation for the Gaussian, Cauchy, and Inverse Multiquadratic RBFs. The activation for the q-Gaussian RBF for different values of q is still presented. One can observe that the q-Gaussian reproduces the Gaussian, Cauchy, and Inverse Multiquadratic RBFs for $q \to 1$, $q = 2$, and $q = 3$. Another interesting property of the q-Gaussian RBF is still presented in Figure 1: a small change in the value of q represents a smooth modification on the shape of the RBF.

In the next section, a methodology to optimize the RBF parameters of the hidden units in RBF Networks via Genetic Algorithms is presented.

2.3. Selection of Parameters via Genetic Algorithms

In the proposed methodology, a Genetic Algorithm (GA) is used to define the number of radial units m, and the parameters of each RBF related to each hidden unit $j = 1, \ldots, m$, i.e., the width, and parameter q for each radial unit with q-Gaussian RBF.

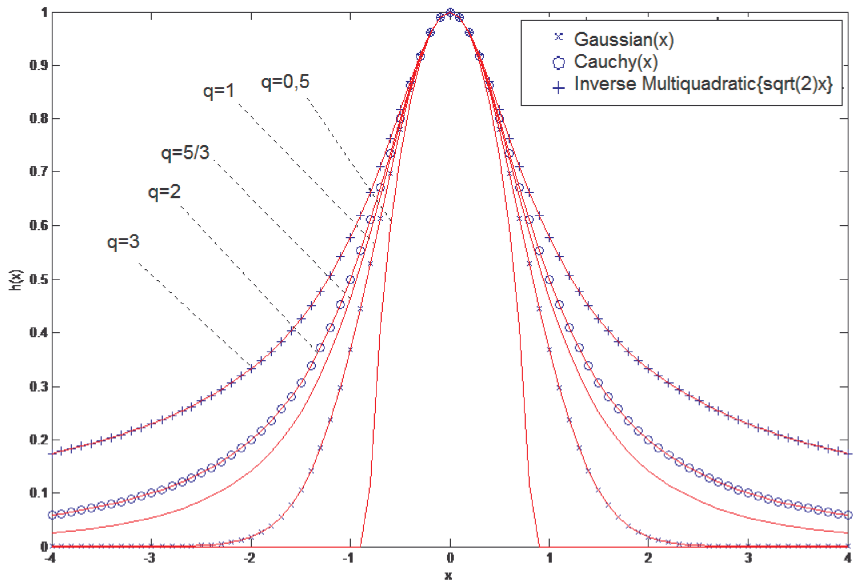

Figure 1. Radial unit activation in an one-dimensional space with $c = 0$ and $r = 1$ for different RBFs: Gaussian ('x'), Cauchy ('o'), Inverse Multiquadratic for $\sqrt{(}2)x$ ('+'), and q-Gaussian RBF with different values of q (solid lines).

2.4. Codification

A hybrid codification (bynary and real) is employed in the GA used in this work. Each individualt i ($i = 1, \ldots, \mu$) is described by a vector (chromosome) with $2N$ elements, where N is the size of the training set. The individual i is defined by the vector

$$\mathbf{z_i}^{\mathrm{T}} = [\ r_1 \quad q_1 \quad r_2 \quad q_2 \quad \ldots \quad r_N \quad q_N\ ] \tag{9}$$

The width and q-parameter respectivelly given by the real numbers r_j and q_j.

2.5. Selection

Tournament selection and elitism are employed here. Elitism is employed in order to preserve the best individuals of the population. Tournament selection is an interesting alternative to the use of fitness-proportionate selection mainly to reduce the problem of premature convergence and the computational cost [10].

2.6. Crossover

When the standard crossover is applied, parts of the chromosomes of two individuals are exchanged. Two point crossover is applied in each pair of new individuals with crossover rate p_c. In this way, individuals exchange all the parameters of a radial unit each time.

2.7. Mutation

Two types of mutation are employed. The standard flip mutation is employed with mutation rate p_m in the elements $\mathbf{z_i}$. When an element r_j or q_j is mutated, its value g_j is changed according to

$$\tilde{g}_j = g_j \exp\left(\tau_m \mathcal{N}(/, \infty)\right) \tag{10}$$

where τ_m denotes the standard deviation of the Gaussian distribution with zero mean employed to generate the random deviation $\mathcal{N}(/, \infty)$.

2.8. Training, validation, and test

For the network training it was used two groups: the training dataset and the validation one, where the last was used for network output error calculation. Both groups were created with 2.000 patterns. The selection of this patterns was done by calculating the Euclidian distance (Eq. 11) between one random pattern and all other patterns in the signal. If the distance is greater than established threshold, the pattern is accepted in the dataset. This procedure is done in order to select the most distinct patterns as possibly.

$$d(p_a, p_b) = \sum_i [p_a(i) - p_b(i)]^2 \tag{11}$$

For each multivariate signal it was trained one different RBF Network. The parameters q and r were randomly initialized between $[0.1, 1.0]$ and $[0.8, 2.8]$, respectively.

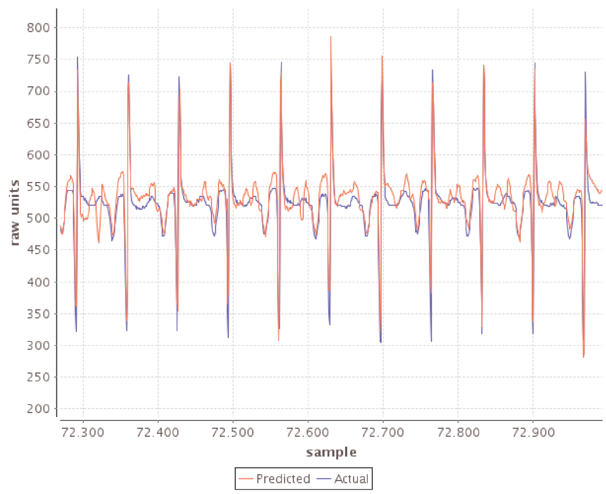

Figure 2. Typical restoration using the proposed method.

3. Results

It was obtained a total score of 25.52 for event 1 and 55.91 for event 2 in dataset ”C”. An example of RBF network restoration can be seen in Figure 2. It’s worth to note that due to the need of large number of generations in GA, the training stage is slow and results cannot be obtained in real time.

4. Discussion and conclusions

This solution for signal restoration is based on a novel method of prediction and classification process, being applicable to real signal monitoring. Despite this novelty aspect, results have shown its good performance for most of the data. As mentioned before, low scores are obtained, in general, for signals which channel with gap is of low quality.

Taking into account the artificial intelligence techniques employed, this method emerges as novel solution for signal restoration problem. This still need to be evaluated in a larger database and other contexts in order to explore the potentiality of this method.

Acknowledgements

The authors thank to FAPESP (grant 2006/00723-9) for financial support.

References

[1] Orr M. Introduction to radial basis function networks. Center for Cognitive Science, Edinburgh University, Scotland, U. K., 1996.

[2] Tinos R, Murta LO. Foundations of Computational Intelligence, volume 5; 127–145.

[3] Moody J, Darken C. Fast learning in networks of locally-tuned processing units. Neural Computation 1989;1:281 – 294.

[4] Billings S, Zheng G. Radial basis function network configuration using genetic algorithms. Neural Networks 1995; 8(6):877–890.

[5] Tinós R, Terra MH. Fault detection and isolation in robotic manipulators using a multilayer perceptron and a rbf network trained by kohonen’s self-organizing map. Rev Controle e Automacc ao 2001;12(1):11–18.

[6] Billings S, Wei HL, Balikhin MA. Generalized multiscale radial basis function networks. Neural Networks 2007; 20:1081–1094.

[7] Umarov S, Tsallis C, Steinberg S. On a q-central limit theorem consistent with nonextensive statistical mechanics. Milan Journal of Mathematic 2008;DOI - 10.1007/s00032-008-0087-y.

[8] Yamano T. Some properties of q-logarithm and q-exponential functions in tsallis statistics. Physica A 2002; 305:486–496.

[9] Tsallis C. What are the number that experiments provide? Quimica Nova 1994;17:468.

[10] Mitchell M. An Introduction to Genetic Algorithms. MIT Press, 1996.

Address for correspondence:

Luiz Otavio Murta Jr
Av. Bandeirantes, 3900, Ribeirão Preto, São Paulo
CEP 14040-41
Brazil
Email: murta@usp.br