ELSEVIER

Neurocomputing 19 (1998) 259–283

NEUROCOMPUTING

Hierarchical RBF networks and local parameters estimate

Nunzio Alberto Borghese[a,*], Stefano Ferrari[a,b]

[a] *Istituto Neuroscienze Bioimmagini - C.N.R., Laboratory of Human Motion Study and Virtual Reality, via f.lli Cervi 93, 20090 Segrate (Milano), Italy*

[b] *Computer Science Department, University of Milano and Bioengineering Centre, Fondazione ProJuventute, Bioengineering Department, Politecnico of Milano, Italy*

Received 12 January 1997; accepted 30 November 1997

Abstract

The method presented here is aimed to a direct fast setting of the parameters of a RBF network for function approximation. It is based on a hierarchical gridding of the input space; additional layers of Gaussians at lower scales are added where the residual error is higher. The number of the Gaussians of each layer and their variance are computed from considerations grounded in the linear filtering theory. The weight of each Gaussian is estimated through a maximum a posteriori estimate carried out locally on a sub-set of the data points. The method shows a high accuracy in the reconstruction, it can deal with non-evenly spaced data points and can be fully parallelizable. Results on the reconstruction of both synthetic and real data are presented and discussed. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: Hierarchical structure; Gridding; Maximum a-posteriori estimate; Linear filtering theory

1. Introduction

The reliable reconstruction of a continuous function from a set of sampled points is one of the key problems in the domain of neural networks to which RBF networks [4,18,27,31] offer an appealing solution. Thanks to their locality, they are less affected by collinearity of the error with respect to the hidden units activity and may converge faster to the optimum solution [4]. From the theoretical point of view, given a sufficient number of hidden units, they have been shown to be able to reconstruct

* Corresponding author. E-mail: borghese@inb.mi.cnr.it

PII S0925-2312(97)00094-5

a continuous function at any degree of approximation [23], provided that their parameters are appropriately set. Whenever Gaussian functions are adopted a RBF network can be expressed by the following form:

$$z = s(\boldsymbol{x}) = \sum_{k=1}^{M} \boldsymbol{w}_k g(\boldsymbol{x}; \boldsymbol{c}_k; \Sigma_k), \tag{1}$$

where the following parameters are considered: the number M and the position $\{\boldsymbol{c}_k\}$ of the Gaussians, their covariance $\{\Sigma_k\}$ and the coefficients $\{\boldsymbol{w}_k\}$ (cf. Fig. 1). Thanks to the structure of Eq. (1), which is a linear combination of non-linear quasi-local functions, the computation of RBF network parameters is suitable to different approaches.

Poggio and Girosi reframed the learning problem in the regularisation domain: given a set of data points and a "smoothness" constraint, a cost function to be minimised can be constructed [10,27] (cf. also [36]). To this clean formulation, which describes, in fact, an optimal control problem, it does not correspond a computability algorithmic solution (cf. [3,19,33]). The use of stochastic gradient [10] suffers from local minima which may prevent to achieve an even acceptable solution, and complex specialised algorithms should be used to get to the global minimum (cf. [1,6,12]). Alternatively, the solution can be searched by a global optimisation with respect to all the parameters using for example genetic algorithms [2], learning automata [20] or simulated annealing [14] which, although theoretically capable to achieve the optimal

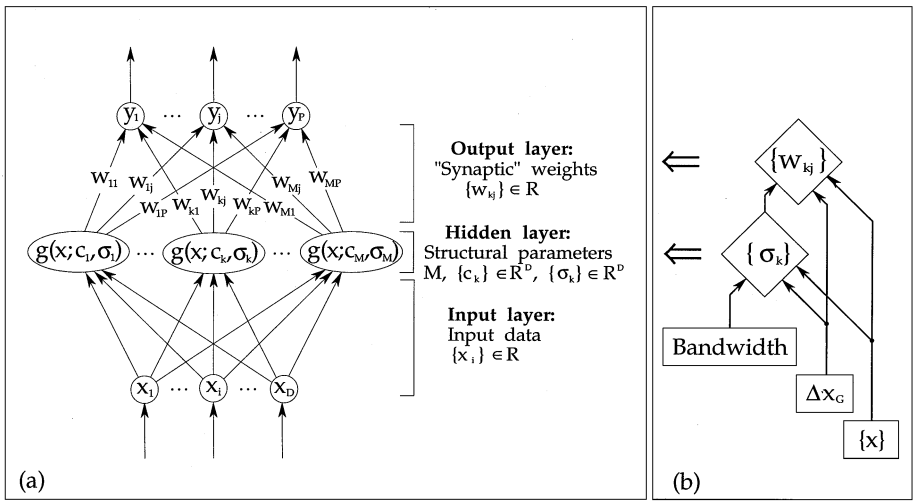


Fig. 1. A RBF Network is constituted of an input layer and two processing layers (a). The first processing layer, the hidden layer, is where the Gaussians are placed and it constitutes the skeleton of the network. The second processing layer, the output layer, contains the "synaptic" weights. In the hybrid learning schemas, adequate learning algorithms are adopted for each of these two layers (b). For the first layer, the variance of the Gaussians is estimated starting from the input data set, the sampling rate, and eventually the bandwidth of the function to be reconstructed. In the second layer, only the weights have to be set, and they can be computed through a linear estimate.

solution, are based an exhaustive search in the solution space and require extensive computation.

A different strategy stems from the observation that Eq. (1) represents essentially a perceptron with non-linear input units of the Gaussian type. Accordingly, the parameters can be subdivided into two sets: the parameters which affect the Gaussian units which will be termed *structural parameters*: (M, $\{\boldsymbol{c}_k\}$ and $\{\Sigma_k\}$) and the parameters "above" the Gaussians, which will be termed "*synaptic weights*:" $\{w_k\}$ (cf. Fig. 1a). This suggests to use different algorithms for each of the two sets of parameters originating what are called hybrid learning schemata [4,18]. In these approaches, the number of the Gaussian units, M, is usually given *a priori* and their position, $\{\boldsymbol{c}_k\}$, is determined through algorithms for optimal clustering [15,17]. The parameters $\{\Sigma_k\}$ are critical as they determine the behaviour of the function in between the samples; and they can be set according to some heuristical considerations [18,21,25,32]. Once the structural parameters have been set, Eq. (1) describes a linear system where the only unknowns are the weights, $\{w_k\}$. Although these could be computed by directly solving the system, this solution can cause both numerical and memory allocation problems for very large networks, and a local schema for the computation of the weights may be preferred. An improvement in the hybrid learning schemata is represented by growing structures [8,9,25], where the number of the Gaussians (and, in general, of the units) in the network is not given *a priori*, but one Gaussian is inserted after the other until a certain criterion has been met. All these solutions are iterative and require an extensive learning to set the parameters to a reasonable value.

An alternative approach, which has its roots in computer vision and optimal control, relies on gridding the input space through a set of equally spaced Gaussians [24,31]. The drawback of this approach lies in the rigidity of the adopted structure which provides a single variance for all the units: the result may be overfitting $s(\boldsymbol{x})$ in some regions while in others some of the finest details can be lost. Moreover, in the regions where overfitting occurs, there will be an excess of hidden units with a waste of resources.

We present here a method, called hierarchical radial basis functions (HRBF), which combines growing structures and linear filtering theory to achieve a stable very fast determination of both the structural parameters and the network weights. First, a simple criterion to set the value of $\{\Sigma_k\}$ is derived; the procedure to determine the weights value directly from a sub-set of the data points is then outlined along with the procedure to allocate incrementally the number of Gaussian units. The methodology is illustrated by simulations and by the reconstruction of an acoustic signal.

2. RBF networks as a Gaussian filter: setting a proper value for σ

When the Gaussians are placed at the crossings of a regular grid, the observation that Eq. (1) is linear in the weights $\{w_k\}$ suggests to analyse the RBF network as an analogical low-pass (Gaussian) filter. For the sake of simplicity, the analysis will be carried out in the one-dimensional space ($P = D = 1$). Moreover, the Gaussians are

supposed to have the same value of $\boldsymbol{\sigma}$; in this case, Eq. (1) is simplified as

$$s(x) = \sum_{k=1}^{N} w_k g(x; c_k; \boldsymbol{\sigma}) = \sum_{k=1}^{N} w_k \frac{e^{-((x-c_k)^2)/\sigma^2}}{\sqrt{\pi}\boldsymbol{\sigma}}. \tag{2}$$

The results are general and can be extended to multi-dimensional spaces by observing that multi-dimensional Gaussians are obtained factorising 1-D Gaussians. The analysis carried out here gives indications on how to set the value for $\boldsymbol{\sigma}$ and the reader not interested can skip this section and take as granted Eqs. (25) and (26).

2.1. The Gaussian filter

An analogical linear filter can reconstruct a continuous function from a set of equally spaced samples. A convenient representation of this kind of filter is through the Fourier transform which describes the filter as a superposition of sinusoids of different frequencies and phases. Each sinusoid is weighted with a coefficient which represents its amplitude and the ensemble of these coefficients constitute the spectrum amplitude of the filter. For the ideal low-pass filter (bold line in Fig. 2a) all the coefficients are equal to 1 for the frequencies contained inside the spectrum of $s(x)$ (Pass Band), and 0 elsewhere (Stop Band). Unfortunately, such a filter is not physically realisable and some approximation has to be accepted: the two Bands are defined through two thresholds: the Pass Band is defined as the interval in which the frequency content amplitude of the filter $F(v)$ is bounded between $[\boldsymbol{\delta}_1, 1]$

$$\delta_1 \leq |F(v)| \leq 1, \quad 0 \leq v \leq v_{\text{cut-off}} \tag{3a}$$

and the Stop Band as the interval in which it is bounded between $[0, \boldsymbol{\delta}_2]$

$$0 \leq |F(v)| \leq \delta_2, \quad v_{\max} \leq v < +\infty. \tag{3b}$$

A third Band (Transition Band), for which the frequencies are progressively attenuated, is also defined (cf. Fig. 2a). We apply these considerations to the Gaussian filter

$$g(x; \boldsymbol{\sigma}) = \frac{1}{\sqrt{\pi}\boldsymbol{\sigma}} e^{-x^2/\sigma^2} \tag{4}$$

which assumes in the frequency domain the following form[1]

$$G(v; \boldsymbol{\sigma}) = \mathcal{F}(g(x; \boldsymbol{\sigma})) = e^{-\pi^2\sigma^2 v^2}, \tag{5}$$

[1] We explicitly observe that a normalised version is here adopted (Eq. (4)) to obtain $|G(v; \boldsymbol{\sigma})| = 1$ at least for the DC component ($v = 0$). This normalised version has unitary norm

$$\frac{1}{\sqrt{\pi}\boldsymbol{\sigma}} \int_{-\infty}^{+\infty} e \frac{x^2}{\sigma^2} dx = 1.$$

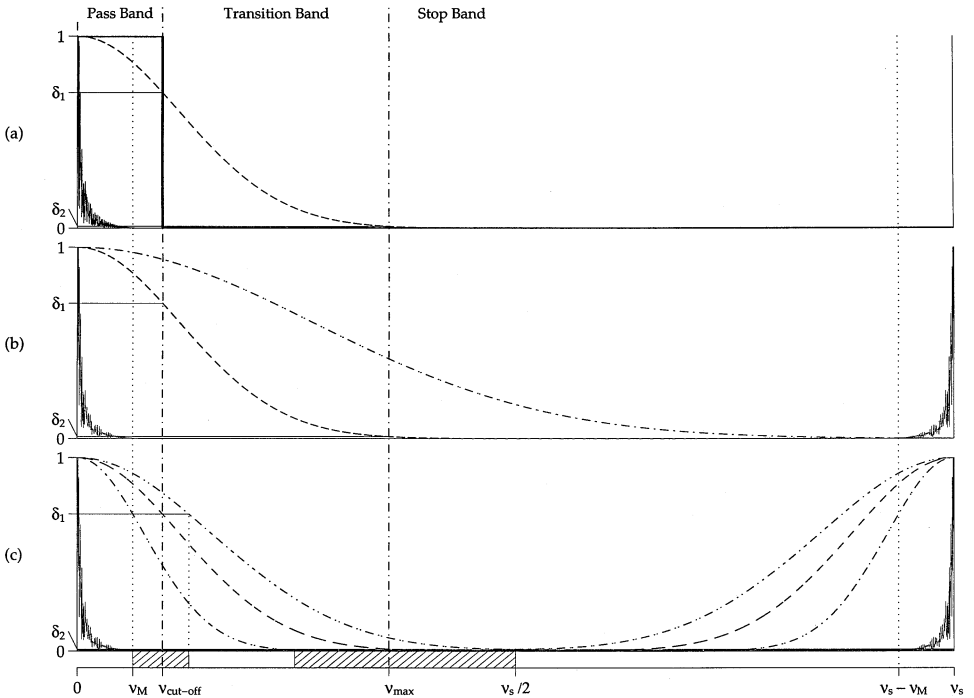


Fig. 2. The spectrum amplitude of the Gaussian is reported in dashed line along with the conditions on the cut-off and maximum allowed frequencies in three different conditions: continuous case (a) $\nu_{\text{cut-off}} > \nu_M$; discrete sampling with an infinite number of data points (b) $\nu_{\text{cut-off}} > \nu_M$ and $\nu_{\max} < \nu_s - \nu_M$; discrete sampling with a finite number of data points in a compact set (c) $\nu_{\text{cut-off}} > \nu_M$ and $\nu_{\max} < \nu_s/2$. The vertical lines denote the limits of the three Bands in which the frequency axis is subdivided (cf. Section 2.1). δ_1 and δ_2 are the tolerance levels used to determine the Bands. The spectrum amplitude of the ideal Low Pass filter is reported in bold line in (a).

where $\mathcal{F}(\,.\,)$ indicates the Fourier transform. The monotonicity of $G(\nu;\boldsymbol{\sigma})$ allows to relate the values of $\nu_{\text{cut-off}}$ and $\nu_{\max}$ to $\boldsymbol{\sigma}$ as

$$\begin{aligned} e^{-\pi^2\sigma^2\nu_{\text{cut-off}}^2} &= \delta_1 \\ e^{-\pi^2\sigma^2\nu_{\max}^2} &= \delta_2 \end{aligned} \Rightarrow \begin{cases} \nu_{\text{cut-off}} = \dfrac{\sqrt{-\ln\delta_1}}{\pi\boldsymbol{\sigma}}, \\ \nu_{\max} = \dfrac{\sqrt{-\ln\delta_2}}{\pi\boldsymbol{\sigma}}. \end{cases} \tag{6}$$

2.2. Continuous case

We will first apply the above concepts to the analysis of a continuous form of RBF Gaussian network (cf. [21]). When the distance between two consecutive points becomes vanishingly small ($(c_{k+1} - c_k) \to 0$), Eq. (2) becomes

$$s(x) = \int_R w(c)g((x-c);\boldsymbol{\sigma})\,\mathrm{d}c \tag{7}$$

which suggests the following statement:

Statement 1. *Let* $w(x)$, $s(x)$ *and* $g(x;\boldsymbol{\sigma}) \in L_1(R)$,[2] *then the formulation in Eq.* (7) (*continuous RBF Network*), *is equivalent to the convolution of the function* $w(x)$ *with the Gaussian function* $(x;\boldsymbol{\sigma})$. *That is*[3]

$$s(x) = \int_R w(c)g((x-c);\boldsymbol{\sigma})\,\mathrm{d}c = w(x)^* g(x;\boldsymbol{\sigma}). \tag{8}$$

For the convolution theorem, the following relationship holds:

$$\mathcal{F}(s(x)) = \mathcal{F}(w(x))\mathcal{F}(g(x;\boldsymbol{\sigma})) \Rightarrow S(\boldsymbol{v}) = W(\boldsymbol{v})G(\boldsymbol{v};\boldsymbol{\sigma}), \tag{9}$$

where $S(\boldsymbol{v})$, $W(\boldsymbol{v})$ and $G(v;\boldsymbol{\sigma})$ are the Fourier transforms, respectively, of $s(x)$, $w(x)$ and $g(x;\boldsymbol{\sigma})$. Substituting $s(x)$ to $w(x)$ in Eq. (8), we get

$$\tilde{s}(x) = \int_R s(c)g((x-c);\boldsymbol{\sigma})\,\mathrm{d}c = s(x)^* g(x;\boldsymbol{\sigma}) \tag{10}$$

and, in the frequency domain,

$$\tilde{S}(v) = S(v)G(v;\boldsymbol{\sigma}). \tag{11}$$

Eq. (10) has the same structure of Eq. (8) and it is a convenient representation of $s(x)$ because the value of the weights in a certain point is equal to the value of the function itself in the same point, and it has not to be computed or learned. We will here examine how to choose $\boldsymbol{\sigma}$ in order to get $\tilde{s}(x)$ close enough to $s(x)$.

By examining Eq. (3a), we notice that the width of the Pass Band of the Gaussian is regulated by $v_{\text{cut-off}}$. Therefore, it should hold

$$v_{\text{cut-off}} > v_{\text{M}}\,, \tag{12}$$

where v_{M} is the maximum frequency constituting $s(x)$.

2.3. Discrete case

When the distance between two Gaussians is finite, one more condition has to be introduced to assure that the reconstructed function is sufficiently close to $\tilde{s}(x)$. Let us hypothesise that the function $s(x)$ has been sampled into a set of data points, equally

[2] L_1 is the Lebesgue vectorial space; when $f(P) \in L_1(T)$, it follows that it can be integrated in absolute value: $\int_T |f(p)|\,\mathrm{d}P < +\infty$.

[3] This statement can be quite general and it applies to all the functions which belong to $L_1(R)$; among these are some of the functions generally proposed as bases for RBF Networks [3] which monotonically decrease to 0 when $x \to \infty$ (e.g. $\phi(\,.\,) = \mathrm{e}^{|x-c|}/\sigma$, $\phi(\,.\,) = \sin^2(x-c)/(x-c)^2$). It should be remarked with [24] that the convolution does not give a finite result and the Fourier transform is not defined for other basis functions common in connectionist literature like polynomials, logistic functions or multiquadric which do not belong to $L_1(R)$.

spaced by Δx_P: $S = \{s_k = s(x_k, y_k) \mid k \in Z\}$, coincident with the position of the Gaussian centres. Under this condition, Eq. (2) becomes

$$s(x) = \sum_{k=-\infty}^{+\infty} w_k g(x; x_k; \boldsymbol{\sigma}) = \frac{1}{\sqrt{\pi}\boldsymbol{\sigma}} \sum_{k=-\infty}^{+\infty} w_k e^{-(x-x_k)^2/\sigma^2}. \tag{13}$$

As in the continuous case, Eq. (13) suggests the following statement:

Statement 2. *Let* $S = \{s_1 = s(x_1, y_1), s_2 = s(x_2, y_2), s_3 = s(x_3, y_3), \ldots, s_k = s(x_k, y_k)\}$ *be a set of points equally spaced on* $R(x_{k+1} - x_k = \Delta x_P \forall k)$ and $\{g(x; x_k; \boldsymbol{\sigma})\}$ *be a set of normalised Gaussian functions centred in the points* $\{x_k\}$; *the RBF network, Eq.* (13), *is equivalent to the convolution of the series of the weights,* $\{w_k\}$, *with the Gaussian kernel* $g(x; x_k; \boldsymbol{\sigma})$..

Eq. (13) can be transformed in the frequency domain by the convolution theorem, to obtain

$$\mathcal{F}(s(x)) = \mathcal{F}(\{w_k\})\mathcal{F}(g(x; \boldsymbol{\sigma})) \Rightarrow S(v) = \breve{W}(v)G(v; \boldsymbol{\sigma}), \tag{14}$$

where $\breve{W}(v)$ is the Fourier transform of the weights series $\{w_k\}$. By substituting in Eq. (13) the product of the data points $\{s_k\}$ by the sampling interval Δx_P, to the weights $\{w_k\}$, we obtain

$$\tilde{s}(x) = \sum_{k=-\infty}^{+\infty} s_k g((x; x_k; \boldsymbol{\sigma})\Delta x_p = \frac{\Delta x_p}{\sqrt{\pi}\boldsymbol{\sigma}} \sum_{k=-\infty}^{+\infty} s_k e^{-(x-x_k)^2/\sigma^2} \tag{15}$$

and in the frequency domain

$$\tilde{S}(v) = \breve{S}(v)G(v; \boldsymbol{\sigma})\Delta x_p \tag{16}$$

which is similar in the formulation to Eq. (11). $\breve{S}(v)$ is the Fourier transform of the series of the data points $\{s_k\}$ and it is a periodical function of period $v_S = 1/\Delta x_P$ (cf. Fig. 2b) and Δx_P is a normalisation factor required to recover the true amplitude of $s(x)$ [22]. As in the continuous case, $v_{cut\text{-}off} > v_M$, is required to make $\tilde{s}(x)$ close to $s(x)$ but it is not sufficient. In fact, to avoid the introduction of spurious high-frequency components in $\tilde{s}(x)$, the spectrum of the Gaussian should not overlap with the replicas of the spectrum of $s(x)$ [22]. This is not possible as the support of $G(v; \boldsymbol{\sigma})$ is unbounded; but with the approximation in Eq. (3b), the following relationship can be derived (cf. Fig. 2b)

$$v_{max} < v_S - v_M \tag{17}$$

A subtle phenomenon, which further constraints v_{max}, occurs when the data points do not constitute a series but are in finite number, N, equally spaced in a compact set $[x_1, x_N]$: $\{S\} = \{s_1 = s(x_1, y_1), \ s_2 = s(x_2, y_2), \ s_3 = s(x_3, y_3), \ldots, s_N = s(x_N, y_N)\}$, $x_k - x_{k-1} = \Delta x_p$, $k = 2, \ldots, N$. In this case, Eq. (16) will be discrete

$$\tilde{S}(v_k) = \hat{S}(v_k)\hat{G}(v_k; \boldsymbol{\sigma}), \tag{18}$$

where $\hat{S}(v_k)$ is discrete with its samples equally spaced by $\Delta v_{s_k} = v_s/N$ (cf. Fig. 2c) and periodical of period v_S. We explicitly remark here that although $G(v;\boldsymbol{\sigma})$ is a continuous function, the Fourier transform of the reconstructed function, $\tilde{S}(v_k)$, is a sequence of samples equally spaced in frequency by Δv_{s_k}. The frequencies, $\bar{v}$, which are in between two frequency samples ($v_k < \bar{v} < v_{k+1}$) of $G(v;\boldsymbol{\sigma})$, are lost. This is equivalent to use a discrete version of the spectrum of the Gaussian, $\hat{G}(v_k;\boldsymbol{\sigma})$, where v_k are the same frequencies constituting $\hat{S}(v_k)$. $\hat{G}(v_k;\boldsymbol{\sigma})$ will therefore be discrete and it can be obtained as the Fourier transform of a sampled version of the original Gaussian function on a compact support. It will be itself periodical of period v_S. This introduces a further constraint over v_{max} (cf. Fig. 2c)

$$v_{max} < \frac{v_S}{2}, \tag{19}$$

which is more stringent than Eq. (17) and will be used in the following.

From Eq. (18), the discrete sequence of the reconstructed samples of $s(x)$ can be obtained as:

$$\tilde{s}(x_j) = \sum_{k=1}^{N} s_k g((x - x_k);\boldsymbol{\sigma}) \qquad \Delta x_p = \frac{\Delta x_p}{\sqrt{\pi}\boldsymbol{\sigma}} \sum_{k=1}^{N} s_k e^{-(x-x_k)^2/\sigma^2}, \tag{20}$$

which can be extended on a continuous support using

$$\tilde{s}(x) = \sum_{k=1}^{N} s_k g((x - x_k);\boldsymbol{\sigma}) \qquad \Delta x_p = \frac{\Delta x_p}{\sqrt{\pi}\boldsymbol{\sigma}} \sum_{k=1}^{N} s_k e^{-(x-x_k)^2/\sigma^2}, \tag{21}$$

obtaining a continuous reconstruction of $s(x)$.

2.4. Setting of $\boldsymbol{\sigma}$ and remarks

We now summarise the conditions over $\boldsymbol{\sigma}$. As $\boldsymbol{v}_{max}$ and $\boldsymbol{v}_{cut\text{-}off}$ are both function of $\boldsymbol{\sigma}$, it is convenient to express $\boldsymbol{v}_{max}$ as a function of $\boldsymbol{v}_{cut\text{-}off}$ and vice versa (cf. Eq. (6))

$$\begin{aligned} e^{-\pi^2\sigma^2 v_{cut\text{-}off}^2} &= \delta_1, \\ e^{-\pi^2\sigma^2 v_{max}^2} &= \delta_2/2. \end{aligned} \tag{22}$$

δ_2 has been divided by a factor of two because when $v_{max} = v_S/2$, $\tilde{G}(v_k;\boldsymbol{\sigma})$ receives an equal contribution from the main lobe and from the closest replica (cf. Fig. 2c): it results $\tilde{G}(v_k;\boldsymbol{\sigma}) < \delta_2$ at least when the first replica of $\tilde{G}(v_k;\boldsymbol{\sigma})$ is considered.[4] In the following, we set $\delta_1 = (\sqrt{2}/2)$ which is a common choice in Digital filtering theory and corresponds to a maximum attenuation in the Pass Band of 3 dB [11], and, somehow

[4] Only the fist lobe is considered as the amplitude of the other lobes at $v = v_S/2$ are of the order of 10^{-9} for the second lobe and smaller for the other more distant lobes.

arbitrary, but in a very conservative way, $\delta_2 = 0.01$. From Eq. (22), the following relationships between v_{max} and σ and between $v_{cut\text{-}off}$ and v_{max} are derived:

$$v_{max} = \frac{0.7327}{\sigma}, \tag{23a}$$

$$v_{cut\text{-}off} = 0.2558 v_{max}. \tag{23b}$$

Taking into account Eq. (23b), Eqs. (12) and (19) can be expressed as a single inequality on $v_{cut\text{-}off}$

$$v_M < v_{cut-off} < 0.2558 v_S / 2 \tag{24}$$

or, equivalently, on v_{max}

$$\frac{v_M}{0.2558} < v_{max} < v_S / 2 \tag{25}$$

from which the following inequality on σ is obtained:

$$\sigma_{min} = \frac{1.465}{v_S} = 1.465\Delta x \leq \sigma \leq \frac{0.1874}{v_M} = \sigma_{Max}. \tag{26}$$

As it will be clear from the simulations, the larger is σ, the smoother will be the reconstruction and the more the noise cleaned. On the other side, the smaller is σ, the more the reconstruction will be close to the finest details of $s(x)$.

From Eq. (24) it follows that, given a certain sampling frequency, v_S, the maximum frequency content of $s(x)$, $\bar{v}_M$, should be

$$\bar{v}_M = 0.1279 v_S. \tag{27}$$

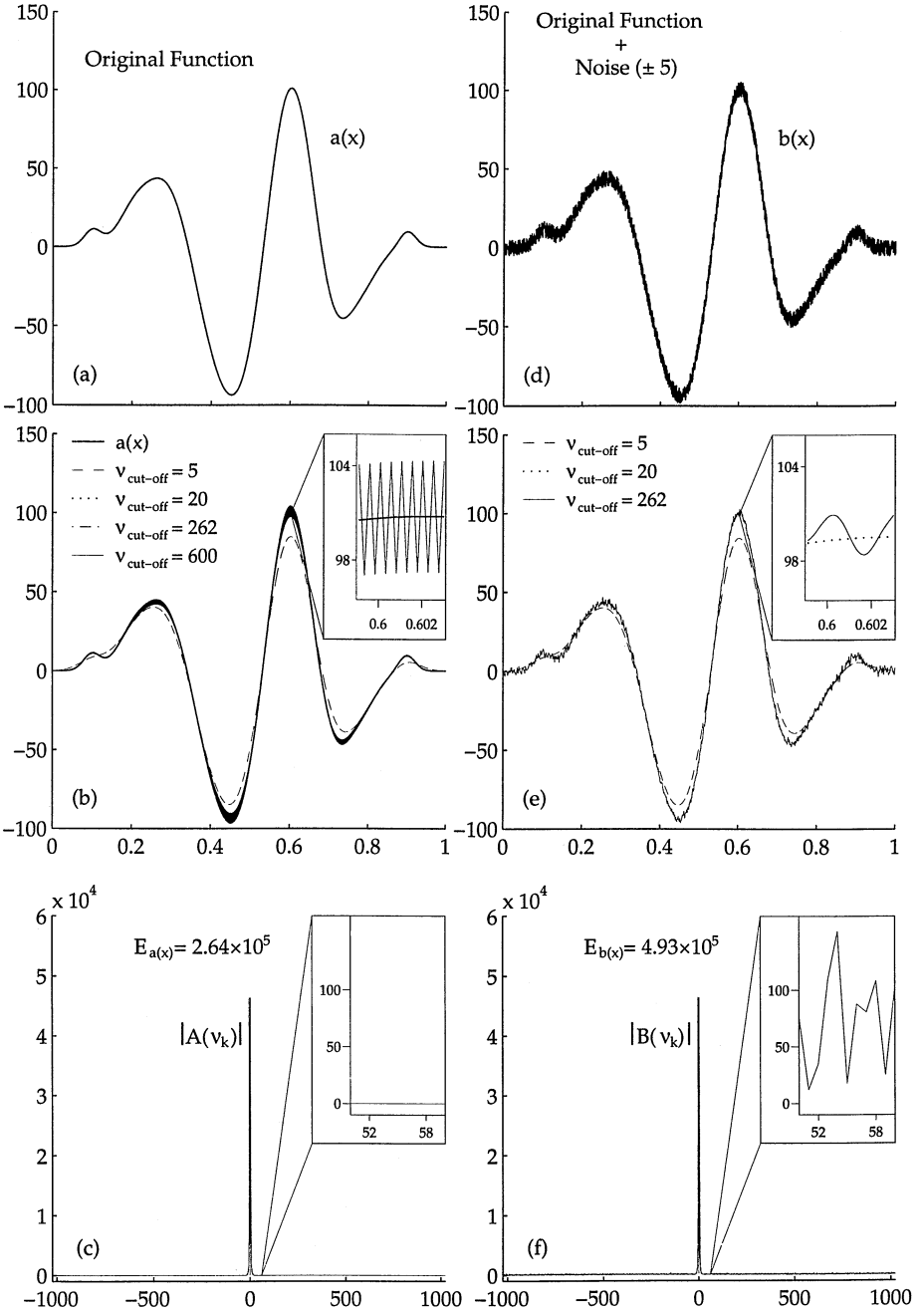
Eq. (27) can be seen under another perspective: given the maximum frequency, $\bar{v}_M$, of the function $s(x)$ to be reconstructed, the minimum allowed sampling frequency, v_s, is computed as

$$v_S = 2\frac{\bar{v}_M}{0.2558} \cong 7.8\bar{v}_M. \tag{28}$$

This has a very important meaning: it states that the function $s(x)$ should be oversampled at least by a factor 3.9 with respect to the Shannon theorem.

3. Simulations with $N=M$

Let us apply the considerations on the choice of σ to the function $a(x)$ reported in Fig. 3a. This has been obtained as a linear combination of nine equally spaced Gaussians with parameters: $\{c_k\} = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$; $\{\sigma_k\} = \{0.03, 0.05, 0.1, 0.1, 0.085, 0.1, 0.08, 0.05, 0.03\}$ and $\{w_k\} = \{10, 10, 60, -70, -85, 140, -80, -10, 10\}$. A learning set has been formed sampling $a(x)$ into $N = 2048$ equally spaced data points ($\Delta x_p = \frac{1}{2048}$, $v_S = 2048$ Hz). The maximum frequency content of


Original Function
a(x)
(a)
Original Function
+
Noise (± 5)
b(x)
(d)
a(x)
$\nu_{cut-off} = 5$
$\nu_{cut-off} = 20$
$\nu_{cut-off} = 262$
$\nu_{cut-off} = 600$
(b)
$\nu_{cut-off} = 5$
$\nu_{cut-off} = 20$
$\nu_{cut-off} = 262$
(e)
x 10^4
$E_{a(x)} = 2.64 \times 10^5$
$|A(\nu_k)|$
(c)
x 10^4
$E_{b(x)} = 4.93 \times 10^5$
$|B(\nu_k)|$
(f)

$a(x)$ has been computed as the frequency for which the energy drops under 0.001% and it is equal to 20 Hz. From Eq. (24), the cut-off frequency should be inside

$$20 \text{ Hz} \leq \nu_{\text{cut-off}} \leq 262 \text{ Hz}. \tag{29}$$

The function $a(x)$ is reconstructed through a RBF network constituted of $M = 2048$ Gaussian units each centred in one data point. As suggested by Eq. (21), the weights $\{w_k\}$ are set at the value assumed by the function $a(x)$ in the points $\{x_k\}$

$$\tilde{a}(x) = \frac{\Delta x_p}{\sqrt{\pi\sigma}} \sum_{k=1}^{M} a(x_k) g((x - x_k); \sigma). \tag{30}$$

The quality of the reconstruction will be quantitatively assessed by the mean square reconstruction error (mse) computed over a test set $\{T\}$ of $T = 16\,384$ points taken equally spaced by $\Delta x_t = \frac{1}{16\,384}$. As can be seen in Fig. 3b, the reconstructions obtained with $\nu_{\text{cut-off}} = 20$ Hz and $\nu_{\text{cut-off}} = 262$ Hz cannot be distinguished one from the other and from the true profile. For $\nu_{\text{cut-off}} < \nu_M$, $\tilde{a}(x)$ is not able to closely replicate the peaks and the valleys of $a(x)$ (dashed line in Fig. 3b, where $\nu_{\text{cut-off}} = 5$ Hz) and for $\nu_{\text{cut-off}} > \nu_S/2$, $\tilde{a}(x)$ will be affected by aliasing which introduces high-frequency oscillations (cf. continuous line in the box, where the reconstruction with $\nu_{\text{cut-off}} = 600$ Hz is reported). The quantitative analysis of the error measured on the test set (Table 1), shows that the larger is the value of $\nu_{\text{cut-off}}$, the smaller is the error, down to 262 Hz above which the error starts to increase again. Therefore, in the noise free situation, the best choice would be to choose $\nu_{\text{cut-off}}$ equal to $0.1279 * \nu_S = 262$ Hz.

Table 1
Mean-square error in the reconstruction of the function $a(x)$ of Fig. 3a using $M = 2048$ Gaussians and $N = 2048$ data points, equally spaced

σ	$\nu_{\text{cut-off}}$	mse No noise	mse Noise $\pm$ 2.5	mse Noise $\pm$ 5
0.0375	5	32.77	33.07	33.39
9.37×10^{-2}	20	0.163	0.223	0.346
1.874×10^{-3}	100	0.266×10^{-3}	0.194	0.774
0.937×10^{-3}	200	1.661×10^{-5}	0.401	1.604
0.715×10^{-3}	262	5.66×210^{-6}	0.529	2.116
0.468×10^{-3}	400	5.063×10^{-5}	0.818	3.273
0.313×10^{-3}	600	1.199	2.439	6.161

Fig. 3. The reconstruction of a synthetic function through a RBF network. The original function is reported in (a). Different reconstructions featuring different values of $\nu_{\text{cut-off}}$ are reported in (b). The effect of aliasing on the reconstruction is magnified in the box where the high frequency oscillations introduced by using Gaussians with $\nu_{\text{cut-off}} = 600$ Hz can be appreciated. The same functions are reported in (d–e) for $b(x)$, obtained adding uniform random noise of $\pm$ 5 to $a(x)$. The two reconstructions obtained with $\nu_{\text{cut-off}} = 20$ and 262 Hz are magnified in the box of (e); the cleaning up of the noise obtained by using a small $\nu_{\text{cut-off}}$ is evident. The frequency contents of $a(x)$ and $b(x)$ are plotted in (c), the spectrum of $b(x)$ contains frequencies upto $\nu_S/2$ and its energy content is 87% larger than $a(x)$.

When the data points $\{a_k\}$ cannot be measured with absolute accuracy, different considerations apply. Let us add uniform random noise $\{n_k\}$ to the points $\{a_k\}$; the sequence $\{b_k\}$ is obtained

$$\{b_k\} = \{a_k\} + \{n_k\}. \tag{31}$$

This presents high-frequency oscillations, as can be seen in Fig. 3d. If $\tilde{a}(x)$ were reconstructed using the maximum allowed cut-off frequency, ($v_{\text{cut-off}} = 262$ Hz), there would have been very little filtering of the noise (Fig. 3e, continuous line). Under the connectionist perspective, this can be considered an overfitted version of $a(x)$. A much better result can be obtained if $v_{\text{cut-off}}$ is reduced to v_M ($v_{\text{cut-off}} = 20$ Hz); most of the noise has been cleaned up as it is evident in the zoom box of Fig. 3e. Similarly to the no noise case, some of the finest details of $a(x)$ are lost in the reconstruction, when $v_{\text{cut-off}} < v_M$ (dashed line in Fig. 3e).

4. Simulations with $M < N$

In the previous case, the density of Gaussians is higher than necessary. In fact, for Eq. (28) the sampling frequency, $\bar{v}_S$, can be reduced to

$$\bar{v}_S = \frac{v_M}{0.1279} = \frac{20}{0.1279} = 156.4 \tag{32}$$

and the number of Gaussians can therefore be reduced from $M = 2048$ to $M = 156$ which is a clear advantage in terms of resources allocated. Let us see what happens when $M = 204$ Gaussians, one out of every ten data points, are used. From Eq. (24), $\nu_{\text{cut-off}}$ is reduced into the small interval [20, 26.2] Hz. The same data and test sets used in Section 3 are used here and the quantitative results are reported in Table 2.

A degradation is evident in the accuracy of the reconstruction when noise is present with an increase in the mse of one order of magnitude. The direct substitution of s_k to

Table 2
Mean-square error in the reconstruction of the function $a(x)$ of Fig. 3a using $M = 204$ Gaussians and $N = 2048$ data points, equally spaced

		Direct computation of the weights		Weighted average of neighbour points
		mse	mse	mse
σ	$v_{\text{cut-off}}$	No noise	Noise $\pm$ 5	Noise $\pm$ 5
0.0375	5	32.77	32.14	34.06
1.874×10^{-2}	10	2.461	2.739	2.971
0.937×10^{-2}	20	0.163	1.151	0.418
0.715×10^{-2}	26.2	0.056	1.544	0.293
0.313×10^{-2}	60	1.233	5.763	1.519

w_k is not a good solution in this case. On the other side, only those 204 data points which are coincident with the Gaussians centres have been used, while all others ($N = 2048 - 204 = 1844$) have been discarded. A more efficient schema for the computation of the weights has therefore been devised.

5. Local estimation of the weights

Let $s_k = s(x_k)$ be the value of the function $s(x)$ in the point x_k. A first observation is that if $s(x)$ is a continuous smooth function, the value of $s(x)$ in the neighbourhood, S_k, of x_k should be close to s_k. On the limit

$$\lim_{x \to x_k} s(x) = s_k. \tag{33}$$

Taking this into account, it is natural to estimate s_k as the average, $\bar{s}_k$, of all the data points belonging to S_k weighted by a quantity which is proportional to their distance from x_k

$$\bar{s} = \frac{\sum_{r=1}^{R} s_r w(|x_r - x_k|)}{\sum_{r=1}^{R} w(|x_r - x_k|)}, \tag{34}$$

where R is the number of data points in S_k. Among the possible choices of weighting functions, $w(\,.\,)$, the Gaussian allows to obtain the maximum a posteriori estimate [10,32] of $s(x_k)$, from the subset S_k. Eq. (34) therefore becomes

$$\bar{s}_k = \frac{\sum_{r=1}^{R} s_r e^{((x_r - x_k)^2)/\sigma_w^2}}{\sum_{r=1}^{R} e^{((x_r - x_k)^2)/\sigma_w^2}}, \tag{35}$$

where $\boldsymbol{\sigma}_w$ is a scale parameter, set to $\boldsymbol{\sigma}_w = \boldsymbol{\sigma}_{\text{cut-off}}/2$ to avoid any additional filtering on the function $s(x)$.

The neighbourhood region, S_k, has been defined here as the interval centred in x_k of amplitude $\pm \Delta x_G$, where Δx_G is the distance between two consecutive Gaussians. This has the rationale that a Gaussian is heavily responsible for the region around its centre (this implicitly defines a receptive field for the Gaussian). Therefore, for each $g(\,.\,)$ in Eq. (30), the neighbourhood data set, S_k, is defined as

$$S_k = \{x_m\} : |x_m - x_k| < \Delta x_G\}. \tag{36}$$

As can be seen from the third column of Table 2, with this approach, the reconstruction error, obtained with $M = 204$ Gaussians, when the estimate of the $\{\bar{s}_k\}$ through Eq. (35) is substituted to the $\{w_k\}$, has been reduced to the level obtained by using 2048 Gaussian units and close to the levels with no noise.

The power of this schema for the computation of the weights lies in the fact that it can be applied also when the data points are not equally spaced, obtaining the same accuracy. This can be appreciated from Table 3 where the mse is reported for the reconstruction of $a(x)$ carried out with $M = 204$ Gaussians, starting from a set of $N = 2048$ data points randomly sampled from $b(x)$.

Table 3
Mean-square error in the reconstruction of the function $a(x)$ of Fig. 3a using $M = 204$ Gaussians and $N = 2048$ data points randomly sampled. Uniform random noise of ± 5 has been added to their coordinates

$\nu_{\text{cut-off}}$	5	10	20	26.2	60
mse	33.94	2.886	0.430	0.332	1.632

6. Hierarchical approximation

In real applications the frequency content of $s(x)$ can be different throughout the input space and the use of a single scale, $\boldsymbol{\sigma}$, for all the Gaussians can be severely questioned. In fact, to guarantee the reconstruction of the finest details of $s(x)$, $\boldsymbol{\sigma}$ should be chosen according to the highest-frequency content of $s(x)$, over all the input space, also when this is concentrated in a narrow region. This would cause a waste of resources in those regions of space where the scale of $s(x)$ is larger and fewer Gaussians, featuring larger scales, could be used. The solution proposed here to save units is to add to a basic network which contains a Gaussian grid with a large scale, grids of equally spaced Gaussians featuring smaller scales.

The process of creation of such a network is made clear in Fig. 4a. A first approximation of $s(x)$, $a_1(x)$, is obtained using a layer of Gaussians with a very large scale. If information on the local bandwidth of the function $s(x)$ were available, this scale would correspond to the smaller local bandwidth over all the input space. Once the scale of this first layer has been set, the number and the spacing between the units can be computed from Eq. (26). This layer will be able to reconstruct only the low frequency components giving a coarse reconstruction of $s(x)$; and the finest details present in those regions where the highest-frequency components are contained, will be lost. As a result, the residual, $r_1(x) = s(x) - a_1(x)$, will be higher in these regions. To be able to improve the reproduction of these details, a second grid of Gaussians, featuring a smaller scale, is added to the network. This layer will not be complete as the Gaussian units will be inserted only in those regions where $r_1(x)$ is higher than a predefined threshold, ε. This second layer will produce a second residual, $r_2(x)$. If this residual were still too large in some regions of the input domain, a third grid of Gaussians featuring an even smaller scale is created. The procedure is iterated until the residual is smaller than ε over the input space (uniform convergence).

At this point a criterion to analyse the residual is required. The maximum error (L_∞ norm) is not a wise choice as it allows the reproduction of both the noise and the outliers and an integral criterion is preferable. In the following the L_1 norm of the residual measured on the points contained in the receptive field of each Gaussian (defined by Eq. (36)), has been adopted. For the 1-D case, a Gaussian is inserted in the position k of the grid if

$$\frac{\sum_{r=1}^{R} |r_k|}{R} > \varepsilon. \tag{37}$$

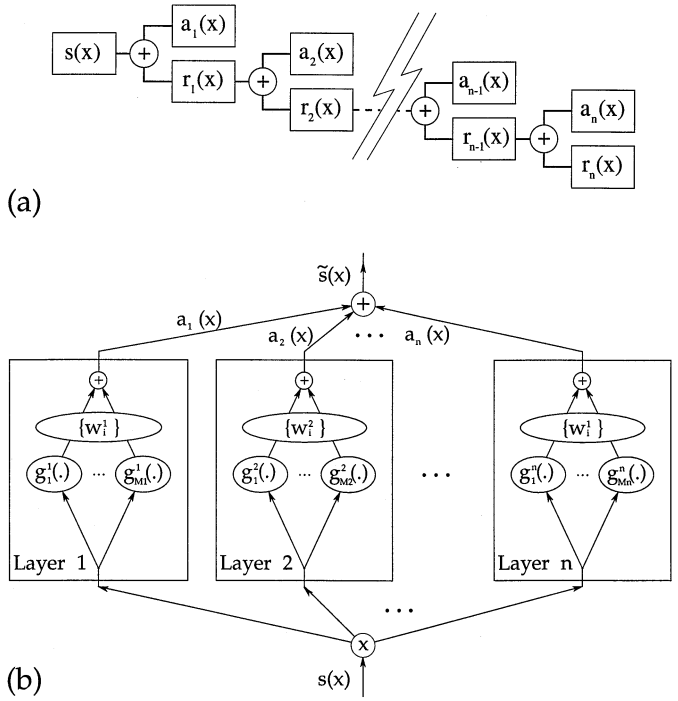


Fig. 4. The hierarchical structure of a RBF network. In the learning stage (a), more layers of Gaussians are added one after the other to approximate the residual of the previous layer. The layers are added until the residual decreases below a certain threshold. In the reconstruction phase (b), the different layers work in parallel on the same input data.

We remark that with this hierarchical structure, a uniform L_1 approximation is achieved, provided that a sufficient number of data points has been sampled.

7. Summary of the learning procedure

For the first layer:

1. A complete grid of Gaussians is created, each featuring the same $\boldsymbol{\sigma}$ ($\boldsymbol{\sigma}_1$) which is chosen to allow the reconstruction of at least the grossest details of $s(x)$.
2. The spacing between two Gaussians, Δx_G, is computed according to Eq. (26). Given the value of Δx_G and the extension of the input space, the number, M_1, and the position of the Gaussians are determined.
3. The weights, $\{w_{1k}\}$, of this first layer are computed through Eq. (35).
4. The output of this first layer, $a_1(x)$ is computed and the residual reconstruction error is determined of all the points $\{x_k\}$ belonging to the input data set as $\{r_1(x_k)\} = \{s(x_k) - a_1(x_k)\}$.

For the higher layers:

5. At each iteration, l, a new grid of Gaussians is inserted. The scale associated to the lth layer, $\boldsymbol{\sigma}_l$, can be computed as $\boldsymbol{\sigma}_l = \boldsymbol{\sigma}_{l-1}/2$.
6. The spacing between two Gaussians, Δx_{G_l}, is computed according to Eq. (26), i.e. it is halved; the number, M_l, and the position of the Gaussians of this layer are determined as in (2).
7. Among the M_l Gaussians, only those which are placed in a region (Eq. (36)) where the residual error is above ε (Eq. (37)), are preserved; the others are discarded. A reduced set of $M'_l \leq M_l$ Gaussians is therefore obtained for these layers.
8. The weights, $\{w_{lk}\}$ of the lth layer are computed through Eq. (35).
9. The output of the lth layer, $a_l(x)$, is computed and the residual reconstruction error is determined as $\{r_l(x_k)\} = \{r_{l-1}(x_k) - a_l(x_k)\}$.
10. The learning procedure stops when the L_1 norm of the residual $r_l(x_k)$ is smaller than ε over all the input domain (uniform approximation).

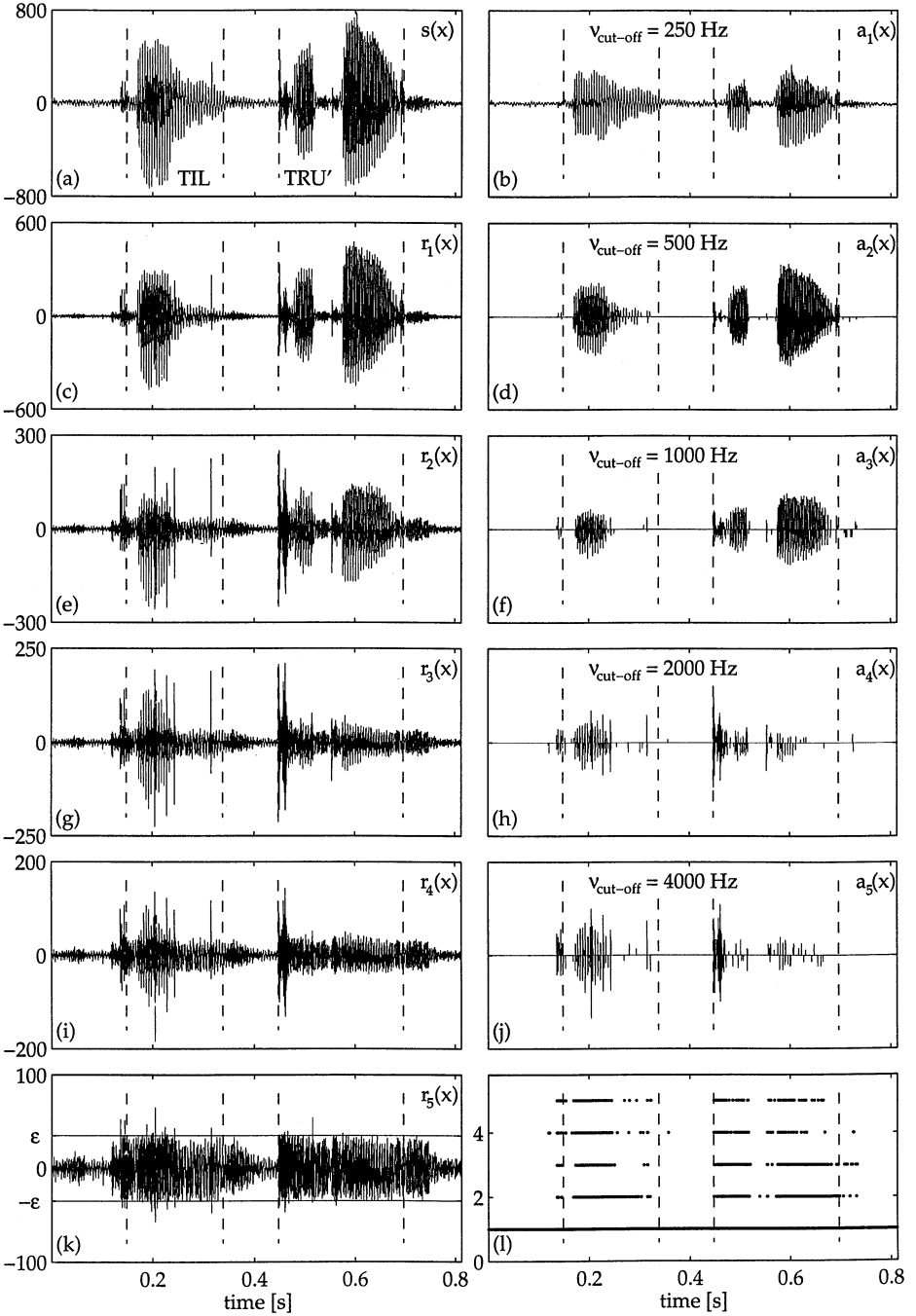
The construction of the network proceeds along with the estimation of the weights, sequentially through the different layers in the learning process. In fact, the residual of the previous layer is required for the construction of the intermediate layers (cf. Fig. 4a). In the reconstruction process, instead, the different layers operate in parallel (cf. Fig. 4b). In fact, each layer receives the same input, namely, the position of a sample in the input space, $\bar{x}$, and outputs a value which is an approximation of the function at the largest scale (first layer), and an approximation of the residual (intermediate layers). The actual approximation of $s(x)$ in the point, $\bar{x}$, $\tilde{s}(\bar{x})$, is obtained by adding up the contributions of all the layers.

8. Reconstruction of a phonetic sequence

This model has been tested on the reconstruction of the acoustic sequence \tiltrù\. Acoustic sequences are interesting as their frequency content is highly time varying. Moreover, they usually present spurious spikes which, from the learning point of view, can be viewed as outliers in the input data set. Finally, the acoustic sequence can also be played back to hear the reconstruction allo wing an effective qualitative evaluation of the result, besides the quantitative one.

The sequence has been sampled at 32 kHz collecting a data set, D, of $N = 26\,000$ samples in a time interval of 0.8125 s (Fig. 5a) and it has been reconstructed continuously by polynomial interpolation. The learning set, S, is constituted of $P = 26\,000$ input data points *randomly* extracted and, therefore, in general, not equally spaced.

Fig. 5. Reconstruction of the phonetic sequence /tiltrù/ through a HRBF network of five layers. The original sequence is reported in (a) and it is constituted of $N = 26\,000$ equally spaced data points ($\nu_s = 32$ kHz). The output of each layer is reported in (b) (d) (f) (h) (j) and the residual at each layer in (c) (e) (g) (i) (k). In (l) the distribution of the Gaussians in each layer is plotted.


800
0
−800
s(x)
(a)
TIL
TRU'
$\nu_{cut-off}$ = 250 Hz
$a_1(x)$
(b)
600
0
−600
$r_1(x)$
(c)
$\nu_{cut-off}$ = 500 Hz
$a_2(x)$
(d)
300
0
−300
$r_2(x)$
(e)
$\nu_{cut-off}$ = 1000 Hz
$a_3(x)$
(f)
250
0
−250
$r_3(x)$
(g)
$\nu_{cut-off}$ = 2000 Hz
$a_4(x)$
(h)
200
0
−200
$r_4(x)$
(i)
$\nu_{cut-off}$ = 4000 Hz
$a_5(x)$
(j)
100
ε
0
−ε
−100
$r_5(x)$
(k)
4
2
0
(l)
0.2
0.4
0.6
0.8
time [s]
0.2
0.4
0.6
0.8
time [s]

Five layers have been used with cut-off frequencies, respectively: $\nu_{\text{cut-off}} = [250, 500, 1000, 2000, 4000]$ Hz. The lowest cut-off frequency (250 Hz) guarantees that the lowest-frequency components of the human speech can be reconstructed; and the higher cut-off frequency (4000 Hz) satisfies the constraint imposed by Eq. (24).

The reconstruction operated by the first layer, $a_1(x)$, is reported in Fig. 5b: only the bulk of the time course of $s(x)$ has been reconstructed and important contributions are lacking as can be seen from the residual, $r_1(x)$ (Fig. 5c). The next intermediate layers are aimed to reconstruct these contributions. In these layers, the Gaussians are inserted only when the residual is over threshold (cf. Section 6) and they are distributed where the phonetic production is concentrated (Fig. 5l) for a total of $M = 4987$ Gaussians. The output of the intermediate layers is reported in Fig. 5d, f, h, j; the residual in Fig. 5e, g, i, k and the reconstruction error in Table 4. The final reconstruction, obtained adding up the contributions of the five layers, is reported in Fig. 6a along with the residual computed over the input data set, D (Fig. 6b). The hearing of the residual reveals that it contains only acoustic noise while the sequence /tiltrù/ reconstructed through HRBF is not distinguishable from the original one: in the reconstruction noise has therefore been cleaned. Another property of the HRBF reconstruction is the filtering of the spurious spikes as can be seen in Fig. 6b. This is achieved both by the filtering property of the MAP estimation of the weights and on the L_1 criterion used for the insertion of new units.

The value of ε is here set automatically by analysing the noise amplitude: under the hypothesis that noise is constant throughout the input sequence, it can be estimated where there is no phonetic production, for example, in the first 100 ms. Three times the standard deviation of the data in this interval is taken as the value for ε; and it is reported as horizontal lines in Fig. 6b and 6d.

These results suggest that HRBF networks can be seen also as a multiresolution representation of the data: the more layers are considered, the more the finest details will be reproduced. Such a representation is extremely useful when $s(x)$ has to be manipulated, transmitted or output with different degree of detail.

As a benchmark, this reconstruction has been compared with that obtained with a complete RBF network constituted of 26 000 equally spaced Gaussian units each centred in one data point ($\Delta x_G = 3.125 \times 10^{-5}$, $\nu_{\text{cut-off}} = 4$ kHz). The data points in this case are not randomly sampled but they constitute the original data set D. The weights assume the value of the sequence in the Gaussian centres (cf. Eq. (21)). The

Table 4
Reconstruction of the sequence \tiltrù\ using the hierarchical approach for the Gaussians allocation. The input data points have been obtained by randomly sampling the continuos acoustic trace

No. of layer	mse on the residual	Global mse	No. of Gaussians	$\nu_{\text{cut-off}}$ (Hz)
1	9185	9362	1588/1588	250
2	1016	1067	1059/3176	500
3	415.1	475.4	1012/6352	1000
4	264.8	316.1	634/12704	2000
5	185.7	225.9	694/25408	4000

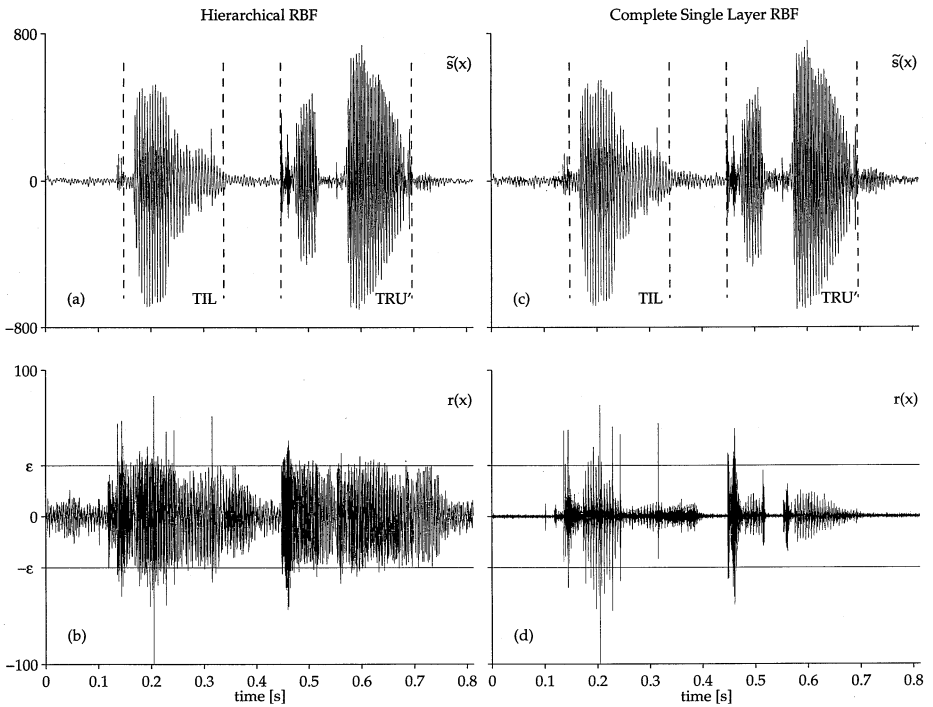


Fig. 6. Benchmark of the reconstruction of the phonetic sequence /tiltrù/. In (a), the reconstruction has been carried out through the HRBF, and in (c), it has been carried out using a complete network of equally spaced Gaussians. The residual of the two reconstructions are reported, respectively, in (b) and (d); the horizontal line represents the residual error ε defined by Eq. (37).

obtained reconstruction is plotted in Fig. 6c along with its residual (Fig. 6d): it is evident that the quantity of noise cleaned up by HRBF network is much higher than by a full network. From the point of view of the allocated resources, it is clear that the Gaussian units inserted where there is no speech at all or where the sound is at lower frequency like in the oscillations associated to /ù/ and /l/, constitute a waste of resources. In the HRBF network, most of the units are allocated where the phonetic production is concentrated allowing to save 21 013 units. Moreover, with the HRBF network, it has been possible to reconstruct the phonetic sequence also if the acoustic samples were not equally spaced.

9. Discussion

The approach presented here belongs to the *two-steps* methods in which the computation of the RBF network parameters is decomposed into two steps (cf. Eq. (1) and Fig. 1). First, the structural parameters of the network, namely, the number of the

Gaussians, their variance and the position of their centres, are determined. In a second step, only the value of the weights is estimated.

9.1. Structural parameters

9.1.1. Gaussians positioning

The methods adopted to compute the Gaussians position in a RBF network can be grouped into two large families: *data driven* and *error driven*. In the data-driven methods [4,18] a predefined number of Gaussians is distributed in the input space according to the local density of the data points, operating a vector quantization of the input space. The underlying assumption is that more points are sampled in the regions where the function is more difficult to be reconstructed, because it is more rapidly varying or, equivalently, its bandwidth is larger; which is not always the case. To decouple the dependence of the density of the Gaussians from the density of the input data, methods which are *error driven* have been proposed [5,7,8,25,31]. These are essentially incremental and insert one Gaussian after the other following criteria based on the residual reconstruction error. Here the number of the Gaussian units does not need to be fixed a priori but it is the result of the learning process. Nevertheless, they are iterative and require extensive learning to set the parameters to a reasonable value. Moreover, they may easily not give an optimal solution as the insertion of one Gaussian modifies the network structure only locally (e.g. three units are modified in [8]).

A reduction in the computational demand is obtained covering the input space with a regular grid of Gaussian units [4,31]. In this approach the location of the hidden units is determined once the spacing between two adjacent units, Δx, has been fixed, the only care is to choose Δx small enough to reconstruct the finest details of the input function (cf. Eq. (27)). The strength of this approach is to suggest a tool to directly set the value of the variance for the hidden units (cf. Sections 2 and 9.1.2), the weakness is that more units than necessary will be allocated when the finest details are concentrated only in few sub-regions of the input space, resulting in a waste of resources. In this approach, the distribution of the Gaussians cannot be considered either related to the input data (data driven) or to the reconstruction error (error driven).

The HRBF approach takes advantage of the strength of the gridding approach and avoids the waste of resources. The key operation is the transformation of the hidden layer into a set of hierarchical layers, each with a characteristic scale; the largest scale being attributed to the first layer and the smallest to the last one. The first layer outputs a reconstruction of the function at a very coarse scale while the Gaussians in the other layers are used to approximate the *local* residual error, constituted of the details that the previous layers were not able to reconstruct. With this construction, the gridding procedure has been transformed into an error-driven procedure: as can be seen in Fig. 6b, the residual error obtained at the end of the learning will be under a predefined threshold (uniform L_1 convergence). An approach similar to HRBF has been proposed by Fritzke using Kohonen maps [9]. The main difference is that here it

is not required that an entire row or column be inserted into the grid, but single shorter segments are allowed on demand.

9.1.2. Estimation of the value of σ

Using equally spaced Gaussians, a single value of σ is adopted for each layer. Although small variations in its value do not change dramatically the mse ([16,32], cf. Table 1), if σ is set outside an adequate range, high distortions in the reconstructed function do arise (cf. Fig. 3b, [22]): a too large value of σ produces a low-pass filtered reconstruction; on the other side, σ cannot be decreased ad libitum as a too small value will lead to a spiky reconstruction. This problem is known as the trade-off between bias and variability in the statistics literature [28]. Sanner and Slotine [31] have provided an analytical procedure to bound the residual error as a function of the frequency content of $s(x)$. This criterion has two drawbacks, it is very conservative and it is based on the L_∞ norm which does not allow the elimination of the outliers. We have preferred here to adopt a constructive criterion: units at smaller scales are inserted where the residual error is high until a uniform (L_1) convergence is achieved. In this case, the choice of the value of σ becomes less critical provided that it is large enough. An empirical criterion to set the value of σ ($\sigma = \Delta x_G$) is given in the domain of Parzen Window estimate [32,35]; it has been shown in Section 3 that this criterion does not guarantee a smooth reconstruction and a more conservative value of $\sigma = 1.465\Delta x_G$ is suggested (Eq. (26)). The difference is significant in terms of amount of overlap between two consecutive Gaussians which increases from 68.2% to 73.3%.

9.2. Weights computation

Once the number, the position and the variance of the Gaussians of one layer have been determined, the structure of that layer has been completed and the "synaptic" weights can be determined. Their optimal value can be computed solving the linear system in Eq. (2) using, for example, the LMS algorithm [25,35] or techniques which are numerically more stable like singular-value decomposition [29]. A better schema which allows to eliminate the outliers has been recently proposed [30]. However, these solutions are computationally demanding and may cause numerical and memory allocation problems for large networks. For these reasons a local computation schema has been devised here. This is based on the observation that the computation carried out in a RBF network constituted of equally spaced units with the same σ is mathematically equivalent to the convolution of the input data set with a Gaussian kernel (cf. Statements 1 and 2). In this condition, the value of the weights can be assumed equal to the value of the function in the centre of each Gaussian obtaining a good reconstruction. A more reliable estimation of the weights can be achieved through a maximum a posteriori estimate which considers all the data points inside the receptive field of each Gaussian unit (Eq. (35), cf. [10]). This makes the overall structure more powerful allowing to approximate a function also when the points are not equally spaced as it happens in the most common neural network problems. Moreover, this estimate, along with the L_1 criterion of insertion of the units, allows to filter out the outliers as can be appreciated in the residual of Fig. 6b where most of the

spikes have not been reproduced by the network. From the computational point of view the learning procedure requires to carry out as many small estimate problems as the number of the Gaussians. This scalability allows the implementation of the algorithm on a parallel hardware whichever is the cardinality of the input data set.

9.3. General remarks

The Gaussian cannot be considered an optimal low-pass filter: as its Transition Band is large (cf. Fig. 2a) a function has to be heavily oversampled (about 3.9 times, Eq. (28)) to obtain a good reconstruction. This turns out to be an advantage for the estimation of the weights as about fifteen data points will be used in the estimate (these are the points which fall inside the receptive field of each Gaussian, cf. Eqs. (35) and (36)). Although different functions have been proposed to interpolate in between the grid crossings (e.g., splines, optimal analogical filters, to which the hierarchical structure can be applied as well), the Gaussian functions are preferred for two main reasons: they are claimed to constitute a processing module common in the human nervous system [26] and they have a straightforward simple implementation in parallel hardware [28] which makes them particularly attractive for real-time network implementations.

RBF networks have been proposed also as the solution of a regularisation problem [10,27]. In fact, the recovery of the function $s(x)$ from a finite set of samples $\{s(x_k)\}$ is an ill-posed problem because the solution is not unique. Constraints on the differential characteristic of the function $s(\,.\,)$ are introduced which transform the problem into a minimisation of a cost function constituted of two terms: the first term penalises deviation from the input data and the second one from smoothness

$$\tilde{s}(x) = \min_{s(x)} H(s) = \sum_{k=1}^{N} (s(x_k) - s_k)^2 + \lambda \int \sum_{m=0}^{+\infty} c_m |\nabla^m s(\,.\,)|^2, \tag{38}$$

where ∇ is the Laplacian operator on $s(\,.\,)$. The actual shape of $\tilde{s}(x)$ depends strongly on the value of the $\{c_m\}$. In particular, when $c_m = \sigma^{2m}/(m!2^m)$, the regulariser assumes the following shape:

$$\lambda \int \sum_{m=0}^{+\infty} c_m |\nabla^m s(\,.\,)|^2 = \lambda \int S(v) S^*(v) e^{v^2}\, dv \tag{39}$$

and the solution has been shown to be a sum of Gaussians centred in $\{x_k\}$ with variance σ^2 [36]. From Eq. (39) it is evident that an increase of both λ and σ, reduces the high-frequency components in $\tilde{s}(x)$ increasing the low-pass filtering capability of the network. It can therefore be questioned the utility of this additional parameter λ and smoothing here is left only to the Gaussian variance of the Gaussians. HRBF networks can be also seen as a special case of the mixture of experts models [13] where each hidden layer can be seen as a sub-network specialised in a certain frequency range. The complex learning machinery of the mixture of experts is substituted by an incremental construction of the layers.

The reconstruction obtained with this approach has the great advantage to be very fast and direct although it is not claimed to be optimal. Nevertheless, it may constitute a very good starting point for iterative methods, like stochastic gradient [27], which eventually converge to the optimal solution when starting from a point sufficiently close to the true one [34].

10. Conclusions

The HRBF network presented here allows to obtain a fast and accurate reconstruction of any continuous function starting from not equally spaced data sets. The incremental architecture allows to achieve a uniform approximation without wasting resources. Provided that the function is heavily oversampled, a good estimate of the weights can be achieved. The complexity of the learning algorithm does not increase with the cardinality of the input data set: the algorithm can be fully parallelisable and of possible implementation in real-time on a parallel machine.

Remark. The synthetic and real data used for this work are available through anonymous ftp at carla.inb.mi.cnr.it/pub/hier_RBF/data.

Acknowledgements

We wish to thank Dr. D. Liberati and Dr. G. Ferrigno for the valuable discussions on linear filtering theory.

References

[1] A.B. Berg, Locating global minima in optimisation problems by a random-cost approach, Nature 361 (1993) 708.
[2] S.A. Billings, G.L. Zheng, Radial basis function network configuration using genetic algorithms, Neural Networks 8 (6) (1995) 877.
[3] N.A. Borghese, M. Arbib, Generation of temporal sequences using local dynamic programming, Neural Networks 8 (1) (1995) 39.
[4] D.S. Broomhead, D. Lowe, Multivariable functional interpolation and adaptive networks, Complex Systems 2 (1988) 321.
[5] M. Cannon, J.E. Slotine, Space-frequency localized basis function networks for nonlinear system estimation and control, Neurocomput. 9 (1995) 293.
[6] B.C. Cetin, J. Barhen, J.W. Burdick, Terminal repeller unconstrained subenergy tunneling (trust) for fast global optimization, J. Optim. Theory Appl. 77 (1) (1993) 97.
[7] S. Chen, C.F. Cowan, P.M. Grant, Orthogonal least squares learning algorithm for radial basis function networks, IEEE Trans. Neural Networks 2 (2) (1991) 302.
[8] B. Fritzke, Growing cell structures – a self-organizing network for unsupervised and supervised learning, Neural Networks 7 (9) (1994) 1441.
[9] B. Fritzke, Growing grid, a self-organizing network with constant neighborhood range and adaptation strength, Neural Process. Lett. 2 (5) (1995) 1.

[10] F. Girosi, M. Jones, T. Poggio, Regularization theory and neural networks architectures, Neural Comput. 7 (1995) 219.
[11] R.C. Gonzales, R.E. Woods, Digital Image Processing, Addison-Wesley, Reading, MA, 1993.
[12] D. Gorse, A. Shepherd, J.G. Taylor, Avoiding local minima by a classical range expansion algorithm, Proc. ICANN 94, June 1994.
[13] R.A. Jacobs, M. Jordan, S.J. Nowlan, G.E. Hinton, Adaptive mixtures of local experts, Neural Comput. 3 (1991) 79.
[14] S. Kirkpatrick, C. Gelatt, M. Vecchi, Optimization by simulated annealing, Science 220 (1983) 671.
[15] S.P. Lloyd, Least squares quantization in PCM, IEEE Trans. Inform. Theory 28 (1982).
[16] S. Marchini, N.A. Borghese, Optimal local estimation of RBF parameters, Proc. of ICAN94, June 1994.
[17] T.M. Martinetz, S.G. Berkovich, K.J. Schulten, Neural-gas network for vector quantization and its application to time-series prediction, IEEE Trans. Neural Networks 4 (4) (1993) 558.
[18] J. Moody, C. Darken, Fast-learning in networks of locally-tuned processing units, Neural Comput. 1 (2) (1989) 281.
[19] K.S. Narendra, K. Parthasarathy, Gradient methods for the optimization of dynamical systems containing neural networks, IEEE Trans. Neural Networks 2 (2) (1991) 252.
[20] K.S. Narendra, M.A. Thathachar, Learning Automata – An Introduction, Prentice-Hall, Englewood Cliffs, NJ, 1989.
[21] M.J.L. Orr, Regularization in the selection of radial basis function centres, Neural Comput. 7 (3) (1995) 606.
[22] A.V. Oppenheim, R.W. Schafer, Digital Signal Processing, Prentice-Hall, Englewood Cliffs, NJ, USA.
[23] J. Park, I.W. Sandberg, Universal approximation using radial-basis-function networks, Neural Comput. 3 (2) (1991) 246.
[24] P. Perona, J. Malik, Scale-space and edge detection using anisotropic diffusion, IEEE Trans. Pattern Anals. Mach. Intell. 12 (7) (1990) 629.
[25] J. Platt, A Resource-allocating network for function interpolation, Neural Comput. 3 (1991) 213.
[26] T. Poggio, A theory of how the brain might work, Cold Spring Harbor Symp. Quantitative Biology, 1990, pp. 899–910.
[27] T. Poggio, F. Girosi, Regularization algorithms for learning that are equivalent to multilayer networks, Science 247 (1990) 978.
[28] T. Poggio, V. Torre, C. Koch, Computational vision and regularization theory, Nature 317 (1985) 314.
[29] W.H. Press, W.T. Vetterling, S.A. Teukolsky, B.P. Flannery, Numerical Recipes in C, Cambridge University Press, Cambridge, 1992.
[30] D.V. Sanchez, Robustization of a learning method for RBF networks, Neurocomput. 9 (1995) 85.
[31] R.M. Sanner, J.E. Slotine, Gaussian networks for direct adaptive control, IEEE Trans. Neural Networks 3 (6) (1992) 837.
[32] D.F. Specht, Probabilistic neural networks, Neural Networks 3 (1990) 109.
[33] C.J. Watkins, P. Dayan, Q-Learning, Machine Learning 8 (1992) 279.
[34] L. Wessels, E. Barnard, Avoiding false local minima by proper initialization of connections, IEEE Trans. Neural Networks 3 (6) (1992) 899.
[35] L. Xu, A. Krzyzak, A. Yuille, On radial basis function nets and kernel regression: statistical consistency, convergence rates, and receptive field size, Neural Networks 7 (4) (1994) 609.
[36] A.L. Yuille, N.M. Grzywacz, A computational theory for the perception of coherent visual motion, Nature 333 (1988) 71.

Nunzio Alberto Borghese received the "laurea" in Electrical Engineering from Politecnico of Milano, Italy, with 100/100 cum laude in 1986. He worked for two years after laurea at the Center for Bioengineering of Politecnico of Milano before joining the Institute of Neuroscience and Bioimages of CNR where he is currently Director of the Laboratory of Human Motion Study and Virtual Reality. In 1991–1992 he spent one sabbatical year at the Centre for Neural Engineering of USC, Los Angeles, CA and at the Department of Electrical Engineering, Caltech, Los Angeles, Pasadena, CA. His research interests include quantitative human motion analysis and modelling, and artificial learning systems. He is a member of IEEE and of ENNS.

Stefano Ferrari received the "laurea" in Computer Science from Università degli Studi of Milan, Italy in 1995. Currently, he is pursuing graduate studies at the Department of Electrical Engineering of Politecnico of Milano. His research interests are image processing and neural networks models.