Random Sieve Based on Projections for RBF Neural Net Structure Selection

Ewa Skubalska-Rafajłowicz and Ewaryst Rafajłowicz

Institute of Computer Engineering control and Robotics,
Wrocław University of Technology, Wrocław , Poland
ewa.rafajlowicz@pwr.wroc.pl

Abstract. Our aim is to propose a method for selecting a radial basis functions terms to be included into a neural net model. As it is frequently met in practice, we consider the case of a deficit in the admissible number of observations (learning sequence) in comparison with a much larger number of candidate terms. The proposed approach is based on a random sieve that aims at selecting only necessary RBF's by a hierarchy of a large number of random mixing of candidate RBF's and testing their significance. The results of simulations are also reported.

Keywords: random projections, RBF, neural nets, model selection.

1 Introduction

RBF neural nets have been the subject of intensive research for many years. We refer the reader to a selected collection of more recent papers and monographs [8], [5], [12], [15], [4], [10] [6], [7], [20], [21], [22], [32], where references to earlier works can be found.

In opposite to the present paper, most of the proposed methods require more observations than candidate terms. Here, we admit a much larger initial size of an RBF net than the length of a learning sequence. One can wonder how it can be possible to select terms in such cases. The idea is based on random projections of a part of the RBF's and considering them as one term with randomly selected parameters (or a random mixture of RBF's). Then, we repeatedly test the validity of such a mixture of terms, repeating also their random projections.

There are some relationships of our approach with group testing [13] for selecting a regression function terms (see [14] for a survey of group testing approaches). Notice, however, that in [13] the grouping is done according to values of terms, while here we propose grouping by their random mixing.

Methods that are based on penalizing too many terms, such as AIC, BIC, C_p, GIC as well as cross-validation or bootstrap (see [9] for these and other criteria) either require candidate nets to be nested or lead to the need of comparing all the subsets of candidate terms. With the exception of so called forward term selection (as done in regression function estimation) they are not applicable in our case of a larger initial net structure than the length of a learning sequence.

L. Rutkowski et al. (Eds.): ICAISC 2013, Part I, LNAI 7894, pp. 193–204, 2013.

An aspect – rarely considered in the literature – is a choice of inputs in a learning sequence for a net structure selection. The exceptions in this respect include: [1], [17] and the bibliography cited there in.

Random projections have proved their usefulness in solving many highly dimensional problems (see [27], [28], [29], where also the references to the probabilistic background of random projections can be found).

Simultaneously with this paper in [26] we have proposed a method for selecting an optimum experiment design when a random projections method is used for selecting terms in a regression function estimation. A method sketched briefly in [26] differs from the one presented here in several respects:

1. the algorithm presented here is dedicated for RBF nets,
2. it is improved in comparison to the one in [26] by adding preliminary reduction of the net structure, which leads to more efficient use of a learning sequence,
3. it can be used not only for selecting proper RBF's to be introduced to a net, but also for the choice of independent variables.

2 Problem Statement

For simplicity of the exposition we consider the following version of a RBF net:

$$y(\mathbf{x}) = \sum_{j=1}^{r} a_j\, Ker(||\mathbf{x} - \mathbf{C}_j||/h_1) + \sum_{k=1}^{\widetilde{K}} b_k\, Ker(||\mathbf{x} - \mathbf{c}_k||/h_2), \qquad (1)$$

where $\mathbf{x} \in \mathbf{R}^d$ is a vector of the net inputs, $y(\mathbf{x})$ is its output (univariate for simplicity of the exposition), while $Ker : \mathbf{R}^1 \to \mathbf{R}^1$ is a nonnegative kernel function such that $\int_{-\infty}^{\infty} Ker(t)\, dt = 1$, $\int_{-\infty}^{\infty} t\, Ker(t)\, dt = 0$, $\int_{-\infty}^{\infty} t^2\, Ker(t)\, dt < \infty$, the Gaussian kernel being the most popular. In (1) the RBF net is split into two parts. The first one has centers at points $\mathbf{C}_j \in \mathbf{R}^d$, weights $\mathbf{C}_j$, $j = 1, 2, \ldots, r$ and smoothing parameter $h_1 > 0$. This part plays a special role, because we consider it as a part of an RBF net that is expected to be present in the final RBF net structure. In applications this part may represent a general trend, while the second summand in (1) is intended to model more subtle details. For this reason, we usually select the number of neurons in this part $\widetilde{K}$ much larger than r, which is the number of terms in the first part. Consequently, RBF centers $\mathbf{c}_k \in \mathbf{R}^d$, $k = 1, 2, \ldots, \widetilde{K}$ are placed more densely than centers $\mathbf{C}_j$'s, while the smoothing parameter h_2 should be smaller than $h - 1$ in order to better approximate fine details.

We assume that we have a learning sequence $(\mathbf{x}_i,\, y_i)$, $i = 1, 2, \ldots, n$ at our disposal, where $\mathbf{x}_i \in \mathbf{R}^d$'s are input vectors, while y_i's are observed outputs of a certain surface or a system that our RBF net is expected to approximate. In order to tune (1) to approximate y_i's by $y(\mathbf{x}_i)$'s we have to choose weights a_i's. We also have to select proper terms in the second part of (1) and tune the corresponding weights. In more detail, our aim is to find

$$\sum_{m=1}^{K} b_{k(m)}\, Ker(||\mathbf{x} - \mathbf{c}_{k(m)}||/h_2), \tag{2}$$

where K is much smaller than $\widetilde{K}$ and a sequence of indexes $k(m)$, $m=1, 2, \ldots, K$, which a subsequence of all indexes $k = 1, 2, \ldots, \widetilde{K}$. In other words, our aim is to select a sub-net of (1) of the form:

$$y(\mathbf{x}) = \sum_{j=1}^{r} a_j\, Ker(||\mathbf{x} - \mathbf{C}_j||/h_1) + \sum_{m=1}^{K} b_{k(m)}\, Ker(||\mathbf{x} - \mathbf{c}_{k(m)}||/h_2) \tag{3}$$

and to tune its parameters in such a way that $\sum_{i=1}^{n}(y_i - y(\mathbf{x}_i))^2$ is minimized.

In our problem statement $0 \le K < \widetilde{K}$ is also a decision variable. In order to ensure the possibility of estimating a_j, $j = 1, 2, \ldots, r$ and $b_{k(m)}$, $m = 1, 2, \ldots, K$ we have to confine to K such that $r + K \le n$.

Our task is difficult, because of our assumption that the length n of the learning sequence is much smaller than $\widetilde{K}$. This assumption implies that we must admit errors in selecting a structure of our RBF net.

We leave outside the scope of this paper the problems of proper selection of smoothing parameters $0 < h_2 \le h_1$ assuming that they are reasonably chosen. We refer the reader to [8], [32], and the bibliography cited therein for methods of selecting smoothing parameters. Concerning the choice of centers $\mathbf{c}_k$'s, $\mathbf{C}_j$'s positions notice that the approach proposed here contains implicitly a way of selecting centers positions $\mathbf{c}_{k(m)}$, $m = 1, 2, \ldots, K$ from a much larger collection $\mathbf{c}_k$, $k = 1, 2, \ldots, \widetilde{K}$. After selecting them, one can adjust their positions as well as positions of $\mathbf{C}_j$'s using more traditional methods that are well suited for a fine positions adjustment of a relatively small number of RBF centers.

It is convenient to introduce a shorthand notations:

N1) for the first sub-net $\mathbf{v}(\mathbf{x}) = [v_1(\mathbf{x}), v_2(\mathbf{x}), \ldots, v_r(\mathbf{x})]^T$, where T denotes the transposition, $v_j(\mathbf{x}) \overset{def}{=} Ker(||\mathbf{x} - \mathbf{C}_j||/h_1)$, $j = 1, 2, \ldots, r$ and $\mathbf{a} = [a_1, a_2, \ldots, a_r]^T$,

N2) for the second sub-net $\mathbf{w}(\mathbf{x}) = [w_1(\mathbf{x}), w_2(\mathbf{x}), \ldots, w_{\widetilde{K}}]^T$, $w_k(\mathbf{x}) \overset{def}{=} Ker(||\mathbf{x} - \mathbf{c}_k||/h_2)$, $k = 1, 2, \ldots, \widetilde{K}$.

Note that $\mathbf{v} : \mathbf{R}^d \to \mathbf{R}^r$ and $\mathbf{w} : \mathbf{R}^d \to \mathbf{R}^{\widetilde{K}}$.

Using this notation our RBF net can be rewritten as follows:

$$y(\mathbf{x}) = \mathbf{a}^T\, \mathbf{v}(\mathbf{x}) + \mathbf{b}^T\, \mathbf{w}(\mathbf{x}), \tag{4}$$

where $\mathbf{b} \overset{def}{=} [b_1, b_2, \ldots, b_{\widetilde{K}}]^T$. In our approach to selecting RBF net structure we shall use the so called t-Student statistical test. Its proper usage requires the assumption that our initial RBF net has a sufficiently rich structure that allows for generating our learning sequence as follows:

$$y_i = (\mathbf{a}^0)^T \mathbf{v}(\mathbf{x}_i) + \sum_{m=1}^{K} b^0_{k(m)}\, w_{k(m)}(\mathbf{x}_i) + \varepsilon_i \quad i = 1, 2, \ldots, n, \tag{5}$$

where $\mathbf{a}^0 \in \mathbf{R}^r$ and $b^0_{k(m)}$, $m = 1, 2, \ldots, K$ are unknown parameters, while output observations y_i contain additive i.i.d. random errors ε_i, $i = 1, 2, \ldots, n$. We assume that $\varepsilon_i \sim \mathcal{N}(0, \sigma^2_\epsilon)$ for formal derivations, although one can use our algorithm on heuristic grounds, even if these assumptions are violated.

3 Random Projections of Model Terms and Outline of Their Selection

Details of the proposed method are presented in the next section, while here we present a general idea.

Our starting point is the following model

$$\bar{y}(\mathbf{x}, \mathbf{a}, \beta, \mathbf{s}) = \mathbf{a}^T \mathbf{v}(\mathbf{x}) + \beta\, \mathbf{s}^T \mathbf{w}(\mathbf{x}), \tag{6}$$

where $\mathbf{a} \in \mathbf{R}^r$ are unknown weights of our preliminary RBF net, $\beta \in \mathbf{R}$ is an unknown weight of randomly mixed RBF's $\mathbf{w}(\mathbf{x})$. Random mixing of these terms is done by multiplying them by random vector $\mathbf{s} \in \mathbf{R}^{\widetilde{K}}$ which is drawn at random by the experimenter from the multivariate Gaussian distribution: $\mathcal{N}(\mathbf{0}, \sigma_s^2\, \mathbf{I}_{\widetilde{K}})$, $\sigma_s > 0$, where $\mathbf{I}_{\widetilde{K}}$ is $\widetilde{K} \times \widetilde{K}$ identity matrix. Later on, we shall write $\mathbf{s} \sim \mathcal{N}(\mathbf{0}, \sigma_s^2\, \mathbf{I}_{\widetilde{K}})$.

Remark 1. *Model (6) resembles a model that was proposed in [2] for selecting terms in a regression function (see also [31] page 131) as well as models used in the dimensionality reduction (see [24] and the bibliography cited therein). However, the fundamental difference is in that here* $\mathbf{s}$ *is selected at random and only* β *is estimated, while in [2] both* β *and* $\mathbf{s}$ *are estimated, which confines the possibility of applying the latter approach when* $\tilde{K} + r << n$*, as assumed here. In [24] parameters of several deterministic projections of* $\mathbf{x}$ *itself are estimated.*

Before starting our random sieve of RBF's in $\mathbf{w}(\mathbf{x})$ it is expedient to test whether our preliminary net, spanned by RBF's contained in $\mathbf{v}(\mathbf{x})$ is properly selected. Notice that we can use classical tools of regression analysis (see, e.g., [31]), because the number of terms in $\mathbf{v}(\mathbf{x})$ is smaller than the length of a learning sequence. In particular, one can estimate $\mathbf{a}$ by minimizing $\sum_{i=1}^n (y_i - \mathbf{a}^T \mathbf{v}(\mathbf{x}_i))^2$ and then test the hypothesis that particular components of $\mathbf{a}$ are zero. After reducing those RBF's that correspond to nonessential parameters, we can start our random sieve.

For fixed $\mathbf{s}$, estimates $\hat{\mathbf{a}}$, $\hat{\beta}$ of parameters $\mathbf{a}$ and β are obtained by ordinary least squares (OLS), i.e., minimizing

$$\min_{\mathbf{a},\beta} \sum_{i=1}^{n} \left[y_i - \bar{y}(\mathbf{x}_i, \mathbf{a}, \beta, \mathbf{s})\right]^2, \tag{7}$$

Then, we state the null hypothesis: $H_0 : \beta = 0$. Under assumptions: (5) and $\varepsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$ we can test it by the well known t-test for regression parameters (see, e.g., [23]). The rejection of H_0 means that our observations contradict the hypothesis that $\beta = 0$. This is an indicator that the mixture $\mathbf{s}^T \mathbf{w}(\mathbf{x})$ may contain terms that are useful in modeling the learning sequence by our RBF net. To convince ourselves, new $\mathbf{s}$ is drawn $\mathcal{N}(\mathbf{0}, \sigma_s^2 \mathbf{I}_{\widetilde{K}})$ and the estimation (7) and the test are repeated *rep* times, say.

If H_0 was rejected a sufficient number of times ($0.2\,rep$, say), we conclude that $\mathbf{w}(\mathbf{x})$ may contain terms that are worth introducing into the model. Otherwise, we stop the algorithm, concluding that only $\mathbf{a}^T \mathbf{v}(\mathbf{x})$ are essential and we have to re-estimate $\mathbf{a}$ by OLS.

If H_0 was rejected sufficiently many times, we have to identify which terms are important. To this end vector $\mathbf{w}(\mathbf{x})$ will be repeatedly divided (roughly) in half in further derivations. To define subdivisions it is convenient to introduce an overloaded notation defined as follows $\widetilde{K}//2$ is : if $\widetilde{K}$ is even, then $\widetilde{K}//2 = \widetilde{K}/2$, otherwise, $\widetilde{K}//2$ is understood as the largest integer less than $\widetilde{K}/2$ for $\mathbf{w}_L(\mathbf{x})$ vectors and as the smallest integer larger than $\widetilde{K}/2$ for $\mathbf{w}_R(\mathbf{x})$ vectors. The same convention is used for further subdivisions $\mathbf{w}_{LL}(\mathbf{x})$, $\mathbf{w}_{LR}(\mathbf{x})$ etc. and for random vectors $\mathbf{s}_L$, $\mathbf{s}_R \sim \mathcal{N}(\mathbf{0}, \sigma_s^2 \mathbf{I}_{\widetilde{K}//2})$, assuming that they have the same dimensions as the corresponding $\mathbf{w}_L(\mathbf{x})$, $\mathbf{w}_R(\mathbf{x})$, $\mathbf{w}_{LL}(\mathbf{x})$, $\mathbf{w}_{LR}(\mathbf{x})$ vectors. Furthermore, we assume that random vectors $\mathbf{s}_L$, $\mathbf{s}_R$, $\mathbf{s}_{LL}$, $\mathbf{s}_{LR}$ etc. are mutually independent.

The corresponding left and right parts of $\mathbf{w}(\mathbf{x})$ will be denoted by $\mathbf{w}_L(\mathbf{x})$, $\mathbf{w}_R(\mathbf{x})$, $\mathbf{w}_{LL}(\mathbf{x})$, $\mathbf{w}_{LR}(\mathbf{x})$, $\mathbf{w}_{RL}(\mathbf{x})$, $\mathbf{w}_{RR}(\mathbf{x})$ etc. In subsequent steps the following RBF nets will be used:

$$\bar{\bar{y}}(\mathbf{x}, \mathbf{a}, \beta_L, \beta_R, \mathbf{S}) = \mathbf{a}^T \mathbf{v}(\mathbf{x}) + \beta_L \mathbf{s}_L^T \mathbf{w}_L(\mathbf{x}) + \beta_R \mathbf{s}_R^T \mathbf{w}_R(\mathbf{x}), \tag{8}$$

where $\mathbf{a} \in \mathbf{R}^r$, $\beta_L, \beta_R \in \mathbf{R}$, $\mathbf{s}_L, \mathbf{s}_R \in \mathbf{R}^{\widetilde{K}//2}$, $\mathbf{S} \overset{def}{=} [\mathbf{s}_L, \mathbf{s}_R]$, $\mathbf{w}_1(\mathbf{x})$, $\mathbf{w}_2(\mathbf{x}) \in \mathbf{R}^{\widetilde{K}//2}$

We state the hypothesis that in (8) $H_{0L} : \beta_L = 0$ and analogously $H_{0R} : \beta_R = 0$. We draw $\mathbf{s}_L$ and $\mathbf{s}_R$ at random and we find the estimate $\hat{\mathbf{a}}$, $\hat{\beta}_L$ and $\hat{\beta}_R$ by

$$\min_{\mathbf{a}, \beta_I, \beta_R} \sum_{i=1}^{n} \left[y_i - \bar{\bar{y}}(\mathbf{x}_i, \mathbf{a}, \beta_L, \beta_R, \mathbf{S})\right]^2 . \tag{9}$$

and t test is applied for $\hat{\beta}_L$ and $\hat{\beta}_R$. Then we store the results of testing and $\mathbf{s}_L$ and $\mathbf{s}_R$ are again drawn at random and (9) and t tests are repeated *rep* times, say. Simultaneously, we increment counters, denoted as c_L, and c_R, each time when $H_{0L} : \beta_L = 0$, respectively $H_{0R} : \beta_R = 0$, is rejected. If, for preselected threshold $0 < \theta < 1$, we have $c_L < \theta\,rep$ and $c_R < \theta\,rep$, then STOP – there are no additional RBF's that are essential for our net.. Otherwise, if $c_L \geq \theta\,rep$ and $c_L > c_R$ we split $\mathbf{w}_L(\mathbf{x})$ in half and we repeat the above steps for model

$$\bar{\bar{\bar{y}}}(\mathbf{x}, \mathbf{a}, \beta_{LL}, \beta_{LR}, \ldots) = \mathbf{a}^T \mathbf{v}(\mathbf{x}) + \beta_{LL} \mathbf{s}_{LL}^T \mathbf{w}_{LL}(\mathbf{x}) + \beta_{LR} \mathbf{s}_{LR}^T \mathbf{w}_{LR}(\mathbf{x}), \tag{10}$$

(or for its 'right' counterpart). Simultaneously, if also $c_R \geq \theta\,rep$, we keep $\mathbf{w}_R(\mathbf{x})$ terms for further considerations as prospective, otherwise we skip $\mathbf{w}_R(\mathbf{x})$ in further steps. If our algorithm attains the stage that $\mathbf{w}_{LR...RL}(\mathbf{x})$ contains only one element we add it, after t test, to the list of candidates to be introduced to our RBF net. If the list of prospective terms is not empty, we enter it as a new $\mathbf{w}(\mathbf{x})$ list and repeat the entire procedure. Finally, we have a list of candidates that is used as the extension of $\mathbf{v}(\mathbf{x})$, the parameters of the extended RBF net are re-calculated and undergo t tests. A more detailed description of the above approach is given in the next section.

4 Detailed Description of the Algorithm.

Below, we present a detailed description of the random sieve algorithm. The notations are the same as in the previous section. In parenthesis we provide suggested values of parameters that were verified in simulations as useful.

Preparations:

- Collect observations $(\mathbf{x}_i, y_i)$, i=1, 2,. . . , n.
- Select RBF centers $\mathbf{C}_j$, $j = 1, 2, \ldots, r$ and $\mathbf{c}_k$, $k = 1, 2, \ldots, \widetilde{K}$.
- Select kernel Ker (Gaussian) and smoothing parameters $0 < h_1 < h_2$ (h_1 and h_2 should be selected taking the number of observations into account and a fine tuning based on cross-validation should be performed).
- Form vectors $\mathbf{v}(\mathbf{x})$ and $\mathbf{w}(\mathbf{x})$ according to N1) and N2).
- Select parameters: $\sigma_s > 0$ ($\sigma_s = 3$) for generating random vectors $\mathbf{s}$ etc.
- Choose working significance level $0 < \alpha < 1$ ($\alpha = 0.1$) that is used in t-test for random sieve and final check significance level $0 < \alpha_f < \alpha < 1$ ($\alpha_f = 0.05$).
- Choose the number of random projections $rep \geq 1$ ($rep = 200$), i.e., the number of repetitions of random projections and t-test before deciding whether a group of RBF's is prospective or not.
- Select the threshold $0 < \theta < 1$ ($\theta = 0.2$) as the fraction of positive trials required to consider a group of RBF's as perspective (see [3] for the explanations on critism when multiple testing is used).

Initialization:

- Set counter $c_0 = 0$. It counts how many times H_0 was rejected for a group of RBF's under consideration.
- Prepare three empty lists: *candidates* (of RBF's to be added to a net), *prospective* (RBF's worth to be considered as the most perspective) and *waiting* (the list of RBF's to be considered later).
- Check whether $\mathbf{v}(\mathbf{x})$ does not contain unnecessary RBF's. To this end, solve the following OLS problem: $\min_{\mathbf{a}} \sum_{i=1}^{n}(y_i - \mathbf{a}^T\, \mathbf{v}(\mathbf{x}_i))^2$ and test the sequence of hypothesis $H_0 : \mathbf{a}^{(j)} = 0$, $j = 1, 2, \ldots, r$. This is realizable due to our assumption that $r < n$. Remove from $\mathbf{v}(\mathbf{x})$ those $v_j(\mathbf{x})$ for which $H_0 : \mathbf{a}^{(j)} = 0$ was not rejected[1]. Rename the obtained vector as $\mathbf{v}(\mathbf{x})$ again and again denote its length by r.

[1] If the resulting list of preliminary RBF's is empty, select at least one additional candidate RBF and add it to this list.

Step 1. Draw at random $\mathbf{s} \in \mathbf{R}^{\widetilde{K}}$, $\mathbf{s} \sim \mathcal{N}(\mathbf{0},\, \sigma_s^2\, \mathbf{I}_{\widetilde{K}})$. Form RBF net (6) and calculate $\hat{\mathbf{a}}(\mathbf{s})$ and $\hat{\beta}(\mathbf{s})$ by OLS. Test the hypothesis: $\hat{\beta}(\mathbf{s}) = 0$ at the level α. If the hypothesis is rejected, the set $c_0 = c_0 + 1$.

Step 2. Repeat Step 1 *rep* times. If $c_0 < \theta\, rep$, STOP with the message: *probably there are no RBF's from the list* $\mathbf{w}(\mathbf{x})$ *to be added*, otherwise, go to Step 3.

Step 3. Enter all the terms from $\mathbf{w}(\mathbf{x})$ to the *prospective* list.

Step 4. Split the *prospective* list in half. Replace $\mathbf{w}_L(\mathbf{x})$ in (8) by the left part of this list and $\mathbf{w}_R(\mathbf{x})$ by the right half. Set counters $c_L = 0$, $c_R = 0$.

Step 5. Generate random Gaussian vectors $\mathbf{s}_L$ and $\mathbf{s}_R$ of the same lengths as the current $\mathbf{w}_L(\mathbf{x})$ and $\mathbf{w}_R(\mathbf{x})$ and calculate $\hat{\mathbf{a}}(\mathbf{S})$, $\hat{\beta}_L(\mathbf{s}_L)$ and $\hat{\beta}_R(\mathbf{s}_R)$ by minimizing (9). Test the hypothesis: $\hat{\beta}_L(\mathbf{s}_L) = 0$ (respectively, $\hat{\beta}_R(\mathbf{s}_R) = 0$) and set $c_L = c_L + 1$ (respectively, set $c_L = c_L + 1$), if it is rejected.

Step 5. Repeat Step 5 *rep* times. If $c_L < \theta\, rep$ AND $c_R < \theta\, rep$, go to Step 6. Otherwise, if $c_L \geq c_R$ and

Step 5a. if current $\mathbf{w}_L(\mathbf{x})$ contains more than one term, then replace all the content of *prospective* list by $\mathbf{w}_L(\mathbf{x})$ and add $\mathbf{w}_R(\mathbf{x})$ to the *waiting* list, but only if $c_R \geq \theta\, rep$, otherwise, reject $\mathbf{w}_R(\mathbf{x})$ from considerations and go to Step 4,

Step 5b. if current $\mathbf{w}_L(\mathbf{x})$ contains exactly one term, than add it to the *candidate* list and add $\mathbf{w}_R(\mathbf{x})$ to the *waiting* list, but only if $c_R \geq \theta\, rep$, otherwise, reject $\mathbf{w}_R(\mathbf{x})$ from further considerations. Then replace the content of the *prospective* list by all the *waiting* list, set the *waiting* list to be empty and go to Step 4.

If $c_L < c_R$, perform Steps 5a) and 5b), replacing the roles $\mathbf{w}_L(\mathbf{x})$ and $\mathbf{w}_R(\mathbf{x})$.

Step 6. Final decisions:

- If list *candidates* is empty, STOP with the message: *probably there are no RBF's from* $\mathbf{w}(\mathbf{x})$ *to be added.*
- If the length of the *candidates* list is larger than 0 but not larger than $(n{-}r)$, then add this list to $\mathbf{v}(\mathbf{x})$, estimate the parameters of the extended RBF net and test their significance at the level α_f. Reject nonsignificant RBF's, re-calculate parameters $\mathbf{a}$ and selected b_k's and STOP, providing the final list of RBF's.
- If the length of the *candidates* list is larger than $(n-r)$, then the candidate list is still too long in comparison to available data. It is desirable to enlarge the learning sequence, replace $\mathbf{w}(\mathbf{x})$ by the list of candidates and go to Step 3. If we cannot get additional learning examples, we can still replace $\mathbf{w}(\mathbf{x})$ by the list of candidates and go to Step 3, but this time it is more probable than certain essential RBF's will be left outside the final net structure.

Remark 2. *The above algorithm can also be used for simultaneous selection of RBF's and input variables. To this end, it suffices to replace* $\mathbf{v}_j(\mathbf{x})$*'s by* $Ker(||Sel_D[\mathbf{x} - \mathbf{C}_j]||/h_1)$ *and* $w_k(\mathbf{x})$ *by* $Ker(||Sel_D[\mathbf{x} - \mathbf{c}_k]||/h_2)$*, where the selector function* $Sel_D[.]$ *is defined as follows.* D *is a subset of those indexes* $\{1,\, 2, \ldots,\, d\}$ *of input variables that are not set to zero by function* Sel*. For example, if* $d = 4$ *and* $D = 1, 4$*, then* $Sel_D[[x^{(1)},\, x^{(2)},\, x^{(3)},\, x^{(4)}]] = [x^{(1)}, 0, 0, x^{(4)}]$.

5 Simulations

The aim of our simulations was to verify performance of the algorithm using an example of moderate size. For clarity of the interpretation we have simulated a simple RBF net with input variables on the unit square. Preliminary positions of Gaussian RBF's, i.e., those included in $\mathbf{v}(\mathbf{x})$ were in the nodes of the following grid: $(i\,0.2,\, j\,0.2)$, $i,\, j = 0,\, 1, \ldots,\, 5$. Thus, $\mathbf{v}(\mathbf{x})$ contained $r = 36$ elements, but

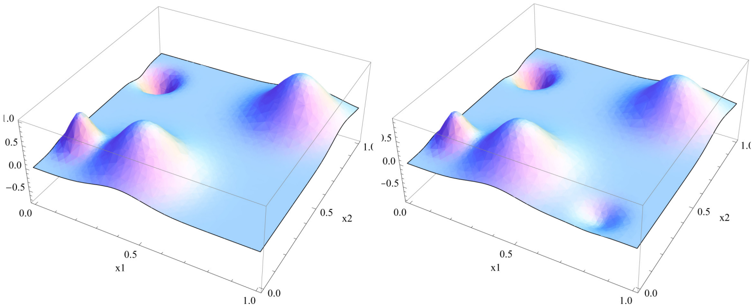


Fig. 1. "True" surface (left panel) and its reconstruction by one of the randomly sieved BRF net (right panel)

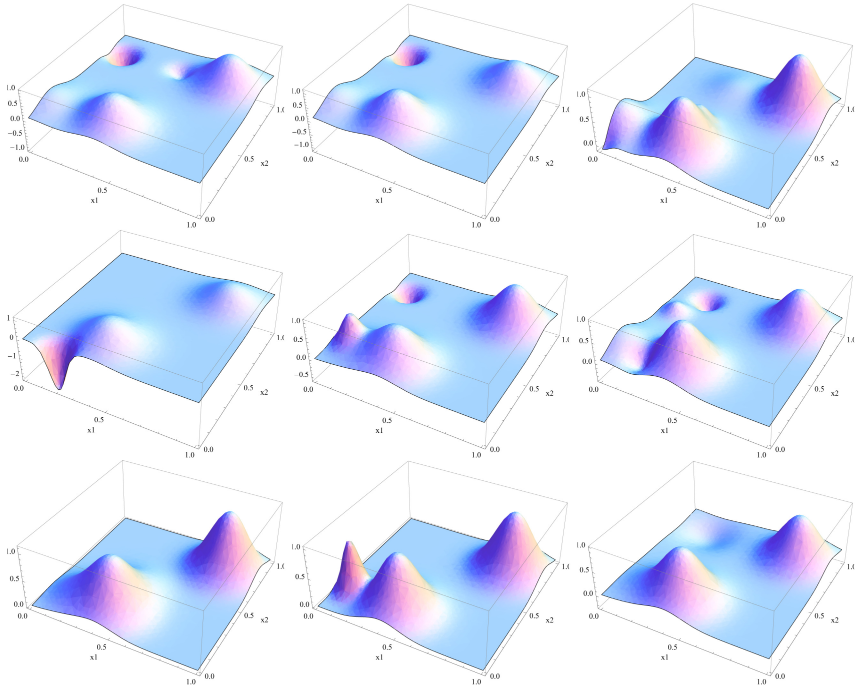


Fig. 2. A collection of randomly sieved RBF nets for approximating the surface shown in Fig. 1 (left panel)

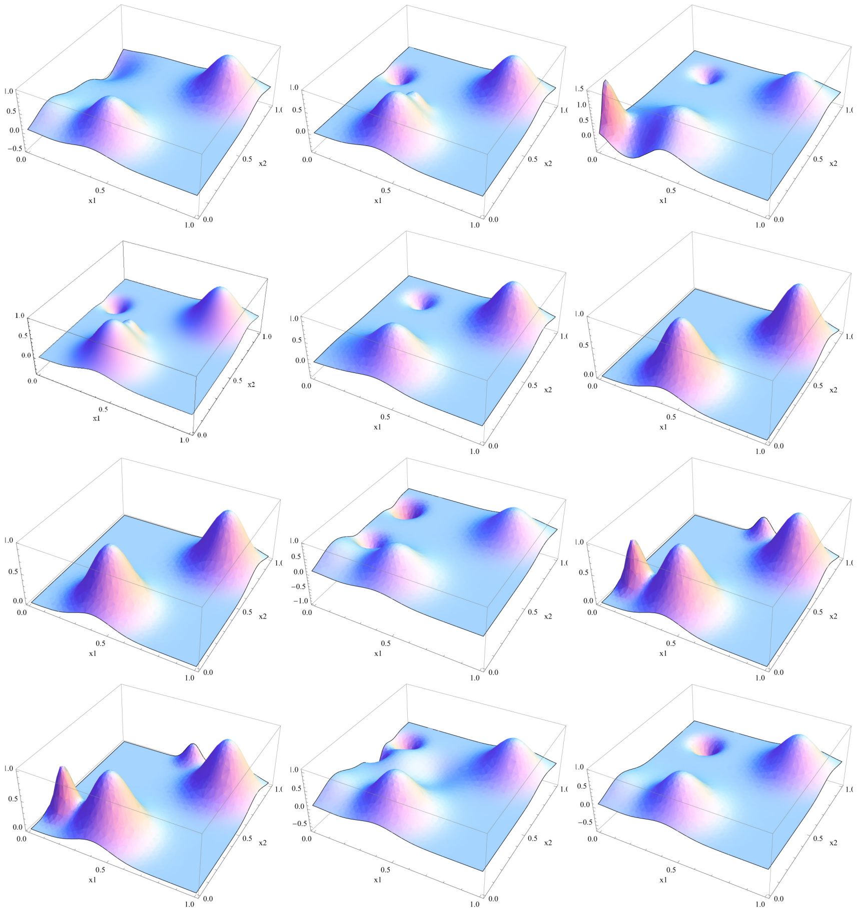


Fig. 3. The second collection of randomly sieved RBF nets for approximating the surface shown in Fig. 1 (left panel)

only two of them (higher hills in Fig. 1) had weights 2.5 in our simulations, the rest of the weights were set to zero. As candidates $\mathbf{w}(\mathbf{x})$ to be entered to the net we take RBF's with centers at the grid: $(i\,0.1,\,j\,0.1)$, $i,\,j = 0,\,1,\ldots,\,10$. Thus, $\widetilde{K} = 121$ and we have $r + \widetilde{K} = 157$ RBF's to be selected. To this end only $n = 50$ observations $(\mathbf{x}_i,\,y_i)$'s were generated, where $\mathbf{x}_i$'s are generated as the Hammersley sequence (see Tablet1). The reason for selecting a quasi-random sequence of the Halton and Hammersley type is that it has proved to be useful in other tasks such as regression estimation (see [19], [30]). Their usefulness results from a better conditioning of the matrix $M \overset{def}{=} \sum_{i=1}^{r} \mathbf{v}(\mathbf{x}_i)\,\mathbf{v}^T(\mathbf{x}_i)$, which has $\kappa(M) \overset{def}{=} \frac{\lambda_{max}(M)}{\lambda_{min}(M)} = 9622$ and we can avoid using a regularization. For comparison, when $\mathbf{x}_i$'s are generated as uniform random variables, then $\kappa(M)$ is of the order 10^6 and a kind of regularization is necessary (see, e.g., [15] for a discussion on this topic). Also classical design points (see [16]) lead to a good conditioning of M, but this is achieved by the necessity of applying a large number of them.

The rest of the parameters were selected as follows $h_1 = 0.025$, $h_2 = 0.005$ and they were not optimized, $\sigma_s = 3$, $\sigma_\epsilon = 0.1$.

Two RBF's (contained in $\mathbf{w}(\mathbf{x})$) should be introduced to the net that is visible in Fig. 1 (left panel) as the smallest hill and as the hole, with weights 0.75 and −0.75, respectively. The two large hills (with weights 2.5) were present in a preliminary part of the net, i.e., in $\mathbf{v}(\mathbf{x})$.

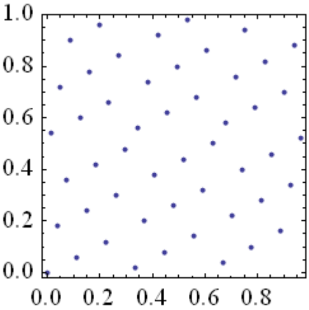
We shall say that our algorithm achieved:

- full success (FS), if it detected all four RBF's and there were no spurious terms detected,
- partial success (PS), if only one additional RBF from $\mathbf{w}(\mathbf{x})$ was detected plus two RBF's from $\mathbf{v}(\mathbf{x})$, independently whether additional terms in improper positions were found or not.

The results of the simulations are summarized in Table 1 (left panel). They seem to be satisfactory, since we had three times more RBF's to be considered than observations and in about eighty percent of runs at least one from two RBF's that were hidden in the noise was detected. The execution time was varied between runs and it took from 9 to 122 seconds on a standard PC with the i7 processor.

Table 1. Left panel – the percentage of full (FS) and partial successes (PS). Right panel – a sequence of 50 Hammersley points used in simulations.

Success	FS	PS	FS+PS
%	18.2	63.6	81.8



6 Concluding Remarks

An important feature of the proposed approach is the dimensionality reduction that comes from random projections of candidate RBF's. The idea of using random projections for this purpose was introduced by the first author in [25] in the context of usually even larger models arising in the identification of nonlinear time series. This aspect of the present paper as well as bounds on the probabilities of properly selecting all necessary RBF's, while avoiding introducing spurious ones are outside the scope of this paper.

One can consider other strategies of running calculations of divided and randomly projected sub-nets that are more suitable for parallel computers. The proposed approach can also be useful in signal restoration (see, e.g., [11]) for selecting sin terms that are used for a signal approximation.

References

1. Bazan, M., Skubalska-Rafajłowicz, E.: A new method of centers location in Gaussian RBF interpolation networks. In: Rutkowski, L., et al. (eds.) ICAISC 2013, Part I. LNCS (LNAI), vol. 7894, pp. 20–31. Springer, Heidelberg (2013)
2. Cook, R.D., Weisberg, S.: Partial one-dimensional regression models. Amer. Stat. 58, 110–116 (2004)
3. Donoho, D., Jin, J.: Higher criticism for detecting sparse heterogeneous mixtures. The Annals of Statistics 32, 962–994 (2004)
4. Fornberg, B., Larsson, E., Flyer, N.: Stable computations with Gaussian radial basis functions. SIAM J. Sci. Comput. 33(2), 869–892 (2011)
5. Fu, X., Wang, L.: Data dimensionality reduction with application to simplifying RBF network structure and improving classification performance. IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics 33(3), 399–409 (2003)
6. Hansen, P.C.: Rank-deficient and discrete ill-posed problems. SIAM, Philadelphia (1998)
7. Girosi, F., Jones, M., Poggio, T.: Regularization theory and neural networks architectures. Neural Computation 7(2), 219–269 (1995)
8. Gyorfi, L., Kohler, M., Krzyżak, A., Walk, H.: A Distribution-free Theory of Nonparametric Regression, ch. 21. Springer, Berlin (2000)
9. Konishi, S., Kitagawa, G.: Information Criteria and Statistical Modeling, Springer (2008)
10. Krzyżak, A., Linder, T.: Radial Basis Function Networks and Complexity Regularization in Function Learning. IEEE Trans. Neural Networks 9, 247–256 (1998)
11. Krzyżak, A., Rafajłowicz, E., Pawlak, M.: Moving average restoration of bandlimited signals from noisy observations. IEEE Transactions on Signal Processing 45, 2967–2976 (1997)
12. Leonardisa, A., Bischof, H.: An efficient MDL-based construction of RBF networks. Neural Networks 11, 963–973 (1998)
13. Lewis, S.M., Dean, A.M.: Detection of interactions in experiments on large numbers of factors (with discussion). Journal of the Royal Statistical Society, Series B 63, 633–672 (2001)
14. Morris, M.D.: An Overview of Group Factor Screening. In: Dean, A.M., Lewis, S.M. (eds.) Screening Methods for Experimentation in Industry, Drug Discovery, and Genetics, ch. 9, pp. 191–207. Springer, New York (2006)
15. Orr, M.J.: Regularization in the selection of basis function centers. Neural Computation 7(3), 606–623 (1995)
16. Rafajłowicz, E., Myszka, W.: Optimum experimental design for a regression on a hypercube-generalization of Hoel's result. Annals of the Institute of Statistical Mathematics 40, 821–827 (1988)
17. Rafajłowicz, E., Pawlak, M.: Optimization of centers' positions for RBF nets with generalized kernels. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI), vol. 3070, pp. 253–259. Springer, Heidelberg (2004)

18. Rafajłowicz, E., Skubalska-Rafajłowicz, E.: RBF nets based on equidistributed points. In: Proceedings of 9th IEEE Int. Conf.: Methods and Models in Automation and Robotics MMAR, pp. 921–926 (2003)
19. Rafajłowicz, E., Schwabe, R.: Halton and Hammersley sequences in multivariate nonparametric regression. Statistics and Probability Letters 76, 803–812 (2006)
20. Rutkowski, L.: Adaptive Probabilistic Neural Networks for Pattern Classification in Time-Varying Environment. IEEE Trans. Neural Networks 15(4), 811–827 (2004)
21. Rutkowski, L.: Generalized Regression Neural Networks in Time-Varying Environment. IEEE Trans. Neural Networks 15(3), 576–596 (2004)
22. Rutkowski, L.: New Soft Computing Techniques for System Modeling. Pattern Classification and Image Processing. Springer, Heidelberg (2004)
23. Seber, G.A.F.: Linear regression Analysis. Wiley, New York (1977)
24. Shaker, A.J., Prendergast, L.A.: Iterative application of dimension reduction methods. Electronic Journal of Statistics 5, 1471–1494 (2011)
25. Skubalska-Rafajłowicz, E.: Experiments with neural network for modeling of nonlinear dynamical systems: Design problems. Lecture presented at The Newton's Mathematical Institute, Cambridge, DAE seminar led by D. Uciński (2011), `www.newton.ac.uk/programmes/DAE/seminars/072010301.html`
26. Skubalska-Rafajłowicz, E., Rafajłowicz, E.: Random projections in regression model selection and corresponding experiment design problems. To be presented at Model Oriented Data Analysis Conference, Lagów, Poland (June 2013)
27. Skubalska-Rafajłowicz, E.: Random projection RBF nets for multidimensional density estimation. International Journal of Applied Mathematics and Computer Science 18(4), 455–466 (2008)
28. Skubalska-Rafajłowicz, E.: Detection and estimation translations of large images using random projections. In: 7th International Workshop Multidimensional (nD) Systems (nDs), September 5-7 (2011)
29. Skubalska-Rafajłowicz, E.: Neural networks with sigmoidal activation functions–dimension reduction using normal random projection. Nonlinear Anal.: Theory, Methods & Appl. 71, e1255–e1263 (2009)
30. Skublska-Rafajłowicz, E., Rafajłowicz, E.: Sampling multidimensional signals by a new class of quasi-random sequences. Multidimensional System and Signal Processing 23, 237–253 (2012)
31. Weisberg S.: Applied Linear Regression. Wiley & Sons, Inc., Hoboken (2005)
32. Xu, L., Krzyżak, A., Yuille, A.: On radial basis function nets and kernel regression: statistical consistency, convergence, rates and receptive field size. Neural Networks 4, 609–628 (1994)