Analysis of Input–Output Clustering for Determining Centers of RBFN

Zekeriya Uykan, Cüneyt Güzeliş, M. Ertugrul Çelebi, Member, IEEE, and Heikki N. Koivo, Senior Member, IEEE

Abstract—The key point in design of radial basis function networks is to specify the number and the locations of the centers. Several heuristic hybrid learning methods, which apply a clustering algorithm for locating the centers and subsequently a linear least-squares method for the linear weights, have been previously suggested. These hybrid methods can be put into two groups, which will be called as input clustering (IC) and input–output clustering (IOC), depending on whether the output vector is also involved in the clustering process. The idea of concatenating the output vector to the input vector in the clustering process has independently been proposed by several papers in the literature although none of them presented a theoretical analysis on such procedures, but rather demonstrated their effectiveness in several applications. The main contribution of this paper is to present an approach for investigating the relationship between clustering process on input–output training samples and the mean squared output error in the context of a radial basis function network (RBFN). We may summarize our investigations in that matter as follows: 1) A weighted mean squared input–output quantization error, which is to be minimized by IOC, yields an upper bound to the mean squared output error. 2) This upper bound and consequently the output error can be made arbitrarily small (zero in the limit case) by decreasing the quantization error which can be accomplished through increasing the number of hidden units.

Index Terms—Input/output clustering, radial basis function neural networks.

I. INTRODUCTION

RADIAL basis function network (RBFN), with the simplicity of its single-hidden layer structure, is a good alternative to multilayer perceptron, especially in the applications requiring local tunable property. Linear output layer and radial basis hidden layer structure of RBFN provide the possibility of learning the connection weights efficiently without local minima problem in a hierarchical procedure so that the linear weights are learned after determining the centers by a clustering process.

The RBF method has traditionally been used for strict interpolation in multidimensional space by Powell [25] and Michelli [18] requiring as many centers as data points (assigning all input data points as centers). Later Broomhead and Lowe [2] removed the “strict” restriction and used less centers than data samples, so allowing many practical RBFN applications in which the number of data samples is very large. Today RBFN has been a focus of studies, not only for numerical analysis but also in neural networks area.

Using the Gaussian kernel function, RBFN is capable of forming an arbitrarily close approximation to any continuous function [12], [15]. Chen and Chen [5], presented a general result on approximating to nonlinear functionals and operators by RBFN using sample data either in frequency or in time domain.

RBFN with its radially symmetric activation functions produces a localized response to inputs. Different radial basis functions have been used in the literature, such as thin-plate splines \( \phi(a) = a^2 \log(a) [9] \), Hardy multiquadratics \( \phi(a) = \sqrt{a^2 + c} \) (\( c \geq 0 \)) [11], the function \( \phi(a) = a^k \) (\( k \) is odd integer) [24], Gaussian kernel \( \phi(a) = e^{-a^2} \) [26], etc. The Gaussian RBFN which is the most commonly used will be considered in this paper. However, the analysis we shall present is also valid for any Lipschitz continuous radial basis function.

The construction of RBFN involves three different layers: Input layer which consists of source nodes, hidden layer in which each neuron computes its output using a radial basis function and output layer which builds a linear weighted sum of hidden layer outputs to supply the response of the network. RBFN with one output neuron (Fig. 1) implements the input–output relation in (1).

\[
F(\lambda; C; x) = \sum_{j=1}^{N} \lambda_j \phi(||x - c_j||_2)
\]

where

- \( \phi(a) = e^{-a^2} \) chosen radial basis function;
- \( N \) number of hidden neurons;
- \( x \in \mathbb{R}^P \) input;
- \( ||\cdot||_2 \) Euclidean norm;
- \( C = [c_1, \ldots, c_N] \) matrix whose columns are the centers of RBFN;
- \( \lambda = [\lambda_1 \cdots \lambda_N] \) linear weight vector.
The analysis of Sections III and IV consider single output RBFN just for brevity. Since the results obtained are also true for multiooutput RBFN, input–output clustering (IOC) method in Section II is described for general multioutput RBFN’s.

The center vectors \( \{c_j\}_{j=1}^N \) are fixed points in \( p \)-dimensional input space. Theoretical and numerical studies show that the performance of RBFN highly depends on the locations of centers and is regardless which radial basis function is used in hidden neurons.

The learning strategies in the literature used for the design of RBFN differ from each other mainly in the determination of centers and can be categorized into the following groups.

1) Fixed Centers Assigned Randomly Among Input Samples [16]: In this method, which is the simplest one, the centers are chosen randomly from the set of input training samples.

2) Orthogonalization of Regressors: The common feature of this class of algorithms is that they orthogonalize the regressors to decouple the contribution of each regressor to an energy based performance index. The most commonly used algorithm is orthogonal least squares (OLS) [7] which selects a suitable set of centers (regressors) — but might not be the optimal set as demonstrated in [27] — among input training samples. Procedures choose centers one by one so that, at each step, the selected center maximizes the increment to the output energy. Besides this, a few other algorithms have been proposed: Fast orthogonal estimation algorithm [32], refined forward regression orthogonalization algorithm [17], genetic evolution of regressors using orthogonal niches [31].

3) Supervised Selection of Centers: In this method, the centers together with all other parameters of RBFN (linear weights, variances) are updated using a backpropagation-like gradient descent algorithm.

4) Input Clustering (IC) [19]: The locations of centers are determined by a clustering algorithm (such as vector quantization (VQ) [14] or k-means [10]) applied to input training sample vectors. In the literature, different clustering algorithms have been used for center determination of RBFN: k-means [19], mean tracking clustering [28], recursive k-means [4], median RBFN [1], an iterative clustering method presented in [20]. Nevertheless none of them discussed rigorously the relationship between the input quantization error and the output error over the training samples.

5) Input–Output Clustering (IOC) [6], [29], [23]: In IOC method, clustering is applied to the augmented vectors in (2) which are obtained by concatenating the output vector to the weighted input vector and the resulting cluster codebook vectors are rescaled and projected into the input space to obtain the centers. Therefore, the locations of centers are influenced not only by the input sample spread but also by the output sample deviations.

In the design of RBFN, once the centers have been fixed, the optimal linear weights can be determined straightforwardly by using a linear least squares algorithm, or taking the pseudoinverse, which generally means less time for learning as compared to the determination of all RBFN parameters using supervised learning.

The idea of concatenating the output vector to the input vector in the clustering process has independently been addressed by several papers such as [3], [6], [13], [23], [29], [30]. The IOC method has also been independently applied in a variety of tasks including nonlinear and/or time-varying dynamical system identification [6], [23], [29], function approximation [6], [23], [29], restoration of nonlinearly degraded images [3], phoneme recognition [8], and chaotic time series prediction [22], in most of which IOC is also compared to IC and is shown to be superior. But none of these studies has revealed an analysis of IOC, rather they offer some heuristics together with demonstrating its effectiveness in above mentioned applications. The main goal of this paper is to investigate the relationship between clustering process and the RBFN output error. The main result obtained is 1) The weighted mean squared input–output quantization error subject to the minimization in the clustering process constitutes an upper bound to the mean squared output error function and 2) this upper bound, and hence the output error, can be made arbitrarily small (zero in the limit) by increasing the number of hidden neurons, i.e., by decreasing the (weighted) input–output quantization error.

This paper is organized as follows. Section II describes the IOC. The relevance of the desired outputs in the determination of the optimal centers of RBFN is explained in Section III. Section IV investigates the relationship between clustering process and the output error function. Section V presents computer simulation results, followed by Conclusions in Section VI.

II. INPUT–OUTPUT CLUSTERING FOR CENTER DETERMINATION

The IOC method can be applied in four different ways [29] depending on how the data is fed to the network:

1) Batch mode clustering of centers and batch mode gradient descent for linear weights.
2) Batch mode clustering of centers and pattern mode gradient descent for linear weights.
3) Pattern mode clustering of centers and pattern mode gradient descent for linear weights.
4) Pattern mode clustering of centers and batch mode gradient descent for linear weights.

We hereby mean by “pattern mode” that adaptation of the parameters is done for every incoming sample and by “batch mode” that the adaptation is done only after collecting a number of samples, even possibly the whole data set, which depends on the designer. For simplicity, only the first way will be described below.

Let us consider an RBFN with \( p \) inputs and \( q \) outputs. The vectors to which a clustering algorithm will be applied are obtained by augmenting the weighted input sample vectors with the desired outputs as in (2)

\[
\hat{d}_s^{\text{IO}} = \left[ \gamma \hat{x}_s^T \ d_s^T \right]^T, \quad \{x_s\}_{s=1}^L \in \mathbb{R}^p, \quad \{d_s\}_{s=1}^L \in \mathbb{R}^q
\]

where \( L \) is the number of training vectors and \( \gamma \) is a weighting factor.

The IOC algorithm where the batch mode is used for both clustering and linear weight update consists of the following steps.

Step 1) Apply a batch mode clustering algorithm to the set of augmented vectors in (2) and then find the
augmented centers $m_j = [(\gamma m_j^x)^T (\gamma m_j^y)^T]^T \in \mathbb{R}^{p+q})$ of the clusters where $\gamma m_j^x \in \mathbb{R}^p$, $m_j^y \in \mathbb{R}^q$. Then rescale the first $p$ entries with $1/\gamma$. 

Step 2) As a projection of the augmented centers into the input space, take the first $p$ entries of $\mathbf{m}_j$'s and form the centers of the RBFN as $c_j^{(o)} = m_j^x \in \mathbb{R}^p$. 

Step 3) Apply a batch mode gradient descent algorithm to learn the optimal linear weights for the fixed centers in Step 2). 

In Step 1), one can choose any metric for quantization error. Euclidean quantization error, namely the mean squared distortion, is the most common. The analysis of Section IV shows that the mean squared output error of RBFN admits an upper bound in terms of a weighted form of this Euclidean input–output quantization error. Weighting specifically the input part of training samples with some factors in IOC processes has also been reported in [14], [6], [23] and Section IV provides elaborations on the usefulness of it. Without loss of generality, we only weight the input part of the augmented vector.

III. DEPENDENCE OF OPTIMAL CENTER LOCATIONS ON DESIRED OUTPUTS

The design of RBFN in learning of a mapping described by a set $\{x_s, d_s\}_{s=1}^L$ of $L$ input-(desired) output samples can be defined as a minimization problem where the cost is a measure of the distance between actual and desired output sets. As conventionally posed, by optimal RBFN parameters (centers and linear weights) we mean the parameters which minimize the following mean squared output error over the finite set of $L$ training samples

$$
\mathcal{E}(\lambda; C) = \frac{1}{L} \sum_{s=1}^L \| d_s - F(\lambda; C; x_s) \|_2^2
$$

(3)

where

- $\lambda = [\lambda_1 \cdots \lambda_N]$ linear weight vector;
- $C = [c_1, \cdots, c_N]$ matrix whose columns are the centers of RBFN;
- $\{x_s, d_s\}_{s=1}^L$ set of input-(desired) output pairs;
- $F(\cdot)$ transfer function of the RBFN defined in (1).

The output error in (3) can be written in a matrix form as follows:

$$
\mathcal{E}(\lambda; C) = \frac{1}{L} \| \mathbf{d} - H \lambda \|_2^2
$$

(4)

where

- $\mathbf{c} = [c_1 \cdots c_N]^T$;
- $\mathbf{d} = [d_1 \cdots d_N]^T$;
- $H_{ij} = \phi'(\|x_s - c_j\|_2)$;
- $\lambda = [\lambda_1 \cdots \lambda_N]^T$.

Substituting the optimal value for the linear weight vector $\lambda^{opt} = H^+ d$ to (4), the output error function is rewritten as

$$
\mathcal{E}(\lambda^{opt}; C) = \frac{1}{L} \| d - HH^+ d \|_2^2 = \frac{1}{L} \| (I - HH^+) d \|_2^2
$$

(5)

where $H^+$ is the pseudoinverse of the matrix $H$.

The expression in (5) shows that the desired output vector $d$ affects the cost function, thus, it should be taken into account in the determination of the optimal centers. Now, the following question arises: How can we choose the centers producing a matrix $H$ such that the row vectors of the matrix $(I - HH^+)$ are as perpendicular to the desired output vector $d$ as possible to obtain the smallest cost? This problem is very hard to be resolved since the entries of the matrix $(I - HH^+)$ are quite complicated functions of the centers. The expression in (5) does not provide an efficient way to find the optimal centers. Nevertheless, it emphasizes the need to include the desired output into the consideration of determining the centers. Another motivation for the IOC method comes from the below explained structure of the local optimal centers as an output-dependent, weighted-sum of input samples.

Using (1) let us rewrite the output error function $\mathcal{E}(\lambda; C)$ in (3) as

$$
\mathcal{E}(\lambda; C) = \frac{1}{L} \sum_{s=1}^L \left( d_s - \sum_{j=1}^N \lambda_j \phi(\|x_s - c_j\|_2) \right)^2
$$

(6)

The first-order necessary conditions for the optimal centers are obtained by setting the gradients of $\mathcal{E}(\lambda; C)$ (with respect to $c_j$'s) to zero, i.e., $\nabla c_j \mathcal{E}(\lambda; C) = 0$, from which one obtains the following structure that the local optimal centers must have:

$$
c_j = \frac{\sum_{s=1}^L \beta_{s,j} x_s}{\sum_{s=1}^L \beta_{s,j}}
$$

(7)

where $\beta_{s,j} = 4 \epsilon_s \lambda_j \phi'_{ij}$, in which $\epsilon_s = d_s - \sum_{j=1}^N \lambda_j \phi(\|x_s - c_j\|_2)$, and $\phi'$ is the derivative of the radial basis function with respect to its argument, i.e., $\phi' = \frac{d \phi(a)}{da}$. Equation (7) refers to “a kind of” clustering process over the whole input–output training set. Each input sample $x_s$ affects all centers $c_j$'s in different degrees depending on the $\beta_{s,j}$ weights. The dependence of the $\beta_{s,j}$ on the desired output samples points out that the IC method, considering only the input samples, in general, does not give optimal center values. As done in IOC method, the desired outputs also should be taken into the consideration.

IV. THE EFFECT OF IOC ON THE OUTPUT ERROR

This section attempts to find an answer to the question “What kind of relationship exists between the clustering process on a set of input–output sample vectors and the mean squared output error of an RBFN obtained for the same sample set?”

Here, we will consider a clustering process which minimizes the following weighted $l_2$ quantization error over the augmented vectors of (2)

$$
\mathcal{D}(M) = \frac{1}{L} \sum_{s=1}^L \| m_s^{(o)} - m_{j(s)} \|_2^2
$$

$$
= \frac{1}{L} \sum_{s=1}^L \| (x_s - m_{j(s)}) \|_2^2
$$

$$
+ \frac{1}{L} \sum_{s=1}^L \| d_s - m_{j(s)} \|_2^2
$$

(8)
where

\[ M = \begin{bmatrix} m_1, \cdots, m_N \end{bmatrix} \]

matrix whose columns are the cluster center vectors in the input–output space;

\[ \gamma \]

weighting factor which balances the roles of the input and output samples on the shape of the clusters and on the locations of the cluster centers;

\[ j(s) : \{1, 2, \cdots, L\} \rightarrow \{1, 2, \cdots, N\} \]

index identifying the cluster to which the sample \( t_s \) belongs.

In order to see the effect of the clustering process on the output error, let us consider the mean squared output error versus the mean squared quantization error such that the centers of RBFN ultimately be chosen by IOC as \( \mathbf{c}_j^{IO} = \mathbf{m}_j^y \)

\[
\frac{1}{L} \sum_{s=1}^{L} \left\| d_s - F(\lambda; \mathbf{C}; \mathbf{x}_s) \right\|^2_v

versus \[
\frac{1}{L} \sum_{s=1}^{L} \left\| \mathbf{c}_j^{IO} - \mathbf{m}_{j(s)} \right\|^2_v.
\]

The mean squared output error \( \mathcal{E}(\lambda; \mathbf{C}) \) in (3) can be rewritten as in (10) by adding and subtracting the term \( F(\lambda; \mathbf{C}; \mathbf{m}_{j(s)}^y) \)

\[
\mathcal{E}(\lambda; \mathbf{C}) = \frac{1}{L} \sum_{s=1}^{L} \left\| d_s - F(\lambda; \mathbf{C}; \mathbf{x}_s) + F(\lambda; \mathbf{C}; \mathbf{m}_{j(s)}^y) - F(\lambda; \mathbf{C}; \mathbf{m}_{j(s)}^y) \right\|^2_v.
\]

Using the triangular inequality rule and subsequently the fact that the \( l_2 \) norm of a vector is not greater than its \( l_1 \) norm, we reach the following inequality:

\[
\mathcal{E}(\lambda; \mathbf{C}) \leq \frac{1}{L} \sum_{s=1}^{L} \left( \left\| F(\lambda; \mathbf{C}; \mathbf{x}_s) - F(\lambda; \mathbf{C}; \mathbf{m}_{j(s)}^y) \right\|_1 + \left\| d_s - F(\lambda; \mathbf{C}; \mathbf{m}_{j(s)}^y) \right\|_1 \right)^2.
\]

Since \( F(\lambda; \mathbf{C}) \) is Lipschitz continuous with a Lipschitz constant \( K \) (see Appendix), then we get a new inequality

\[
\mathcal{E}(\lambda; \mathbf{C}) \leq \frac{1}{L} \sum_{s=1}^{L} \left( K \left\| \mathbf{x}_s - \mathbf{m}_{j(s)}^y \right\|_1 + \left\| d_s - F(\lambda; \mathbf{C}; \mathbf{m}_{j(s)}^y) \right\|_1 \right)^2.
\]

The argument of the square term on the right-hand side of the inequality (12) is indeed a weighted \( l_1 \) norm of an augmented vector such that \( K \) weights the input part. This observation together with the fact that the \( l_1 \) norm of a \((p + q)\)-dimensional vector is not greater than \( \sqrt{p + q} \) times its \( l_2 \) norm yields a greater upper bound as given in (13). Herein, for the considered single-output RBFN case, \( q \) is equal to unity

\[
\mathcal{E}(\lambda; \mathbf{C}) \leq \frac{1}{L} \sum_{s=1}^{L} \left( \sqrt{p + q} \left\| \left( K \left( \mathbf{x}_s - \mathbf{m}_{j(s)}^y \right) \right)^T \left( \mathbf{d}_s - F(\lambda; \mathbf{C}; \mathbf{m}_{j(s)}^y) \right) \right\|_2 \right)^2.
\]

Hence, we obtain the following upper bound \( UB(\lambda; \mathbf{C}; \mathbf{M}) \) for the mean squared output error \( \mathcal{E}(\lambda; \mathbf{C}) \)

\[
UB(\lambda; \mathbf{C}; \mathbf{M}) = (p + q) \left( \frac{K^2}{L} \sum_{s=1}^{L} \left\| \mathbf{x}_s - \mathbf{m}_{j(s)}^y \right\|_2^2 \right) + \frac{1}{L} \sum_{s=1}^{L} \left\| \mathbf{d}_s - F(\lambda; \mathbf{C}; \mathbf{m}_{j(s)}^y) \right\|_2^2.
\]

Let us consider the case that \( \mathbf{C} = \mathbf{M}^x \) and the linear weight vector \( \lambda \) is chosen to make the RBFN a strict interpolator transforming the input space projections \( \mathbf{m}_j^x \)'s to the output space projections \( \mathbf{m}_j^y \)'s of the cluster centers, i.e., for all \( j \)

\[
\mathbf{m}_j^y = F(\lambda_{\text{opt}}^{\text{clust}}; \mathbf{M}^x; \mathbf{m}_j^x)
\]

where \( \lambda_{\text{opt}}^{\text{clust}} \) is the optimal linear weight vector that defines this interpolator, which can be computed by solving (15). In the case of \( \mathbf{C} = \mathbf{M}^x \) and \( \lambda = \lambda_{\text{opt}}^{\text{clust}} \), with considering the dependence of the Lipschitz constant \( K \) on the RBFN linear weight parameters, the upper bound can be given as

\[
UB(\lambda_{\text{opt}}^{\text{clust}}; \mathbf{M}^x; \mathbf{M}) = (p + q) \left( \frac{(K_{\text{opt}}^{\text{clust}})^2}{L} \sum_{s=1}^{L} \left\| \mathbf{x}_s - \mathbf{m}_{j(s)}^y \right\|_2^2 \right) + \frac{1}{L} \sum_{s=1}^{L} \left\| \mathbf{d}_s - \mathbf{m}_j^{y(\text{opt})} \right\|_2^2
\]

where \( K_{\text{opt}}^{\text{clust}} = \sqrt{(2/\alpha)} |\lambda_{\text{opt}}^{\text{clust}}|_1 \) for unit-variance Gaussian RBFN (see Appendix).

If the weighting factor \( \gamma \) is chosen equal to the \( K_{\text{opt}}^{\text{clust}} \), then \( UB(\lambda_{\text{opt}}^{\text{clust}}; \mathbf{M}^x; \mathbf{M}) \) becomes the same as \((p + q)\) times the weighted mean squared input–output quantization error \( D(\mathbf{M}) \) in (8), and hence their minimizations with respect to the matrix \( \mathbf{M} \) become equivalent.

Since \( \mathcal{E}(\lambda; \mathbf{C}) \leq UB(\lambda; \mathbf{C}; \mathbf{M}) \), we have \( \mathcal{E}(\lambda_{\text{opt}}^{\text{clust}}; \mathbf{M}^x) \leq UB(\lambda_{\text{opt}}^{\text{clust}}; \mathbf{M}^x; \mathbf{M}) \). By choosing the linear weight vector equal to \( \lambda_{\text{opt}}^{\text{clust}} \) which denotes the optimal \( \lambda \) calculated for the whole training data and for a considered \( \mathbf{M}^x \) as the matrix of centers, we obtain a mean squared output error not greater than the error obtained for \( \lambda_{\text{opt}}^{\text{clust}} \). i.e.,

\[ \mathcal{E}(\lambda_{\text{opt}}^{\text{whole}}; \mathbf{M}^x) \leq \mathcal{E}(\lambda_{\text{opt}}^{\text{clust}}; \mathbf{M}^x). \]

This follows from the fact that both errors are obtained over the same training data, but \( \lambda_{\text{opt}}^{\text{clust}} \) is optimal only for the set \( \{\mathbf{m}_j^x, \mathbf{m}_j^y\}_{j=1}^{N} \) of pairs
of cluster center projections, not for the training set. These relations finally provide the main result of our analysis: The mean squared output error, calculated for the linear weights chosen optimal after fixing the centers of RBFN, admits an upper bound in terms of the mean squared quantization error, and the quantization is taken over the input–output training set used in the design of the RBFN

\[
\frac{1}{L} \sum_{s=1}^{L} (d_s - F(\chi_{\text{out}}^{\text{opt}}; M^x; x_s))^2 \\
\leq (p + q) \left( \frac{1}{L} \sum_{s=1}^{L} \left| K_{\text{clust}}^{\text{opt}} (x_s - m_s) \right|^2_2 \\
+ \frac{1}{L} \sum_{s=1}^{L} \left| d_s - m_{j(o)} \right|^2_2 \right). \tag{17}
\]

The established connection between the output and quantization errors is useful for understanding the abilities of IOC and IC in selecting the optimal RBFN centers. Under the assumption of choosing a suitable weighting factor \( \gamma \) during the clustering process, we can state the following fact: For a given training set and a given number of hidden neurons, a weighted \( l_2 \) clustering in the input–output space can be used for minimizing an upper bound of the mean squared output error. As IOC method minimizes the quantization error in input–output space, IC method, on the other hand, takes care of only the first term in the upper bound, i.e., the scaled mean squared input quantization error, and therefore it cannot, in general, give the minimum of the upper bound. For a weighting factor \( \gamma \) which is greater than Lipschitz constant \( K_{\text{clust}}^{\text{opt}} \cdot D(M) \) constitutes an upper bound to the output error. This upper bound is tight only when \( \gamma \) is chosen close to the minimum Lipschitz constant \( K_{\text{clust}}^{\text{opt}} \). These two facts imply that, for a badly chosen weighting factor \( \gamma \), the scaled input–output quantization error \( D(M) \) subject to the minimization by IOC may not constitute an upper bound to the output error or it may not be tight, therefore, its minimization may not have any relevance to the minimization output error.

Remark: \( UB(\chi_{\text{clust}}^{\text{opt}}; M^x; M) \) in (16) is not the only measure for the input–output quantization error such that it constitutes an upper bound to the mean squared output error. Some other measure yielding smaller upper bound can also be found. Such an upper bound is as follows:

\[
UB^*(\chi_{\text{clust}}^{\text{opt}}; M^x; M) = \left( \frac{K_{\text{clust}}^{\text{opt}}}{L} \sum_{s=1}^{L} \left| x_s - m_{j(o)} \right|_1 \right)^2 \\
+ \frac{1}{L} \sum_{s=1}^{L} \left| d_s - m_{j(o)} \right|_1^2. \tag{18}
\]

This upperbound may be derived from (11) in a way similar to the derivation of the upper bound in (16).

\( UB^*(\chi_{\text{clust}}^{\text{opt}}; M^x; M) \) is not a standard measure. It constitutes a vector norm based measure such that individual quantization errors are measured in \( l_1 \) norm and they are averaged in a mean squared sense. Considering the tightness of \( UB^*(\chi_{\text{clust}}^{\text{opt}}; M^x; M) \), one can think of devising an IOC algorithm which minimizes this quantization error for determining the centers. However, according to the following facts, it seems that there is no need to introduce such a new clustering algorithm, and hence an \( l_2 \) based IOC algorithm can be used also for the minimization of \( UB^*(\chi_{\text{clust}}^{\text{opt}}; M^x; M) \) in an approximate sense. First fact to be mentioned is that two upper bounds are qualitatively equivalent since it is straightforward to show that \( UB(\chi_{\text{clust}}^{\text{opt}}; M^x; M) \leq UB^*(\chi_{\text{clust}}^{\text{opt}}; M^x; M) \leq UB(\chi_{\text{clust}}^{\text{opt}}; M^x; M) \leq UB(\chi_{\text{clust}}^{\text{opt}}; M^x; M) \) which is exactly the function subject to the minimization by an \( l_2 \) based IOC algorithm. For these reasons, in the applications, we used an \( l_2 \) based IOC algorithm and considered \( UB^*(\chi_{\text{clust}}^{\text{opt}}; M^x; M) \) as well as \( UB(\chi_{\text{clust}}^{\text{opt}}; M^x; M) \) as the upper bounds.

V. COMPUTER SIMULATION RESULTS

Two numerical examples are presented in the following. In Example 1, the output error function and the above mentioned upper bounds are computed using some samples taken from a scalar function. Example 2 compares the performance of IOC with that of IC in system identification problem.

Example 1: The aim of this example is to test the upper bound numerically (for the unit-variance RBFN designed by the IOC) on some samples from a simple scalar function addressing also the issue of \( \gamma \)'s selection due to its relationship with \( \chi_{\text{clust}}^{\text{opt}} \). The scalar function used is given in (19) and is plotted in Fig. 2 together with training samples marked as “o”

\[
g(x) = 0.5(\sin(2\pi x/5) + \sin(2\pi x/3)) \quad 0 \leq x \leq 10. \tag{19}
\]

As explained in Section IV, \( \gamma \) should be chosen equal to the \( K_{\text{clust}}^{\text{opt}} \). However, \( K_{\text{clust}}^{\text{opt}} \) depends on \( \chi_{\text{clust}}^{\text{opt}} \), which is not known in advance. Therefore, the following iterative procedure is taken. Initially, \( \gamma \) is chosen to be unity. Later, with respect to this initial value, the optimal centers are found using IOC, and the optimal linear weights are computed thereafter. The new value of \( \gamma \) is calculated according to \( \gamma_{\text{new}} = K_{\text{clust}}^{\text{opt}} \). This procedure continues until the computed upper bound falls below a sufficiently small value.
Fig. 3. Output error (OE) obtained by IOC and its upper bounds (UB, UB+) for the function in (19).

Considering the relation between γ and $K_{clust}^{opt}$, γ can be calculated from $K_{clust}^{opt} = \sqrt{\text{var}(\text{quant})}$ (see Appendix) or alternatively from input–output data set as follows: For each data sample, calculate $K_s = \|F(x_{\text{clust}}^{opt}; M; \mathbf{x}_s) - F(x_{\text{clust}}^{opt}; M^*; \mathbf{m}_{j(s)}^*)\|_1$ and choose $\gamma = \max_s\{K_s\}$. In our simulations, γ is calculated in the latter way.

The output error $(1/L) \sum_{s=1}^{L} (d_s - F(x_{\text{clust}}^{opt}; M^*; \mathbf{x}_s))^2$ and its upper bounds $UB(x_{\text{clust}}^{opt}; M^*; M)$ and $UB(x_{\text{clust}}^{opt}; M^*; M)$ obtained for the RBFN’s designed by IOC are shown in Fig. 3 with respect to the number of hidden neurons. The VQ algorithm of [14] is applied in the clustering process. The optimal linear weights are computed by well-known linear least mean square rule. Fig. 3 shows that the upper bounds minimized by IOC are quite tight for γ chosen close to minimum Lipschitz constant.

Example 2: The nonlinear dynamical system in (20) which is borrowed from [21] is considered here to compare the performance of IOC method with that of IC in system identification task. The clustering and the computation of optimal linear weights are done as in Example 1

$$y_{k+1} = \frac{y_k y_{k-1} y_{k-2} y_{k-3} - 1(y_k - 1) + y_k}{1 + y_k^2 + y_{k-1}^2 + y_{k-2}^2}.$$  

(20)

Where $y_k$ and $y_k$ are the input and output at time $k$, respectively. A uniformly distributed random excitation sequence with zero mean and with amplitude between $[-1, 1]$ is applied to the system and 100 input–output pairs are provided. A cross validation test over the set of 100 samples is carried out as follows: 15 (15%) randomly selected samples are excluded from the set—to be used in the variance calculation—and the design of unit-variance Gaussian RBFN models is based on the rest 85 samples-training set. This is repeated 30 times. Thus, the average variance is calculated over the excluded samples as $\sum_{s=1}^{30} (\sum_{i=1}^{30}(d_i - F(\lambda; C; \mathbf{x}_i))^2)/30$ where $d_i$ and $F(\lambda; C; \mathbf{x}_i)$ are the outputs of the system and the RBFN model respectively. The results are plotted in Fig. 4, in which the IOC method gave better performance in comparison with the IC method.

VI. CONCLUSIONS

A bridge between the RBFN output error and quantization error over the same training set has been constructed in a deterministic setting. It has been shown that $l_2$ output error admits an upper bound in terms of a weighted $l_2$ input–output quantization error. For a suitable weighting factor, the upper bound and then the output error can be made arbitrarily small by choosing sufficiently large number of hidden nodes. These results provide a way of understanding the underlying mechanism of clustering processes (specifically, IOC and IC) in the design of RBFN. The derivations have been made for Gaussian basis functions with unit variance; however, they can straightforwardly be extended to any Lipschitz continuous RBF case.

Using different measures for output and quantization errors, some other connections may be constructed which might lead a deeper understanding of the ability of IOC and IC in approximating to optimal RBFN.

APPENDIX

In the sequel, we will prove that the RBFN $F(\cdot)$ in (1) is Lipschitz continuous and it has a global Lipschitz constant depending on the linear weight parameters.

Let us first show the Gaussian basis function with the unity variance ($\phi(\alpha) = e^{-\alpha^2}$) has the global Lipschitz constant $k = \sqrt{2/c}$.

Since $\phi(\cdot)$ is a differentiable function, we can apply the mean value theorem to write the difference of the range values of the function in terms of their inverse images

$$\phi(a) - \phi(b) = e^{a^2} - e^{b^2} = (a - b)e^{\mu a + (1 - \mu)(b - a)}$$

with $\mu \in [0, 1]$. The derivative of $\phi(\cdot)$ is $\phi'(\alpha) = -2\alpha e^{-\alpha^2}$ whose minimum is at the point $a = (1/\sqrt{2})$. Indeed, the absolute value of the derivative has the point $a = (1/\sqrt{2})$ as its maximum, i.e., $|\phi'(\alpha)| \leq \sqrt{2/c}$. So we obtain the following inequality:

$$|\phi(a) - \phi(b)| \leq k|a - b|$$

(21)
where $k = \sqrt{2/c}$ is the global Lipschitz constant of the radial basis function.

With the definition of $F(\cdot)$, we have

$$
\|F(x_0) - F(x)\|_1 \leq \sum_{j=1}^{N} \lambda_j \|\phi(||x_a - c_j||_2) - \phi(||x_0 - c_j||_2)||_1.
$$

(24)

Since the absolute value of the multiplication of two real numbers is smaller than the multiplication of their absolute value, one obtains the following inequality:

$$
\|F(x_0) - F(x)\|_1 \leq \sum_{j=1}^{N} |\lambda_j| \|\phi(||x_a - c_j||_2) - \phi(||x_0 - c_j||_2)||_1.
$$

(25)

Consequently, the global Lipschitz constant of $F(\cdot)$ is $K = \sqrt{2/c}||\lambda||_1$.

REFERENCES


Cüneyt Güzelis received the B.Sc., M.Sc., and Ph.D. degrees from Istanbul Technical University (ITU), Istanbul, Turkey, in 1981, 1984, and 1989, respectively. In 1982 he joined ITU as an Assistant Professor and is now a Professor. He was a Visiting Researcher and Lecturer in the Department of Electrical Engineering and Computer Science, University of California, Berkeley, from April 1989 to April 1991, and was a Visiting Professor in the Information Laboratory, University of Paris-Nord, Paris, France, in September 1996 and June 1997. His research interests include nonlinear circuits and systems and neural networks and their signal processing applications.

M. Ertugrul Çelebi (S’84–M’86) received the B.S. degree from Istanbul Technical University (ITU), Istanbul, Turkey, in 1981, the M.S. and the Ph.D. degrees from Polytechnic University, Brooklyn, NY, in 1984 and 1990, respectively, all in electrical engineering.

Since 1992, he has been with the Faculty of Electrical and Electronics Engineering, ITU, Istanbul, Turkey, where he is currently serving as Associate Professor. His research interests include detection and estimation theory, nonlinear system identification, neural networks, higher order statistics, and image restoration and compression.

Heikki N. Koivo (S’67–M’71–SM’86) received the B.S.E.E. degree from Purdue University, West Lafayette, IN, and the M.S. degree in electrical engineering and the Ph.D. degree in control sciences from University of Minnesota, Minneapolis, MN.

He has served in various academic positions at the University of Toronto, Toronto, ON, Canada, and at Tampere University of Technology, Tampere, Finland. Since 1995 he has been Professor in Control Engineering in Helsinki University of Technology, Helsinki, Finland.

Dr. Koivo is a member of the Editorial Board of Journal of Intelligent and Fuzzy Systems and Intelligent Automation and Soft Computing. He was Associate Editor of IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION and member of the Administrative Council of the IEEE Robotics and Automation Society. He is a Fellow of Finnish Academy of Technology.