MIT Media Laboratory Perceptual Computing Group Technical Report #271
To appear in: *IEEE Transactions on Neural Networks.*

On The Efficiency Of The Orthogonal Least Squares Training Method For Radial Basis Function Networks

Alex Sherstinsky[†‡]
Rosalind W. Picard[‡]

[†]Department of Electrical Engineering and Computer Science
[‡]The Media Laboratory

February, 1992
(revised October, 1993)

Abstract

The efficiency of the Orthogonal Least Squares (OLS) method for training approximation networks is examined using the criterion of energy compaction. We show that the selection of basis vectors produced by the procedure is not the most compact when the approximation is performed using a non-orthogonal basis. Hence, the algorithm does not produce the smallest possible networks for a given approximation error. Specific examples are given using the Gaussian Radial Basis Functions (RBF) type of approximation networks.

Keywords: approximation networks, Radial Basis Functions, interpolation, approximation, Linear Least Squares Estimation, Gram-Schmidt, Orthogonal Least Squares, Karhunen-Loève Transform, Gaussians, energy compaction.

Inquiries can be addressed to:
Alex Sherstinsky (e-mail: shers@media.mit.edu) or to
Rosalind W. Picard (e-mail: picard@media.mit.edu),
MIT Media Laboratory, E15-383, 20 Ames Street, Cambridge, MA 02139.

1 Introduction

A number of feed-forward networks with one hidden layer of processing units have been proven to possess the ability to approximate any continuous function arbitrarily well [1], [2]. One such approximation scheme, the Radial Basis Function (RBF) network, has been used as a classifier [3], [4], [5], [6], [7]. The training problem for an RBF network can be viewed as interpolation and solved by inverting a matrix. But this approach often causes numerical problems, because the matrices involved are typically large. This problem has led to several alternatives aimed at reducing the training complexity without significant losses in approximation accuracy [8], [9], [10].

This report analyzes the efficiency of one such method, Orthogonal Least Squares (OLS), proposed by Chen *et al* [11], [10]. Since its original publication, the OLS technique has found use in several applications, such as automatic control [12], [13], [14], fuzzy logic networks [15], [16], and others. However, none of these papers discuss the method's efficiency. The present study suggests that the OLS algorithm is inefficient in its selection of significant basis functions.

Section 2 reviews the RBF approximation problem and the OLS algorithm for solving it. Section 3 presents the compaction criteria, which are subsequently used in Section 4 to analyze the efficiency of the OLS method. Examples using Gaussian RBFs are also given in Section 4. Finally, Section 5 summarizes the present study.

In a more detailed report, we address the issue of using the OLS method in order to judge the overall efficiency of the RBF expansion for image coding [17].

2 Background

2.1 Radial Basis Functions

A non-linear function $h(\vec{x}, \vec{c})$, where $\vec{x}$ is the independent variable and $\vec{c}$ is the constant parameter, is called a Radial Basis Function (RBF) when it depends only on the radial distance $r = \|\vec{x} - \vec{c}\|$, where $\vec{c}$ is its "center". The RBF method is one of the possible solutions to the real multivariate interpolation problem, stated as follows [18], [8], [19], [2], [20], [21]:

Interpolation Problem: *Given N different points $\{\vec{x}_i \in \mathcal{R}^d \mid i = 1, \ldots, N\}$, where d is the number of dimensions, and N real numbers $\{y_i \in \mathcal{R} \mid i = 1, \ldots, N\}$, find a function F from $\mathcal{R}^d$ to $\mathcal{R}$ satisfying the interpolation conditions:*

$$F(\vec{x}_i) = y_i, \qquad i = 1, \ldots, N. \tag{1}$$

The RBF approach consists of choosing the function F to be an expansion of the form

$$F(\vec{x}) = \sum_{j=1}^{N} w_j h(\|\vec{x} - \vec{c}_j\|), \tag{2}$$

where the centers of the expansion $\vec{c}_j = \vec{x}_j$ must be the the known data points, and $\{w_j \in \mathcal{R} \mid j = 1, \ldots, N\}$ are the corresponding weights.

The unknown weights can be recovered by imposing the interpolation conditions. An RBF matrix $H \in \mathcal{R}^{N \times N}$ is constructed by evaluating $h(\|\vec{x}_i - \vec{c}_j\|)$ at each x_i and c_j; $i, j = 1, \ldots, N$:

$$H = [h_{ij}], \; h_{ij} = h(\|\vec{x}_i - \vec{c}_j\|). \tag{3}$$

In other words, each column of H is a basis vector corresponding to a particular center. The resulting linear system

$$H\vec{w} = \vec{y} \tag{4}$$

can be solved if H^{-1} exists:

$$\vec{w} = H^{-1}\vec{y}. \tag{5}$$

From (5), a necessary and sufficient condition for the existence of a unique solution to the interpolation problem is the invertibility of the matrix H. The RBF matrix will be invertible if the column vectors of H form a basis in $\mathcal{R}^N$. This condition is satisfied for a number of RBFs [19].

Figure 1 shows a realization of (2) in the form of a network with one layer of hidden units [10]. Since each radial hidden unit defines a $(d+1)$-dimensional hypersurface, the RBF network interpolates by reconstructing the data with scaled hypersurfaces. The examples in this report employ a special case of RBFs: Gaussians of constant variance.

2.2 Training RBF Networks

In most applications, N is large, deeming the direct use of (5) impractical. However, a well-known result allows dimensionality reduction to $M < N$. Starting with $H \in \mathcal{R}^{N \times N}$, which is a basis in $\mathcal{R}^N$, $\hat{H}$ is obtained by selecting $M = N - k$, $k = 1, \ldots, N$ basis vectors from H, such that $\hat{H} \in \mathcal{R}^{N \times M}$. Then the product $(\hat{H}^T\hat{H}) \in \mathcal{R}^{M \times M}$ is an invertible matrix and thus a basis in $\mathcal{R}^M$ [22]. Using this result, an approximation to (4) can be formulated and solved by the method of Linear Least Squares Estimation (LLSE) [8]:

Approximation Problem: *Given $\hat{H} \in \mathcal{R}^{N \times M}$ and $\vec{y} \in \mathcal{R}^N$, related by*

$$\vec{y} = \hat{H}\vec{w} + \vec{e}, \tag{6}$$

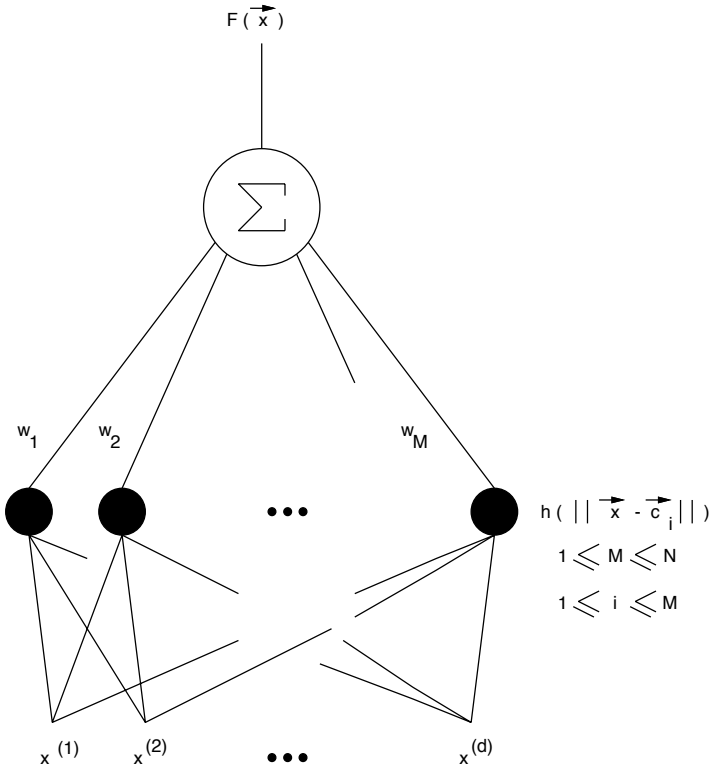

Figure 1: Schematic of an RBF network. The subscripts denote the indices of RBF centers; the superscripts denote the components of the input vector.

find an optimal coefficient vector $\vec{w} \in \mathcal{R}^M$ *such that the error energy* $\vec{e}^T\vec{e}$ *is minimized. This can be equivalently stated as: Find* $\{w_j^\star \in \mathcal{R} \mid j = 1, \ldots, M\}$ *such that* $w_j = w_j^\star$ *solves*

$$\min_{w_j} (y_i - \sum_{j=1}^{M} w_j h(\|\vec{x}_i - \vec{c}_j\|))^2, \; i = 1, \ldots, N. \quad (7)$$

In contrast with the interpolation problem, the approximation problem does not require the centers $\vec{c}_j$ to coincide with $\vec{x}_j$, so one may choose any $\vec{c}_j \in \mathcal{R}^d$. However, the centers are commonly chosen to be a subset of data points [10].

The data is subsequently approximated using

$$\vec{y}_a = \hat{H}\vec{w}^\star \quad (8)$$

or, equivalently, using

$$F_a(\vec{x}) = \sum_{i=1}^{M} w_i^\star h(\|\vec{x} - \vec{c}_i\|), \qquad M \leq N, \quad (9)$$

where $\vec{y}_a$ and $F_a(\vec{x})$ are the approximated values of the data samples and the generalizing function, respectively.

The well-known LLSE optimal solution is in the form of (5):

$$\vec{w}^\star = \hat{H}^+\vec{y}, \quad (10)$$

where $\hat{H}^+$ is the pseudoinverse of $\hat{H}$:

$$\hat{H}^+ = (\hat{H}^T\hat{H})^{-1}\hat{H}^T. \quad (11)$$

If an RBF network with $M << N$ centers adequately approximates the data, then the above approach provides a computationally efficient procedure for determining the weights. However, arbitrarily selecting the centers from data points often results in poor performance in a sense that the networks end up with more nodes than necessary for a desired accuracy of approximation [10].

2.3 Orthogonal Least Squares

In order to improve the performance of an RBF network trained by solving the approximation problem, a judicious selection of centers is needed. It has been reported in [10] that the approximation problem, stated in (6), lends itself to the Orthogonal Least Squares (OLS) method, which is a recursive algorithm for selecting a suitable subset of data points as centers. A basis vector produced at each step of the procedure maximizes the increment of the explained energy of the desired output.

We now review the process of center selection performed by OLS using the concept of permutation matrices.

Definition 2.1 *A permutation of* $H \in \mathcal{R}^{N\times N}$ *is* $H' \in \mathcal{R}^{N\times N}$ *such that each column vector of* H' *is identical to exactly one column vector of* H*. Formally,*

$$H' = HP,$$

where $P \in \mathcal{R}^{N\times N}$ *is a permutation matrix comprised of the column vectors of the identity matrix, whose positions are arranged in one of* $N!$ *possibilities.*

Definition 2.2 *A selection matrix* $S \in \mathcal{R}^{N\times M}$ *is obtained by selecting* $M = N - k$, $k = 1, \ldots, N$ *column vectors of a particular permutation matrix* $P \in \mathcal{R}^{N\times N}$*.*

In OLS, a selection of the original RBF matrix H is obtained and orthonormalized using the classical Gram-Schmidt process (GS). Let $\vec{a}_i$ be the column vectors of $A \equiv HS$. The GS process finds $A = \hat{Q}R$, where the matrix $\hat{Q} \in \mathcal{R}^{N\times M}$ consists of orthonormal column vectors, and the right-triangular matrix $R \in \mathcal{R}^{M\times M}$ contains projection and normalization coefficients computed by GS [1].

[1] For a detailed treatment, consult a standard linear algebra text, such as [22], [23], etc.

Using the selection matrix notation, the approximation problem, stated in (6), takes the following form:

$$\begin{aligned} \vec{y} &= HS\vec{w} + \vec{e}, & (12)\\ HS &= \hat{Q}R, & (13)\\ \vec{y} &= \hat{Q}R\vec{w} + \vec{e}, & (14) \end{aligned}$$

where $\vec{w} \in \mathcal{R}^M$ is the coefficient vector and $\vec{e} \in \mathcal{R}^N$ is the error vector. By defining

$$\vec{g} = R\vec{w}, \tag{15}$$

we obtain an orthonormal expansion of the data vector:

$$\vec{y} = \hat{Q}\vec{g} + \vec{e}. \tag{16}$$

Since (16) is a special case of the approximation problem, due to the orthonormality, its LLSE solution is particularly simple (and well-known):

$$\vec{g} = \hat{Q}^T\vec{y}, \tag{17}$$

from which $\vec{w}$ can be recovered by back substitution:

$$R\vec{w} = \vec{g}. \tag{18}$$

Since $\hat{Q}^T\hat{Q} = I$, the M-dimensional identity matrix, then

$$\begin{aligned} \vec{y}^T\vec{y} &= \vec{g}^T\vec{g} + \vec{e}^T\vec{e} = \sum_{j=1}^{M} g_j^2 + \vec{e}^T\vec{e} \\ &= \sum_{j=1}^{M} (\vec{q_j}^T\vec{y})^2 + \vec{e}^T\vec{e}. \end{aligned} \tag{19}$$

The OLS algorithm begins with H consisting of $M = N$ RBF vectors $\vec{h_j}$, $j = 1, \ldots, N$ and produces $\hat{Q}$ consisting of $M \leq N$ orthonormal regressors [2] $\vec{q_i}$ as well as the selection matrix S. In fact, the key difference between OLS and GS is the computation of the selection matrix S. The OLS method finds S so that GS maximizes $g_i^2 = (\vec{q_i}^T\vec{y})^2$ at each step. In other words, on each iteration $i = 1, \ldots, M$ of the GS orthonormalization procedure, the OLS method selects from the remaining $N - i + 1$ choices the values j and $\vec{h_j}$ such that the resulting regressor $\vec{q_i}$ will give the largest possible energy g_i^2. The algorithm keeps track of the order in which the original basis vectors are selected to form HS by setting $s_{ji} = 1$. The selection procedure is terminated when the error energy has been reduced to the specified tolerance.

[2] We follow the terminology in which "regressor" denotes the orthonormal columns of Q, and "basis vector" is reserved for the columns of H.

3 Efficiency Using the Energy Compaction Criterion

Before giving the numerical examples, it is helpful to distinguish two approaches aimed at finding efficient bases: one is "variational", while the other is not. The variational approach allows the components of the basis vectors to depend on the data, and finds the optimal set of basis vectors, corresponding to some criterion and constraint. In contrast, the non-variational approach starts with fixed basis vectors and searches for a combination that best approximates the data.

In the context of the approximation problem, the criterion is typically the minimization of the mean-squared error, and "smoothness" of the solution is a possible choice for the constraint [21]. Alternatively, using the same criterion, but constraining the basis matrix to be orthogonal, and applying the variational approach leads to the method of "principal components".

3.1 Principal Components Analysis is Variational

It is well-known that the eigenvectors of the covariance matrix of the data are the "principal components", which form the basis that possesses the best energy compaction properties [24], [25]. This basis constitutes the Karhunen-Loève Transform (KLT), which decorrelates the data and maximizes the incremental energy (or variance, in the statistical sense) explained by each regressor. The KLT basis vectors are orthonormal, allowing the approximation problem, (6), to take the form of (16):

$$\vec{y} = \hat{Q}\hat{\vec{g}} + \vec{e}. \tag{20}$$

Let $E[\vec{y}]$ be the expected value of a random vector $\vec{y}$. Then for a general stochastic vector, the principal components are the eigenvectors of the covariance matrix, $C_{\vec{y}}$, sorted in the order of decreasing eigenvalues λ_j (variance or energy):

$$\begin{aligned} C_{\vec{y}} &= E[(\vec{y} - E[\vec{y}])(\vec{y} - E[\vec{y}])^T] \\ &= Q\Lambda Q^T, \\ \Lambda &= \mathrm{diag}(\lambda_1 \ \ldots \ \lambda_N). \\ \text{Since } \vec{g} &= Q^T\vec{y}, \\ C_{\vec{g}} &= E[(\vec{g} - E[\vec{g}])(\vec{g} - E[\vec{g}])^T] \\ &= Q^T C_{\vec{y}} Q = \Lambda. \end{aligned}$$

Even though $\mathrm{trace}(C_{\vec{g}}) = \mathrm{trace}(C_{\vec{y}})$, meaning that the total energy is preserved by Q, the distribution of energy in $C_{\vec{g}}$ is more skewed towards the first few eigenvalues. This is a direct consequence of the fact that the KLT expansion is the solution of the variational problem with

the mean-squared error criterion and the orthogonality constraint. Thus, the KLT is the most compact orthogonal basis, because it produces the most skewed $C_{\vec{g}}$.

The significance of the KLT is in its energy efficiency, and networks that "learn" the principal components of the data need the smallest number of processing units for a given amount of error [26].

3.2 OLS is Non-Variational

The objective of the OLS method is to find the smallest subset of a fixed original basis (while not exceeding the allowed approximation error); therefore, the choices available to the procedure are restricted to various combinations of the original basis vectors. Since the number of candidate subsets is finite, it is natural to view efficiency as a relative measure. Thus, we will adopt the following definition of energy compactness in order to evaluate the efficiency of the OLS method:

Definition 3.1 *Consider the following two schemes for approximating the same data:*

$$\begin{aligned} \vec{y} &= B_1 S_1 \vec{w_1} + \vec{e_1} \quad \textit{and} \\ \vec{y} &= B_2 S_2 \vec{w_2} + \vec{e_2}, \end{aligned}$$

where B_1 and B_2 are bases (i.e., each has an inverse and is capable of interpolating the data). Let S_1 and S_2 be the selection matrices (according to Definition 2.2) with M_1 and M_2 columns, respectively. Assume

$$\vec{e_1}^T \vec{e_1} = \vec{e_2}^T \vec{e_2}.$$

Then B_1 is more compact than B_2 if $M_1 < M_2$.

3.3 Deterministic KLT

In order to judge the energy compaction properties of the OLS method, it is helpful to consider the degenerate or "deterministic" case of the KLT. In the deterministic case, the "covariance matrix" of the data (after the sample mean has been removed) is $C_{\vec{y}} = \vec{y}\vec{y}^T$ and is of rank 1. The entire KLT basis is reduced to only one principal component, which becomes the normalized version of the data vector itself. Thus the energy compaction properties of any orthogonal basis can be judged by how well its vectors align with the data vector. The i-th regressor's energy, g_i^2, is related to the alignment via $g_i^2 = (\vec{q_i}^T \vec{y})^2$. Using this measure, a basis with good energy compaction properties will need only a small number of its vectors to be retained in order to explain the required percentage of the data energy. The remaining basis vectors, which align poorly with the data, can be discarded.

3.4 Orthogonality

A convenient property of an orthogonal basis is that the energy contributions of the component vectors are decoupled. A maximally compact permutation of an orthogonal basis matrix can be formed by computing the projections of the basis vectors onto the data and rearranging the column vectors in the order of decreasing energy. In this new matrix, the energy, g_i^2, of a basis vector (which, due to orthogonality, is also a regressor) as a function of its index, $i = 1, \ldots, M$ becomes monotonically decreasing. As $\vec{e}^T\vec{e}$ in (12) decreases, the basis vectors of progressively smaller energy become involved in the approximation process as needed. It follows that a permutation of an orthogonal basis is the most compact if and only if g_i^2 is monotonically decreasing; no other permutation of the original basis matrix can yield the same error with a smaller M.

In the case of both GS and OLS, determining the energy efficiency is more complicated, because the starting basis is non-orthogonal and the basis vectors cannot be treated separately. A permutation of the basis matrix, whose regressors have monotonically decreasing g_i^2, no longer assures maximal energy compaction. As a consequence, for different error allowances different permutations of the original basis matrix will be the most compact. This will be illustrated with examples in Section 4.2.

4 Energy Compaction Provided by OLS

The most compact permutation produces the smallest possible RBF network for a given error tolerance. Therefore, it is interesting to find out whether or not the selection performed by the OLS procedure is the best in terms of energy packing. It has been stated in [10] that the OLS algorithm can be used to select centers so that adequate and parsimonious RBF networks can be obtained. However, the OLS method is not "optimally-parsimonious". Given a required level of unexplained energy, an optimally-parsimonious training method will pick no more basis vectors than needed, and thereby produce an RBF network with fewer nodes than one with randomly selected centers. We show that the selection made by OLS is not guaranteed to contain the smallest number of centers.

4.1 OLS is not Always Efficient for Non-Orthogonal Bases

If the RBF basis is non-orthogonal, the energy con-

tributions of different basis vectors are mixed (i.e., not independent). Hence, for a general data vector, every step of the OLS procedure is unable to locate the regressor with maximal alignment in the global sense. In other words, even though every step yields a regressor with the largest possible alignment, "the largest" may not be large enough. Since the data vector is the principal component, the OLS algorithm is effectively trying to approximate this principal component as closely as possible at each local step, with no regard for the global energy distribution properties. In a sense, this method is analogous to pursuing "short term profits" as opposed to "long term profits".

Evidently, as the examples below indicate, it is possible to benefit from relaxing the restriction of maximal alignment between the regressor and the data at each step of GS. Admitting some basis vectors that produce regressors with poor alignment may steer GS toward future basis vectors that produce regressors with excellent alignment, such that the overall energy compaction is improved. However, there is no mechanism in OLS to decide ahead of time what the optimal permutation of H should be for a specified error value.

4.2 Examples Using Gaussian RBFs

The goal of this section is to illustrate some cases where the OLS procedure does not select the optimal subset of basis vectors in the energy compaction sense. For clarity, the following examples use one-dimensional data and a 3×3 Gaussian RBF matrix with variance $\sigma = 1$:

$$H = \begin{bmatrix} 1 & 0.606531 & 0.135335 \\ 0.606531 & 1 & 0.606531 \\ 0.135335 & 0.606531 & 1 \end{bmatrix}.$$

4.2.1 Example 1

Let

$\vec{y} = \begin{bmatrix} 190 \\ 80 \\ 200 \end{bmatrix}$, which gives the total energy $\vec{y}^T\vec{y} = 82500$.

In the first step, the OLS procedure selects the second column of H, because it gives the largest projection onto $\vec{y}$. After one step of GS, performed on the remaining columns of H, the third column produces a regressor with the largest alignment. In the third step, the first column must be selected. Combining these steps yields the following permutation of H:

$$HP_{OLS} = \begin{bmatrix} 0.606531 & 0.135335 & 1 \\ 1 & 0.606531 & 0.606531 \\ 0.606531 & 1 & 0.135335 \end{bmatrix},$$

which produces regressors with the energies:

$$g_1^2 = 57728.1,\ g_2^2 = 3447.38,\ \text{and}\ g_3^2 = 21324.6,$$

respectively, and

$$\frac{g_1^2 + g_2^2}{\vec{y}^T\vec{y}} \approx 0.74.$$

For a given data vector, the OLS method always selects the same sequence of regressors, regardless of the desired error. In this case, all three basis vectors are needed in order to approximate as much as 75% of the total energy. However, if OLS were able to select the following permutation of H for this data:

$$HP_{opt} = \begin{bmatrix} 0.135335 & 1 & 0.606531 \\ 0.606531 & 0.606531 & 1 \\ 1 & 0.135335 & 0.606531 \end{bmatrix},$$

which produces regressors with the energies:

$$g_1^2 = 54253.2,\ g_2^2 = 17759.4,\ \text{and}\ g_3^2 = 10487.4,$$

respectively, and

$$\frac{g_1^2 + g_2^2}{\vec{y}^T\vec{y}} \approx 0.87,$$

only two basis vectors would be needed in order to approximate up to 87% of the total energy. Note that even though this permutation of H produces regressors with monotonically decreasing energies g_i^2, it still may not be optimal if it is desired to satisfy (i.e., just exceed) a different error value.

4.2.2 Example 2

In the following example, the first column vector of H is nearly orthogonal to $\vec{y}$. Therefore, one of the other two basis vectors (in this case, the second) must produce a regressor that is nearly parallel to $\vec{y}$, resulting in a large value of the energy. The OLS misses this opportunity, because it searches for the largest alignment at each step. Let

$\vec{y} = \begin{bmatrix} -100 \\ 100 \\ 100 \end{bmatrix}$, which gives the energy $\vec{y}^T\vec{y} = 30000$.

The OLS procedure selects the following permutation of H:

$$HP_{OLS} = \begin{bmatrix} 0.135335 & 1 & 0.606531 \\ 0.606531 & 0.606531 & 1 \\ 1 & 0.135335 & 0.606531 \end{bmatrix},$$

which produces regressors with the energies:

$$g_1^2 = 15614.1,\ g_2^2 = 8019.76,\ \text{and}\ g_3^2 = 6366.17,$$

respectively, and

$$\frac{g_1^2 + g_2^2}{\vec{y}^T\vec{y}} \approx 0.79.$$

However, the following (identity) permutation of H:

$$HP_{opt} = \begin{bmatrix} 1 & 0.606531 & 0.135335 \\ 0.606531 & 1 & 0.606531 \\ 0.135335 & 0.606531 & 1 \end{bmatrix},$$

which produces regressors with the energies:

$$g_1^2 = 480.691,\ g_2^2 = 29305.3,\ \text{and}\ g_3^2 = 213.976,$$

respectively, and

$$\frac{g_1^2 + g_2^2}{\vec{y}^T\vec{y}} \approx 0.99,$$

is clearly superior in case 80% or more of the energy is needed. The OLS will require all three basis vectors, while the other permutation will need only two. Again, note that the optimality is error dependent. If we only needed a 52% accuracy, the one-vector subset chosen by OLS would be optimal.

4.3 Sorting Regressors does not Improve Efficiency

Since the energy function of the orthonormal regressors produced by OLS is not monotonic in general, a seemingly obvious next step would be to sort the regressors in the order of decreasing energy and delete the ones with the smallest g_i^2 contributions, without exceeding the allowed error. However, such a parsimonious sorting of the regressors alters the specific permutation adhered to by the GS process.

The premise of OLS is that the chosen selection of the original RBF vectors can be recovered by reversing GS, a forward recursion procedure. If the columns of $\hat{Q}$ and R are sorted without deleting any $\vec{q}_i$, A can still be recovered. However, deleting any regressors, no matter how insignificant their g_i^2 may be, upsets the consistency of the GS process, thereby precluding the recovery of A and destroying the connection to the original RBF weights. Since every permutation of the original RBF matrix leads via GS to a different set of orthonormal regressors, their energies are meaningful only in the context of a particular permutation selected by OLS. In other words, the sorted regressors cannot be used to train the RBF approximation network.

5 Conclusions

While the OLS method has been believed to find a more efficient selection of RBF centers than a random-based approach [10], it does not produce the smallest RBF network for a given approximation accuracy. Simple examples were constructed to provide intuition about the sources of inefficiency.

Acknowledgements

The authors would like to thank Federico Girosi and Terence Sanger for their comments on the manuscript.

References

[1] G. Cybenko, "Approximations by superpositions of a sigmoidal function," *Math. Control Signals Systems*, vol. 2, no. 4, pp. 303–314, 1989.

[2] T. Poggio and F. Girosi, "Networks and the best approximation property," A.I. Memo #1164, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1989.

[3] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[4] S. Renals and R. Rohwer, "Phoneme classification experiments using radial basis functions," in *Proceedings Of IEEE International Joint Conference on Neural Networks*, (Washington DC), pp. I:461–467, June 1989.

[5] K. Ng, "A comparative study of the practical characteristics of neural network and conventional pattern classifiers," Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, 1990.

[6] R. P. Lippmann, "An introduction to computing with neural nets," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 4–22, Apr. 1987.

[7] R. P. Lippmann, "Pattern classification using neural networks," *IEEE Transactions on Communications*, vol. 27, no. 11, pp. 47–64, 1989.

[8] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Systems*, vol. 2, pp. 321–355, 1988.

[9] J. Moody and C. Darken, "Fast-learning in networks of locally-tuned processing units," *Neural Computation*, vol. 1, no. 2, pp. 281–294, 1989.

[10] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Transactions on Neural Networks*, vol. 2, pp. 302–309, Mar. 1991.

[11] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their application to nonlinear system identification," *International Journal of Control*, vol. 50, no. 5, pp. 1873–1896, 1989.

[12] S. Chen and S. A. Billings, "Neural networks for nonlinear dynamic system modelling and identification," *International Journal of Control*, vol. 56, no. 2, pp. 319–346, 1992.

[13] S. Chen, S. A. Billings, and P. M. Grant, "Recursive hybrid algorithm for non-linear system identification using radial basis function networks," *International Journal of Control*, vol. 55, no. 5, pp. 1051–1070, 1992.

[14] S. Mukhopadhyay and K. S. Narendra, "Disturbance rejection in nonlinear systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 4, pp. 63–72, Jan. 1993.

[15] L.-X. Wang and J. M. Mendel, "Fuzzy basis functions, universal approximation, and orthogonal least squares learning," *IEEE Transactions on Neural Networks*, vol. 3, pp. 807–814, Sept. 1992.

[16] J.-S. R. Jang and C.-T. Sun, "Functional equivalence between radial basis function networks and fuzzy inference systems," *IEEE Transactions on Neural Networks*, vol. 4, pp. 156–159, Jan. 1993.

[17] A. Sherstinsky and R. W. Picard, "On training gaussian radial basis functions for image coding," Tech. Rep. 188, M.I.T. Media Lab Vision and Modeling Group, Feb. 1992.

[18] M. J. D. Powell, "Radial basis functions for multivariable interpolation: A review," in *Algorithms for Approximation* (J. C. Mason and M. G. Cox, eds.), (Oxford), Clarendon Press, 1987.

[19] T. Poggio and F. Girosi, "A theory of networks for approximation and learning," A.I. Memo #1140, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1989.

[20] T. Poggio and F. Girosi, "Extension of a theory of networks for approximation and learning: Dimensionality reduction and clustering," A.I. Memo #1167, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1990.

[21] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proceedings of the IEEE*, vol. 78, pp. 1481–1497, Sept. 1990.

[22] G. Strang, *Linear Algebra and Its Applications*. Academic Press, 1980.

[23] G. Strang, *Introduction to Applied Mathematics*. Cambridge, MA: Wellesley-Cambridge Press, 1986.

[24] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 1965.

[25] R. J. Clarke, *Transform Coding of Images*. Orlando: Academic Press, Inc., 1985.

[26] T. D. Sanger, "Optimal unsupervised learning in a single-layer linear feedforward neural network," *Neural Networks*, vol. 2, pp. 459–473, 1989.