Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks

S. Chen, C. F. N. Cowan, and P. M. Grant

Abstract—**The radial basis function network offers a viable alternative to the two-layer neural network in many applications of signal processing. A common learning algorithm for radial basis function networks is based on first choosing randomly some data points as radial basis function centers and then using singular value decomposition to solve for the weights of the network. Such a procedure has several drawbacks and, in particular, an arbitrary selection of centers is clearly unsatisfactory. The paper proposes an alternative learning procedure based on the orthogonal least squares method. The procedure chooses radial basis function centers one by one in a rational way until an adequate network has been constructed. The algorithm has the property that each selected center maximizes the increment to the explained variance or energy of the desired output and does not suffer numerical ill-conditioning problems. The orthogonal least squares learning strategy provides a simple and efficient means for fitting radial basis function networks, and this is illustrated using examples taken from two different signal processing applications.**

I. Introduction

FEEDFORWARD layered neural networks have increasingly been used in many areas of signal processing. Using a feedforward neural network to process complex signals can be viewed as performing a curve-fitting operation in a multidimensional space, and this may explain why most of the applications in this field employ neural networks to realize some complex nonlinear decision function or to approximate certain complicated data-generating mechanisms. The theoretical justification for such applications is that, provided that the network structure is sufficiently large (that is, a sufficient number of hidden neurons), any continuous function can be approximated to within an arbitrary accuracy by carefully choosing parameters in the network [1], [2]. This result is valid even for networks with only one hidden layer. An obvious disadvantage of neural networks is that they are highly nonlinear in the parameters. Learning must be based on nonlinear optimization techniques, and the parameter estimate may become trapped at a local minimum of the chosen optimization criterion during the learning procedure when a gradient descent algorithm is used. Other optimization techniques, such as the genetic algorithm [3], learning automata [4], and simulated annealing [5], although capable of achieving a global minimum, require extensive computation.

A viable alternative to highly nonlinear-in-the-parameter neural networks is the radial basis function (RBF) network. The RBF method has traditionally been used for strict interpolation in multidimensional space [6]–[8]. The original RBF method requires that there be as many RBF centers as data points, which is rarely practical in signal processing applications, as the number of data points is usually very large. The approach adopted by Broomhead and Lowe [9] gives an approximation to the original RBF method and provides a more suitable basis for the application to signal processing. An RBF network can be regarded as a special two-layer network which is linear in the parameters by fixing all RBF centers and nonlinearities in the hidden layer. Thus the hidden layer performs a fixed nonlinear transformation with no adjustable parameters and it maps the input space onto a new space. The output layer then implements a linear combiner on this new space and the only adjustable parameters are the weights of this linear combiner. These parameters can therefore be determined using the linear least squares (LS) method, which is an important advantage of this approach. Because of the strong connection between RBF and neural networks, it is reasonable to believe that an RBF network can offer approximation capabilities similar to those of the two-layer neural network, provided that the hidden layer of the RBF network is fixed appropriately. This heuristic belief is strongly supported by the theoretical results on the RBF method as a multidimensional interpolation technique (e.g. [8]).

The nonlinearity within an RBF network can be chosen from a few typical nonlinear functions. A general consensus is that the choice of the nonlinearity is not crucial for performance and this opinion can also be justified using the results of a theoretical investigation [8]. However the performance of an RBF network critically depends upon the chosen centers. In practice the centers are often chosen to be a subset of the data. Although researchers are well aware that the fixed centers should suitably sample the input domain, most published results simply assume that the centers are arbitrarily selected from data points. Such a mechanism is clearly an unsatisfactory method for building RBF networks. The resulting RBF networks often either perform poorly or have a large size. Furthermore numerical ill-conditioning frequently occurs owing to the near linear dependency caused by, for example, some centers being too close.

The present study adopts a systematic approach to the problem of center selection. Because a fixed center corresponds to a given regressor in a linear regression model, the selection of RBF centers can be regarded as a problem of subset model selection. The orthogonal least squares (OLS) method can be employed as a forward regression procedure [10] to select a suitable set of centers (regressors) from a large set of candidates. At each step of the regression, the increment to the explained variance of the desired output is maximized. Furthermore oversize and ill-conditioning problems occurring frequently in random selection of centers can automatically be avoided. This rational approach provides an efficient learning algorithm for fitting adequate RBF networks. The modeling of a real-world time series using an RBF network and the training of an RBF network as a communications channel equalizer are used as two

Manuscript received August 20, 1990; revised November 19, 1990. This work was supported by the U.K. Science and Engineering Research Council (Grant Ref. GR/E/10357).

The authors are with the Department of Electrical Engineering, University of Edinburgh, Mayfield Road, Edinburgh EH9 3JL, Scotland, U.K.

IEEE Log Number 9042026.

1045-9227/91/0300-0302$01.00 © 1991 IEEE

examples to demonstrate the effectiveness of the OLS learning algorithm.

II. The Radial Basis Function Network

A schematic of the RBF network with n inputs and a scalar output is depicted in Fig. 1. Such a network implements a mapping f_r: $\boldsymbol{R}^n \rightarrow \boldsymbol{R}$ according to

$$f_r(\boldsymbol{x}) = \lambda_0 + \sum_{i=1}^{n_r} \lambda_i \phi(\|\boldsymbol{x} - \boldsymbol{c}_i\|) \tag{1}$$

where $\boldsymbol{x} \in \boldsymbol{R}^n$ is the input vector, $\phi(\cdot)$ is a given function from $\boldsymbol{R}^+$ to $\boldsymbol{R}$, $\|\cdot\|$ denotes the Euclidean norm, λ_i, $0 \le i \le n_r$, are the weights or parameters, $\boldsymbol{c}_i \in \boldsymbol{R}^n$, $1 \le i \le n_r$, are known as the RBF centers, and n_r is the number of centers. Although the scalar output case is considered here for notational simplicity, extension to the multioutput case is straightforward. In fact, a multioutput RBF network can always be separated into a group of single-output RBF networks. In the RBF network the functional form $\phi(\cdot)$ and the centers $\boldsymbol{c}_i$ are assumed to have been fixed. By providing a set of the input $\boldsymbol{x}(t)$ and the corresponding desired output $d(t)$ for $t = 1$ to N, the values of the weights λ_i can be determined using the linear LS method. However the choices of $\phi(\cdot)$ and $\boldsymbol{c}_i$ must be carefully considered in order for the RBF network to be able to match closely the performance of the two-layer neural network.

Theoretical investigation and practical results suggest that the choice of the nonlinearity $\phi(\cdot)$ is not crucial to the performance of the RBF network. For example, the thin-plate-spline function

$$\phi(\nu) = \nu^2 \log(\nu) \tag{2}$$

and the Gaussian function

$$\phi(\nu) = \exp(-\nu^2/\beta^2) \tag{3}$$

where β is a real constant, are two typical choices [7], [8]. For the nonlinearity (2) $\phi(\nu) \rightarrow \infty$ as $\nu \rightarrow \infty$, and for (3) $\phi(\nu) \rightarrow 0$ as $\nu \rightarrow \infty$. Although these two nonlinearities have quite different properties, both the resulting RBF networks have good approximation capabilities [8]. Two other common choices [7], [8] of $\phi(\cdot)$ are the multiquadric function

$$\phi(\nu) = (\nu^2 + \beta^2)^{1/2} \tag{4}$$

and the inverse multiquadric function

$$\phi(\nu) = (\nu^2 + \beta^2)^{-1/2}. \tag{5}$$

In practice the centers are normally chosen from the data points $\{\boldsymbol{x}(t)\}_{t=1}^{N}$. The key question is therefore how to select centers appropriately from the data set. A commonly used method to date is to choose arbitrarily some data points as centers. Apparently such a method cannot guarantee adequate performance because it may not satisfy the requirement that centers should suitably sample the input domain. Furthermore, in order to achieve a given performance, an unnecessarily large RBF network may be required. This adds computational complexity and often causes numerical ill-conditioning. Because parameter estimation is frequently ill-conditioned, singular value decomposition [11] and other techniques [12] have to be employed.

Alternatively, the OLS algorithm [10] can be used to select centers so that adequate and parsimonious RBF networks can be obtained. In order to understand how this works, it is essential to view the RBF network (1) as a special case of the linear

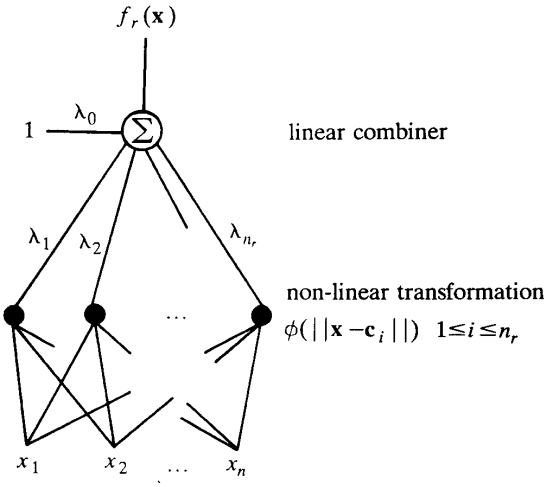

Fig. 1. Schematic of radial basis function network.

regression model

$$d(t) = \sum_{i=1}^{M} p_i(t)\theta_i + \epsilon(t) \tag{6}$$

where $d(t)$ is the desired output and is also called the dependent variable, the θ_i are the parameters, and the $p_i(t)$ are known as the regressors which are some fixed functions of $\boldsymbol{x}(t)$:

$$p_i(t) = p_i(\boldsymbol{x}(t)). \tag{7}$$

The error signal $\epsilon(t)$ is assumed to be uncorrelated with the regressors $p_i(t)$. A constant term can be included in (6) by setting the corresponding $p_i(t) = 1$. It is apparent that a fixed center $\boldsymbol{c}_i$ with a given nonlinearity $\phi(\cdot)$ corresponds to a regressor $p_i(t)$ in (6), and the problem of how to select a suitable set of RBF centers from the data set can be regarded as an example of how to select a subset of significant regressors from a given candidate set. An efficient learning procedure for selecting a subset model from (6) can readily be derived based on the OLS method.

III. Orthogonal Least Squares Learning Algorithm

The geometric interpretation of the LS method is best revealed by arranging (6) for $t = 1$ to N in the following matrix form:

$$\boldsymbol{d} = \boldsymbol{P}\Theta + E \tag{8}$$

where

$$\boldsymbol{d} = [d(1) \cdots d(N)]^T \tag{9}$$

$$\boldsymbol{P} = [\boldsymbol{p}_1 \cdots \boldsymbol{p}_M], \quad \boldsymbol{p}_i = [p_i(1) \cdots p_i(N)]^T, \quad 1 \le i \le M \tag{10}$$

$$\Theta = [\theta_1 \cdots \theta_M]^T \tag{11}$$

$$E = [\epsilon(1) \cdots \epsilon(N)]^T. \tag{12}$$

The regressor vectors $\boldsymbol{p}_i$ form a set of basis vectors, and the LS solution $\hat{\Theta}$ satisfies the condition that $\boldsymbol{P}\hat{\Theta}$ be the projection of $\boldsymbol{d}$ onto the space spanned by these basis vectors. In other words,

the square of the projection $P\hat{\Theta}$ is part of the desired output energy that can be counted by the regressors. Because different regressors are generally correlated, it is not clear how an individual regressor contributes to this output energy.

The OLS method involves the transformation of the set of p_i into a set of orthogonal basis vectors, and thus makes it possible to calculate the individual contribution to the desired output energy from each basis vector. The regression matrix P can be decomposed into

$$P = WA \tag{13}$$

where A is an $M \times M$ triangular matrix with 1's on the diagonal and 0's below the diagonal, that is,

$$A = \begin{bmatrix} 1 & \alpha_{12} & \alpha_{13} & \cdots & & \alpha_{1M} \\ 0 & 1 & \alpha_{23} & \cdots & & \alpha_{2M} \\ 0 & 0 & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & & \\ & & & & 1 & \alpha_{M-1M} \\ 0 & \cdots & & 0 & 0 & 1 \end{bmatrix} \tag{14}$$

and W is an $N \times M$ matrix with orthogonal columns w_i such that

$$W^T W = H \tag{15}$$

where H is diagonal with elements h_i:

$$h_i = w_i^T w_i = \sum_{t=1}^{N} w_i(t)\, w_i(t), \qquad 1 \le i \le M. \tag{16}$$

The space spanned by the set of orthogonal basis vectors w_i is the same space spanned by the set of p_i, and (8) can be rewritten as

$$d = Wg + E. \tag{17}$$

The orthogonal LS solution $\hat{g}$ is given by

$$\hat{g} = H^{-1} W^T d \tag{18}$$

or

$$\hat{g}_i = w_i^T d/(w_i^T w_i), \qquad 1 \le i \le M. \tag{19}$$

The quantities $\hat{g}$ and $\hat{\Theta}$ satisfy the triangular system

$$A\hat{\Theta} = \hat{g}. \tag{20}$$

The classical Gram–Schmidt and modified Gram–Schmidt methods [13] can be used to derive (20) and thus to solve for the LS estimate $\hat{\Theta}$. A similar orthogonal decomposition of P can be obtained using the Householder transformation method [14]. As an illustration, the well-known classical Gram–Schmidt method computes one column of A at a time and orthogonalizes P as follows: at the kth stage make the kth column orthogonal to each of the $k-1$ previously orthogonalized columns and repeat the operation for $k = 2, \cdots, M$. The computational procedure can be represented as

$$\left.\begin{aligned} w_1 &= p_1 \\ \alpha_{ik} &= w_i^T p_k/(w_i^T w_i), \qquad 1 \le i < k \\ w_k &= p_k - \sum_{i=1}^{k-1} \alpha_{ik} w_i \end{aligned}\right\} \quad k = 2, \cdots, M. \tag{21}$$

The OLS method has superior numerical properties compared with the ordinary LS method. Our interest in the OLS method, however, is to use it for subset selection. In the case of RBF networks, the number of data points $x(t)$ is often very large and centers are to be chosen as a subset of the data set. In general the number of all the candidate regressors, M, can be very large and an adequate modeling may only require $M_s (\ll M)$ significant regressors. These significant regressors can be selected using the OLS algorithm operating in a forward regression manner [10]. Because w_i and w_j are orthogonal for $i \neq j$, the sum of squares or energy of $d(t)$ is

$$d^T d = \sum_{i=1}^{M} g_i^2 w_i^T w_i + E^T E. \tag{22}$$

If d is the desired output vector after its mean has been removed, then the variance of $d(t)$ is given by

$$N^{-1} d^T d = N^{-1} \sum_{i=1}^{M} g_i^2 w_i^T w_i + N^{-1} E^T E. \tag{23}$$

It is seen that $\Sigma\, g_i^2 w_i^T w_i/N$ is the part of the desired output variance which can be explained by the regressors and $E^T E/N$ is the unexplained variance of $d(t)$. Thus $g_i^2 w_i^T w_i/N$ is the increment to the explained desired output variance introduced by w_i, and an error reduction ratio due to w_i can be defined as

$$[\mathrm{err}]_i = g_i^2 w_i^T w_i/(d^T d), \qquad 1 \le i \le M. \tag{24}$$

This ratio offers a simple and effective means of seeking a subset of significant regressors in a forward-regression manner. We will again use the classical Gram–Schmidt scheme as an example. The regressor selection procedure is summarized as follows:

□ At the first step, for $1 \le i \le M$, compute

$$\left.\begin{aligned} w_1^{(i)} &= p_i \\ g_1^{(i)} &= (w_1^{(i)})^T d/((w_1^{(i)})^T w_1^{(i)}) \\ [\mathrm{err}]_1^{(i)} &= (g_1^{(i)})^2 (w_1^{(i)})^T w_1^{(i)}/(d^T d) \end{aligned}\right\}.$$

Find

$$[\mathrm{err}]_1^{(i_1)} = \max\left\{[\mathrm{err}]_1^{(i)}, \quad 1 \le i \le M\right\}$$

and select

$$w_1 = w_1^{(i_1)} = p_{i_1}.$$

□ At the kth step where $k \ge 2$, for $1 \le i \le M$, $i \neq i_1, \cdots, i \neq i_{k-1}$, compute

$$\left.\begin{aligned} \alpha_{jk}^{(i)} &= w_j^T p_i/(w_j^T w_j), \qquad 1 \le j < k \\ w_k^{(i)} &= p_i - \sum_{j=1}^{k-1} \alpha_{jk}^{(i)} w_j \\ g_k^{(i)} &= (w_k^{(i)})^T d/((w_k^{(i)})^T w_k^{(i)}) \\ [\mathrm{err}]_k^{(i)} &= (g_k^{(i)})^2 (w_k^{(i)})^T w_k^{(i)}/(d^T d) \end{aligned}\right\}$$

Find

$$[\mathrm{err}]_k^{(i_k)} = \max\left\{[\mathrm{err}]_k^{(i)}, \quad 1 \le i \le M, i \neq i_1, \cdots, i \neq i_{k-1}\right\}$$

and select

$$w_k = w_k^{(i_k)} = p_{i_k} - \sum_{j=1}^{k-1} \alpha_{jk} w_j$$

where $\alpha_{jk} = \alpha_{jk}^{(i_k)}$, $1 \le j < k$.

□ The procedure is terminated at the M_sth step when

$$1 - \sum_{j=1}^{M_s} [\text{err}]_j < \rho \tag{25}$$

where $0 < \rho < 1$ is a chosen tolerance. This gives rise to a subset model containing M_s significant regressors.

In practice the mean of $d(t)$ does not need to be removed because adding a constant to the denominator of the error reduction ratio (24) will not affect the result of maximization in the selection procedure. In any case if $d(t)$ contains a statistically significant mean, a constant term will be selected to model it.

Similar selection procedures can be derived using the modified Gram–Schmidt method and Householder transformation method, and they are given in [10] and [15]. The geometrical interpretation of this OLS procedure is obvious. At the kth step, the dimension of the space spanned by the selected regressors is increased from $k - 1$ to k by introducing one more basis vector. The newly added regressor maximizes the increment to the explained variance of the dependent variable. The orthogonal property makes the whole selection procedure simple and efficient. Given a required level of unexplained variance, it is apparent that this OLS learning procedure will generally produce an RBF network smaller than a randomly selected RBF network. This parsimonious property is a significant advantage of the learning algorithm.

The tolerance ρ is an important instrument in balancing the accuracy and the complexity of the final network. In many signal processing applications the desired value for ρ can actually be learned during the selection procedure. Consider, for example, modeling noisy observations from complex systems. It is apparent from (23) that ρ should ideally be larger than but very close to the ratio $\sigma_\epsilon^2/\sigma_d^2$, where σ_ϵ^2 is the variance of the residuals and σ_d^2 is the variance of the measured or desired outputs. The quantity σ_d^2 is known from the measured data, and during the selection procedure, an estimate of σ_ϵ^2 can be computed. After a few trials, an appropriate estimate for $\sigma_\epsilon^2/\sigma_d^2$ can usually be found. A more detailed discussion and some simulation examples are given in [15]. The criterion (25) emphasizes only the performance of the network (variance of residuals). Because a more accurate performance is often achieved at the expense of using a more complex network, a trade-off between the performance and complexity of the network is often desired. Akaike-type criteria which compromise between the performance and the number of parameters can be written as

$$AIC(\chi) = N \log(\sigma_\epsilon^2) + M_s \chi \tag{26}$$

where χ is the critical value of the chi-squared distribution with one degee of freedom and for a given level of significance. The value $\chi = 4$ corresponds to the significance level of 0.0456 and is often a suitable choice [16]. This kind of statistical test can be combined with the error reduction ratio in such a way that regressors are selected based on (24) as described previously and the selection procedure is automatically terminated when $AIC(\chi)$ reaches its minimum. Other statistical criteria [17] can also be employed to terminate the selection.

A mechanism to avoid numerical ill-conditioning can be built into the OLS learning procedure. The relation $w_k^T w_k = 0$ simply implies that p_k is a linear combination of p_1 to p_{k-1}. Therefore if $w_k^T w_k$ is less than a small preset threshold, the regressor p_k will not be selected. This has an important implication for fitting RBF networks. When the centers are arbitrarily chosen, it is frequently found that the LS problem is ill-conditioned, and more robust techniques such as singular value decomposition [11] must be used. Such a numerical problem is easily overcome in the OLS selection algorithm.

IV. Application Examples

The application of the OLS learning algorithm to RBF networks is illustrated using examples taking from two different areas of signal processing.

A. Modeling Time Series Data

Most real-world processes are nonlinear to some extent, and in many practical applications nonlinear models may be required in order to achieve an acceptable predictive accuracy. Nonlinear system identification using RBF networks has been investigated by Chen *et al.* [18], [19]. In the current study, a RBF network is employed to model the time series of monthly unemployment figures in West Germany from January 1948 to December 1980. These 396 observations of the series can be found in [20, appendix D], and they are plotted in Fig. 2. Let $y(t)$ be the current time series value. The idea is to use the RBF network

$$\hat{y}(t) = f_r(x(t)) \tag{27}$$

as the one-step-ahead predictor for $y(t)$, where the inputs to the RBF network

$$x(t) = [y(t-1) \cdots y(t-n_y)]^T \tag{28}$$

are past observations of the series.

The nonlinearity $\phi(\cdot)$ within the RBF network was chosen to be the thin-plate-spline function (2). Because it is known that there was a seasonality (twelve-month period) and a linear trend in the series, the lag n_y was set to 13. The RBF centers were to be chosen from the data points $\{x(t)\}$, and this gave rise to a total of about 400 candidates. The OLS learning procedure described in Section III was used to identify a RBF network model. During the selection it was found that an appropriate tolerance was $\rho = 0.003$. A constant term and 55 centers were chosen, and the final RBF network model can be written as

$$\hat{y}(t) = \hat{\lambda}_0 + \sum_{i=1}^{55} \hat{\lambda}_i \phi(\|x(t) - c_i\|)$$

with

$$\|x(t) - c_i\|^2 = \sum_{k=1}^{13} (y(t-k) - c_{ki})^2, \qquad 1 \le i \le 55$$

where $\phi(\cdot)$ is defined in (2) and c_i are chosen centers. The one-step-ahead prediction $\hat{y}(t)$ looks essentially the same as $y(t)$ shown in Fig. 2, and the prediction error or residual

$$\epsilon(t) = y(t) - \hat{y}(t) \tag{29}$$

is depicted in Fig. 3.

The autocorrelations of $\{\epsilon(t)\}$ are plotted in Fig. 4. It is well known that the autocorrelation test alone is not sufficient to validate a nonlinear time series model. Additional model validity tests were used based on the chi-squared statistical test [16],

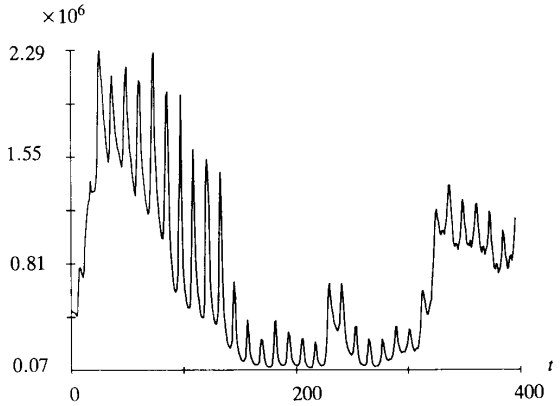

Fig. 2. Time series of monthly unemployment figures in West Germany.

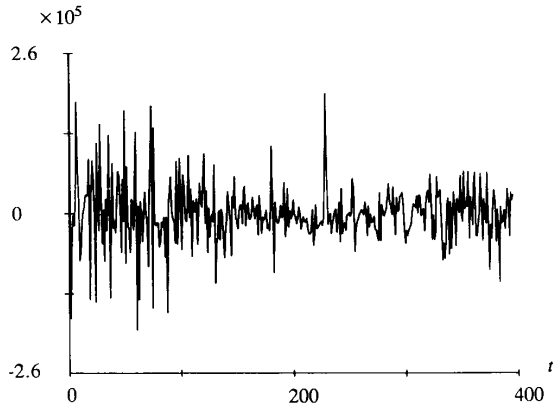

Fig. 3. Residual sequence.

[21]. Define $\Omega(t)$ as an η-dimensional vector-valued function

$$\Omega(t) = [\omega(t)\ \omega(t-1) \cdots \omega(t-\eta+1)]^T \quad (30)$$

where $\omega(t)$ is some chosen function of the past observations and prediction errors, and let

$$\Gamma^T\Gamma = N^{-1} \sum_{t=1}^{N} \Omega(t)\,\Omega^T(t). \quad (31)$$

The chi-squared statistic is calculated according to

$$\zeta = N\mu^T(\Gamma^T\Gamma)^{-1}\mu \quad (32)$$

where

$$\mu = N^{-1} \sum_{t=1}^{N} \Omega(t)\,\epsilon(t)/\sigma_\epsilon \quad (33)$$

and σ_ϵ^2 is the variance of the residual $\epsilon(t)$. Several chi-squared tests for the selected RBF network were calculated and they were all within a 95% confidence band. Fig. 5 shows two typical chi-squared tests. The model validity tests confirm that this RBF network is an adequate model for the time series.

Whenever there are sufficient data points, the data should be divided into a fitting set and a testing set. The former is used in the selection procedure and the latter is used to validate the selected network. This provides an interpretation for the two fundamental concepts in neural networks, namely learning and generalization. Learning can be viewed as producing a surface in multidimensional space that fits the set of data in some best sense, and generalization is then equivalent to interpolating the test data set on this fitting surface. This approach was not applied to the above example because the data set was not sufficiently large and the data were highly nonstationary. However some application examples and further discussion on this issue can be found in [9], [18], and [19].

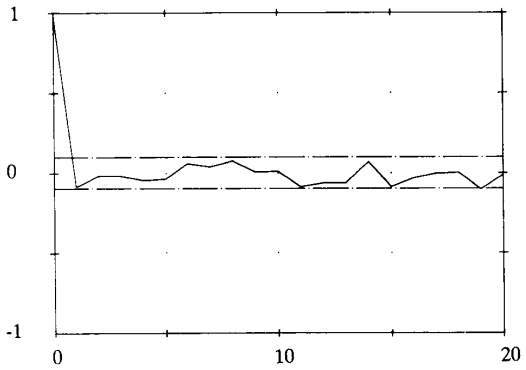

Fig. 4. Autocorrelations of residuals. —·— 95% confidence band.

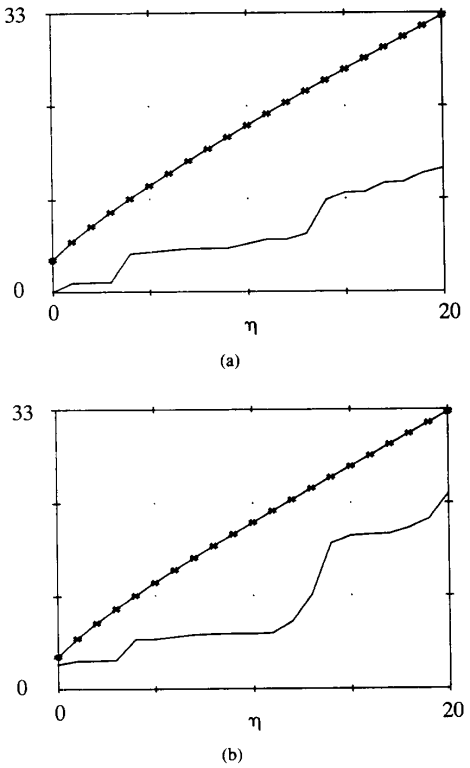

Fig. 5. Chi-squared tests: (a) $\omega(t) = \epsilon^2(t-1)$; (b) $\omega(t) = \epsilon^2(t-1)y^2(t-1)$. —#— 95% confidence band.

B. Communications Channel Equalization

The digital communications system considered is illustrated in Fig. 6, where a binary sequence $s(t)$ is transmitted through a dispersive channel and then corrupted by additive noise. The transmitted symbol $s(t)$ is assumed to be an independent sequence taking values of either 1 or -1 with an equal probability. The channel is commonly modeled as a finite impulse response filter whose transfer function is defined by

$$C_n(z) = \sum_{i=0}^{n} c_i z^{-i}. \tag{34}$$

The additive noise $e(t)$ is assumed to be a white Gaussian sequence. The task of the equalizer is to reconstruct input signals using the information contained in the channel observations

$$\mathbf{y}(t) = \left[y(t) \cdots y(t-m+1)\right]^T \tag{35}$$

where the integer m is known as the order of the equalizer. Often a delay τ is introduced to the equalizer so that at the sample instant t the equalizer estimates the input symbol $s(t-\tau)$. Traditionally the equalization problem defined in Fig. 6 is viewed as an inverse filtering [22] in which the equalizer forms an approximation to the inverse of the distorting channel (34). Thus the classical equalizer is constructed as a linear finite filter followed by a decision slicer. The equalization problem is, however, an inherently nonlinear one and nonlinear filters are required in order to achieve fully or nearly optimal performance [23].

It can be shown that the minimum bit-error-rate equalizer is defined as follows [23].

$$\hat{s}(t-\tau) = \operatorname{sgn}\left(f_o(\mathbf{y}(t))\right) \tag{36}$$

where

$$\operatorname{sgn}(y) = \begin{cases} 1, & y \geq 0 \\ -1, & y < 0 \end{cases} \tag{37}$$

represents a slicer. $f_o(\cdot)$, known as the optimal decision function, is specified by the channel transfer function $C_n(z)$ and the channel noise distribution together with the equalizer order m and delay τ. The set of points $\mathbf{y}$ that satisfy

$$f_o(\mathbf{y}) = 0 \tag{38}$$

is often referred to as the optimal decision boundary, which partitions the m-dimensional space into two sets Y_1 and Y_{-1}. The decision rule

$$\hat{s}(t-\tau) = \begin{cases} 1, & \mathbf{y}(t) \in Y_1 \\ -1, & \mathbf{y}(t) \in Y_{-1} \end{cases} \tag{39}$$

results in the minimum error probability. $f_o(\cdot)$ is generally not known *a priori* and is certainly nonlinear. A RBF network can be trained to realize or to approximate this optimal solution. During the training the error signal is defined as

$$\epsilon(t) = s(t-\tau) - f_r(\mathbf{y}(t)). \tag{40}$$

In a previous investigation [23] it was found that the RBF centers critically influence the performance of the RBF equalizer. RBF equalizers constructed by arbitrarily choosing some data points as centers are often inadequate. The OLS learning strategy offers an ideal solution to the construction of RBF equalizers and a computer simulation study was carried out to demonstrate this. The nonlinearity $\phi(\cdot)$ for the RBF network was chosen as the Gaussian function (3), where the constant β was chosen to be 1.

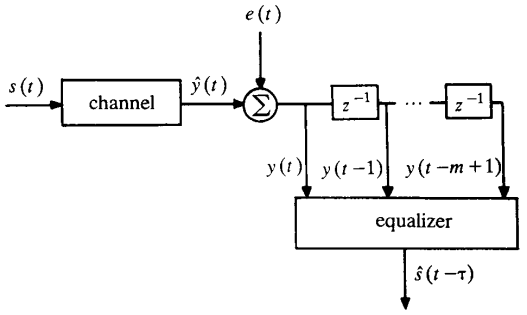

Fig. 6. Schematic of data transmission system.

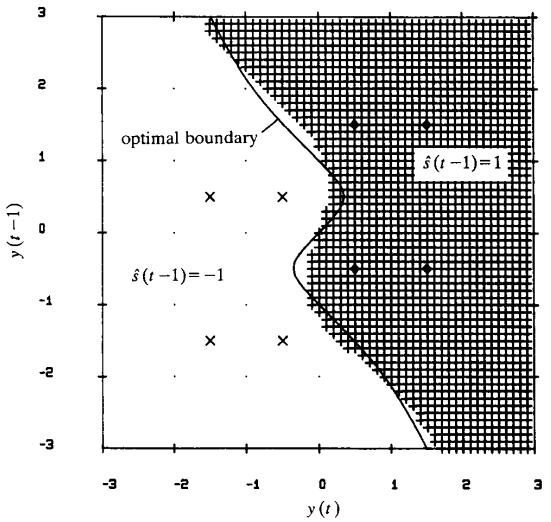

Fig. 7. Decision regions formed by RBF equalizer (12 centers selected by OLS algorithm).

In the first example, the channel transfer function was given by

$$C_1(z) = 0.5 + 1.0z^{-1}$$

and the additive white Gaussian noise had variance 0.2. The equalizer had a structure of $m = 2$ and $\tau = 1$. A training sequence of 400 data points was generated and an RBF network of 12 centers was selected using the OLS learning procedure. Fig. 7 illustrates how this RBF equalizer partitions the input space where the shaded region represents $f_r(\mathbf{y}(t)) \geq 0$. The four squares (□) and four crosses (×) in Fig. 7 represent all the possible channel output points in the noise-free case. RBF equalizers obtained using an arbitrary selection of centers can only achieve a similar accuracy by significantly increasing the number of centers. Fig. 8 shows the decision regions of a typical RBF equalizer with 50 randomly selected centers.

The second example compares the bit error rates achieved by the optimal and RBF equalizers for a variety of signal to noise ratios. The channel transfer function was

$$C_2(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}.$$

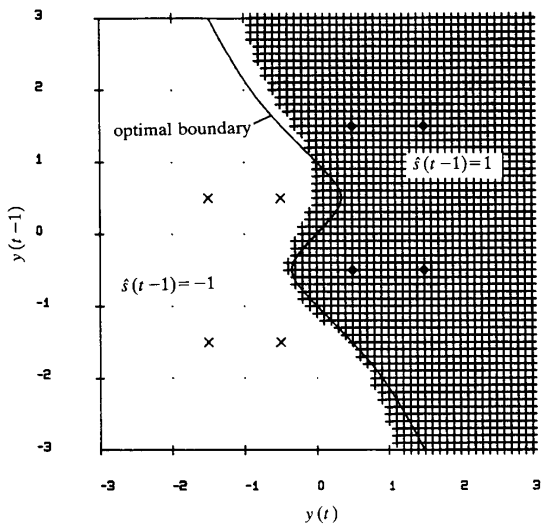

Fig. 8. Decision regions formed by RBF equalizer (50 centers selected randomly).

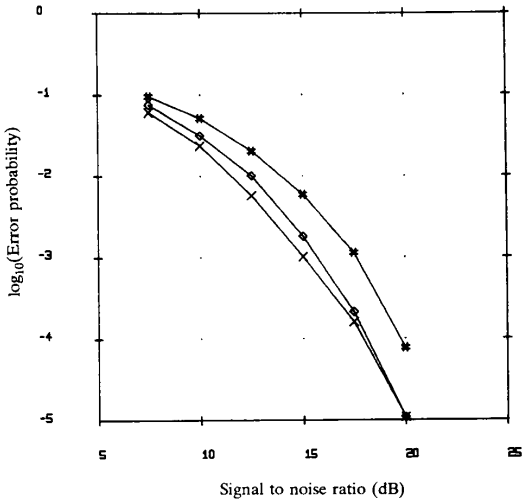

Fig. 9. Performance comparison: —×— optimal equalizer; —⊟— RBF equalizer with 70 centers selected by OLS algorithm; —#— RBF equalizer with 70 centers selected randomly.

The equalizer order was $m = 4$ and delay $\tau = 1$. For each signal-to-noise ratio tested, 600 points of training data were generated and an RBF network of 70 centers was selected using the OLS learning procedure. The performance of this RBF equalizer is depicted in Fig. 9, where the test sequence for computing each error probability with a given signal-to-noise ratio had more than half a million data points. It is seen that the bit-error-rate curve of this equalizer closely matches to the optimal one. For comparison, the performance of a typical RBF equalizer with 70 centers selected randomly from the training data set is also shown in Fig. 9. It is seen that the OLS learning procedure indeed offers much better performance over a random selection of centers.

The OLS learning algorithm is essentially a block-data algorithm. In many adaptive applications it is desired to update filter parameters as each sample signal is collected. In such a situation it is convenient first to select an RBF network using the OLS learning algorithm based on a block of training data and then continuously to adapt the weights of the linear combiner within the selected RBF network using, for example, the least mean squares algorithm. This allows an RBF network to operate in a time-varying environment.

V. Conclusions

The crucial question of how to select radial basis function centers from the data points has been investigated and a learning strategy based on the orthogonal least squares algorithm has been developed for the construction of radial basis function networks. Operating in a forward regression procedure, the orthogonal least squares algorithm provides a systematic approach to the selection of RBF centers, one which is far superior to a random selection of centers. It has been shown that this learning strategy offers a powerful procedure for fitting adequate and parsimonious radial basis function networks in practical signal processing.

Some researchers have also recognized the inadequacy of randomly selecting radial basis function centers. Moody and Darken [24] proposed a novel approach which requires two presentations of the training data. In the first pass of the training data a clustering technique was employed to choose the centers, and the connection weights were then computed as usual based on the least squares criterion in the second pass. The orthogonal least squares learning algorithm requires only one pass of the training data and the selection of centers is directly linked to the reduction of error signals. Further work will be carried out to compare the performance of these two learning algorithms.

Acknowledgment

The authors wish to thank the reviewers for their valuable comments on the manuscript.

References

[1] G. Cybenko, "Approximations by superpositions of a sigmoidal function," *Math. Contr., Signals Syst.*, vol. 2, no. 4, pp. 303–314, 1989.
[2] K. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Networks*, vol. 2, pp. 183–192, 1989.
[3] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA.: Addison-Wesley, 1989.
[4] K. S. Narendra and M. A. L. Thathachar, *Learning Automata—An Introduction*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
[5] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.
[6] C. A. Micchelli, "Interpolation of scattered data: distance matrices and conditionally positive definite functions," *Construct. Approx.*, vol. 2, pp. 11–22, 1986.
[7] 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, 1987, pp. 143–167.
[8] M. J. D. Powell, "Radial basis function approximations to polynomials," in *Proc. 12th Biennial Numerical Analysis Conf.* (Dundee), 1987, pp. 223–241.
[9] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Syst.*, vol. 2, pp. 321–355, 1988.

[10] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *Int. J. Contr.*, vol. 50, no. 5, pp. 1873–1896, 1989.
[11] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numerische Mathematik*, vol. 14, pp. 403–420, 1970.
[12] N. Dyn, "Interpolation of scattered data by radial functions," in *Topics in Multivariate Approximation*, C. K. Chui, L. L. Schumaker and F. I. Utreras, Eds. New York: Academic Press, 1987, pp. 47–61.
[13] A. Björck, "Solving linear least squares problems by Gram–Schmidt orthogonalization," *Nordisk Tidskr. Informations-Behandling*, vol. 7, pp. 1–21, 1967.
[14] G. Golub, "Numerical methods for solving linear least squares problems," *Numerische Mathematik*, vol. 7, pp. 206–216, 1965.
[15] S. A. Billings and S. Chen, "Extended model set, global data and threshold model identification of severely non-linear systems," *Int. J. Contr.*, vol. 50, no. 5, pp. 1897–1923, 1989.
[16] I. J. Leontaritis and S. A. Billings, "Model selection and validation methods for nonlinear systems," *Int. J. Contr.*, vol. 45, no. 1, pp. 311–341, 1987.
[17] T. Söderström, "On model structure testing in system identification," *Int. J. Contr.*, vol. 26, no. 1, pp. 1–18, 1977.
[18] S. Chen, S. A. Billings, C. F. N. Cowan, and P. M. Grant, "Non-linear systems identification using radial basis functions," *Int. J. Syst. Sci.*, vol. 21, no. 12, pp. 2513–2539, 1990.
[19] S. Chen, S. A. Billings, C. F. N. Cowan, and P. M. Grant, "Practical identification of NARMAX models using radial basis functions," *Int. J. Contr.*, vol. 52, no. 6, pp. 1327–1350, 1990.
[20] T. Subba Rao and M. M. Gabr, *An Introduction to Bispectral Analysis and Bilinear Time Series Models* (Lecture Notes in Statistics). New York: Springer-Verlag, 1984.
[21] T. Bohlin, "Maximum-power validation of models without higher-order fitting," *Automatica*, vol. 4, pp. 137–146, 1978.
[22] S. U. H. Qureshi, "Adaptive equalization," *Proc. IEEE*, vol. 73, no. 9, pp. 1349–1387, 1985.
[23] S. Chen, G. J. Gibson, C. F. N. Cowan, and P. M. Grant, "Reconstruction of binary signals using an adaptive radial-basis-function equalizer," *Signal Process.*, vol. 22, no. 1, 1991.
[24] J. Moody and C. Darken, "Fast-learning in networks of locally-tuned processing units," *Neural Comput.*, vol. 1, no. 2, pp. 281–294, 1989.