LEARNING WITH MULTI-OBJECTIVE CRITERIA

G.P. Liu and V. Kadirkamanathan

University of Sheffield, UK

Abstract:- This paper presents a new algorithm for learning with neural networks based on *multiobjective performance criteria*. It considers three performance indices (or cost functions) as the objectives, which are the Euclidean distance and maximum difference measurements between the real nonlinear system and the nonlinear model (L_2, L_∞-norms) , and the complexity measure of the nonlinear model, instead of a single performance index. An algorithm based on the method of inequalities, least squares and genetic algorithms is developed for optimising over the multi-objective criteria. Genetic algorithms are also used simultaneously for model selection in which the structure of the neural networks are determined. The Volterra polynomial basis function network and the Gaussian radial basis function network are applied to the identification of a liquid level nonlinear system.

INTODUCTION

The learning problem, where an unknown underlying nonlinear function is to be found that maps a set of inputs to a set of outputs, can be posed as a function approximation problem. It is well known that the polynomial and many other approximation schemes can approximate arbitrary well a continuous function [10]. In recent years, neural networks have emerged as an alternative [3], but are similar, to these schemes. Much of the learning with neural networks is carried out with a single performance index (or cost function), the most common being the *mean square error* or the L_2-norm (Euclidean distance) of the difference between the underlying model and the network approximation. This results in the well known *least squares* algorithm, chosen often for its computational simplicity. The assumption behind choosing the L_2-norm is that the noise in the process and measurements have Gaussian (normal) distributions.

A commonly adopted approach where a single performance index, the joint sum of the L_2-norm difference and some model complexity measure, is optimised avoids the problem of *overfitting* in a selected network. This improves generalisation in a chosen model. However, it does not indicate if the model approximation is the best that can be achieved. The problem of comparing several models, such as *Bayesian model selection* [9], *Minimum Description Length* (MDL) [12], have also been developed. These procedures allow the selection of the best amongst a small number of candidate models [9].

In this paper, we extend the above ideas in two directions. Firstly, we define a multi-objective criteria to increase the robustness to learning, where the mean squared error (L_2-norm), the maximum error (L_∞-norm) and a model complexity measure are minimised. These objectives can sometimes be conflicting and no solution may exist that optimises all the objectives. Hence, an inevitable trade-off has to be made. Secondly, we develop an algorithm to select a subset from a larger set of basis functions that is optimal in the multi-objective sense. This is equivalent to selecting a model amongst a large number of candidate models determined by all the possible subset combinations of the basis functions. The algorithm is demonstrated using two types of neural networks, namely, the Volterra polynomial basis function (VPBF) network and the Gaussian radial basis function (GRBF) network.

NEURAL NETWORKS

Let the underlying process generating the input – output observations be

$$y = f^*(\mathbf{x}) + \eta \qquad (1)$$

where $y \in \Re$ is the output, $\mathbf{x} \in \Re^M$ is the input, η is the noise with unknown distribution and $f^*(.)$ is the unknown underlying nonlinear function that needs to be learned or estimated. Neural networks (NN) are candidate models to approximate the unknown nonlinear functions based on the observations. Its functional form is given by [7],

$$f(\mathbf{x};\mathbf{p}) = \sum_{k=1}^{K} w_k \; g_k(\mathbf{x};\mathbf{d}_k) \qquad (2)$$

where $\mathbf{p}$ is the set of parameters in the model or network, w_k are the coefficients, $g_k(.)$ are the basis functions (formed at the hidden layer in a single hidden layer NN), $\mathbf{d}_k$ are the parameters in the kth basis function (input – hidden layer weights in NN) and K denotes the number of basis functions. The form of the network depends on the basis functions. For example, the sigmoidal function is used in the *multilayer perceptron*. Here, we use networks with the following basis functions: the Volterra polynomial basis functions (VPBF) and the Gaussian radial basis functions (GRBF).

Multivariate (Volterra) polynomial expansions [14], well known in function approximation, has been cast into the framework of nonlinear system approximations and neural networks. A second order Volterra polynomial expansion is given by,

$$\begin{aligned} f^*(\mathbf{x},\mathbf{p}) &= a + \mathbf{x}^T\mathbf{b} + \mathbf{x}^T\mathbf{C}\mathbf{x} \\ &= \sum_{k=1}^{K} w_k g_k(\mathbf{x}) \end{aligned} \tag{3}$$

where,

$$[w_1, w_2, w_3, ..., w_{M+2}, w_{M+3}, ..., w_{K_0}] = \\ [a, b_1, b_2, ..., c_{11}, c_{12}, c_{22} ..., c_{MM}] \tag{4}$$

$$[g_1, g_2, g_3, ..., g_{M+2}, g_{M+3}, g_{M+4}, ..., g_{K_0}] = \\ [1, x_1, x_2, ..., x_1^2, x_1x_2, x_2^2, ..., x_M^2] \tag{5}$$

are the set of linear weights and the set of basis functions being linearly combined, respectively.

Radial basis functions (RBF) were introduced as a technique for multivariable interpolation [11], which can be cast into an architecture similar to that of the multilayer perceptron [1]. RBF networks provide an alternative to the traditional NN architectures and have good approximation properties. One of the commonly used RBF networks is the *Gaussian radial basis function (GRBF) network*, also called the localised receptive field network. The function mapped by the GRBF model is given by,

$$f^*(\mathbf{x},\mathbf{p}) = \sum_{k=1}^{K} w_k \exp\left\{-(\mathbf{x}-\mathbf{d}_k)^T \mathbf{C}_k (\mathbf{x}-\mathbf{d}_k)\right\} \tag{6}$$

where $\mathbf{C}_k$ is the weighting matrix of the kth basis function whose centre is $\mathbf{d}_k$, which can transform the equidistant lines from being hyperpherical to the hyperellipsoidal, $\mathbf{C}_k = \mathbf{I}$ in this paper, and $\mathbf{p}$ is the parameter vector containing w_k and $\mathbf{d}_k$ $(k = 1, 2, ..., K)$.

MODEL SELECTION

Genetic algorithms (GA) are search procedures which emulate the natural genetics [4]. They are different from traditional search methods encountered in engineering optimisation in following ways: (a) The GA searches from a population of points, not a single point and (b) the GA uses probabilistic and not deterministic transition rules. GAs have been succesfully used with neural networks to determine the network parameters and structure [13], with NARMAX models [5] and for nonlinear basis function selection and RBF centre selection using Bayesian criteria [8]. This paper applies the GA approach to the model selection and identification of nonlinear systems using multiobjective criteria as the basis for selection.

Model selection is carried out with the GA where each model is expressed by a K_0-bit binary model code $\mathbf{c}$, *ie.*, a chromosome representation in GA. The 1 bits of the binary model code $\mathbf{c}$ relate to the selected subset of the basis functions from the set and the 0 relate to the omitted ones. For example, if

$$\mathbf{G} = [g_1, g_2, ..., g_{K_0}] \tag{7}$$

is the set of basis functions with g_k being the individual basis functions. and if the binary model code is $\mathbf{c} = [1 \;\; 0 \;\; 0 \;\; 1 \;\; 0 \;\; 0 \;\; 1 \;\; 0 \;\; \ldots]$, the coded model can be written as,

$$f(\mathbf{x};\mathbf{p}) = w_1 g_1(\mathbf{x}) + w_4 g_4(\mathbf{x}) + w_7 g_7(\mathbf{x}) + \ldots \tag{8}$$

Hence model selection can be seen as a subset selection problem. For the model represented by the VPBF, with $\mathbf{x} \in \Re^M$, the maximum number of the model terms is given by $K_0 = (M+1)(M+2)/2$ and there are 2^{K_0} possible models for selection.

For the model represented by GRBF, the maximum number of the model terms is given by K_0, the total number of the Gaussian RBFs and there are 2^{K_0} possible models for selection. Also, there are K_0 RBF centre parameters $\mathbf{d}_k$. Thus a chromosome reprentation in genetic algorithms consists of a K_0-bit binary model code $\mathbf{c}$ and real basis function centres $\mathbf{d}_k$ $(k = 1, 2, ..., K_0)$, *ie.*,

$$[\mathbf{c}, \mathbf{d}_1^T, \mathbf{d}_2^T, \ldots, \mathbf{d}_{K_0}^T] \tag{9}$$

Only the basis functions corresponding to the nonzero bits of the binary model code $\mathbf{c}$ are included in the selected model. Given a parent set of binary model codes and basis function parameter vectors, the model chosen is one that optimises some performance criteria.

MULTIOBJECTIVE CRITERIA

Let us define the following performance functions:

$$\phi_1(\mathbf{p}) = \|f(\mathbf{x}) - f^*(\mathbf{x},\mathbf{p})\|_2 \tag{10}$$

$$\phi_2(\mathbf{p}) = \|f(\mathbf{x}) - f^*(\mathbf{x},\mathbf{p})\|_\infty \tag{11}$$

$$\phi_3(\mathbf{p}) = \sigma(\mathbf{c}) \tag{12}$$

where $\|.\|_2$ and $\|.\|_\infty$ are the 2- and ∞-norms of the function (.), $\sigma(\mathbf{c})$ is the number of the non-zero elements in the binary model code $\mathbf{c}$.

For learning in nonlinear systems, there are good reasons for giving attention to the performance functions $\phi_i(\mathbf{p})$, $(i = 1, 2, 3)$. The practical reasons for considering the performance function $\phi_1(\mathbf{p})$ are even stonger than the other two, $\phi_2(\mathbf{p})$ and $\phi_3(\mathbf{p})$. Statistical considerations show that it is the most appropriate choice for data fitting when errors in the data have a normal distribution. Often the performance function $\phi_1(\mathbf{p})$ is preferred for its computational simplicity in solving the estimation problem.

The performance function $\phi_2(\mathbf{p})$ is at the foundation of much of approximation theory. It is known from approximation theory that when $\phi_2(\mathbf{p})$ is small, the performance function $\phi_1(\mathbf{p})$ is small also. But the converse statement may not be true. Also, the performance function $\phi_2(\mathbf{p})$ represents the accuracy bound of the approximation achieved by the estimated model. Such a bound may be necessarily enforced by some desired approximation accuracy in some approximation tasks. Using $\phi_2(\mathbf{p})$ can also be justified on statistical grounds if the noise η has a uniform distribution.

The performance function $\phi_3(\mathbf{p})$ is used as a measure of the model complexity. This measure is proportional to the complexity measure used in the *Akaike Information Criterion.* Small performance function $\phi_3(\mathbf{p})$ indicates a simple model. The reason for choosing $\phi_3(\mathbf{p})$ is that under similar performances in $\phi_1(\mathbf{p})$ and $\phi_2(\mathbf{p})$ by two models, the simpler model is statistically likely to be a better model (due to reduced degree of freedom in fitting the data). It is important to note that alternative criteria or additional criteria may equally well be used considering different noise distributions and complexity measures. For example, the L_1-norm could be used as a performance function and the criteria used in Minimum Description Length [12] or one based on Bayesian statistics [9] could be used as a measure for model complexity. The criteria selected in this paper are principally to demonstrate the principle and its application.

If one of the performance functions ϕ_i $(i = 1, 2, 3)$ is minimized individually (single-objective approach), then unacceptably large values may result for other performance functions ϕ_j $(j \neq i,\ j = 1, 2, 3)$. Generally, there does not exist a solution for all performance function $\phi_i(\mathbf{p})$ for $i = 1, 2, 3$ to be minimized by the same parameter vector $\mathbf{p}$. Following the method of inequalities [15], we reformulate the optimization into a multiobjective problem as,

$$\phi_i(\mathbf{p}) \leq \varepsilon_i, \quad \text{for } i = 1, 2, 3 \tag{13}$$

where the positive real number ε_i represents the numerical bound on the performance function $\phi_i(\mathbf{p})$ and is determined by the designer.

OPTIMISATION PROCEDURE

As we are concerned with three objectives (or cost functions) for model selection and estimation, this section develops the optimisation procedure based on the method of inequalities [15]. Let us normalise the multiobjective performance functions as follows:

$$\psi_i(\mathbf{p}) = \begin{cases} \dfrac{\phi_i(\mathbf{p})}{\varepsilon_i}, & \text{for } \varepsilon_i \neq 0 \\ \phi_i(\mathbf{p}) + 1, & \text{for } \varepsilon_i = 0 \end{cases} \tag{14}$$

Let Γ_i be the set of parameter vectors $\mathbf{p}$ for which the ith performance criterion is satisfied:

$$\Gamma_i = \{\mathbf{p} : \psi_i(\mathbf{p}) \leq 1\}. \tag{15}$$

Then the admissible or feasible set of parameter vectors for which all the performance criteria hold is the intersection

$$\Gamma = \Gamma_1 \cap \Gamma_2 \cap \Gamma_3. \tag{16}$$

Clearly, $\mathbf{p}$ is an admissible parameter vector if and only if

$$\max\{\psi_1(\mathbf{p}), \psi_2(\mathbf{p}), \psi_3(\mathbf{p})\} \leq 1. \tag{17}$$

which shows that the search for an admissible $\mathbf{p}$ can be pursued by optimization, in particular by solving

$$\min_{\mathbf{p}}\{\max\{\psi_1(\mathbf{p}), \psi_2(\mathbf{p}), , \psi_3(\mathbf{p})\}\} \leq 1. \tag{18}$$

Now, let $\mathbf{p}^n$ be the value of the parameter vector at the nth step, and define

$$\Gamma_i^n = \{\mathbf{p} : \psi_i(\mathbf{p}) \leq \Delta^n\}, \quad \text{for } i = 1, 2, 3, \tag{19}$$

where

$$\Delta^n = \max\{\psi_i(\mathbf{p}^n)\} \tag{20}$$

and also define

$$\Gamma^n = \Gamma_1^n \cap \Gamma_2^n \cap \Gamma_3^n, \tag{21}$$

$$E^n = \psi_1(\mathbf{p}^n) + \psi_2(\mathbf{p}^n) + \psi_3(\mathbf{p}^n). \tag{22}$$

Γ^n is the nth set of parameter vectors for which all performance functions satisfy

$$\psi_i(\mathbf{p}) \leq \Delta^n, \quad \text{for } i = 1, 2, 3. \tag{23}$$

It is clear that Γ^n contains both $\mathbf{p}^n$ and the admissible set Γ. E^n is a combined measurement of all performance functions. If we find a new parameter vector $\bar{\mathbf{p}}^n$, such that

$$\bar{\Delta}^n < \Delta^n, \tag{24}$$

or

$$\bar{\Delta}^n = \Delta^n \quad \text{and} \quad \bar{E}^n < E^n, \tag{25}$$

where $\bar{\Delta}^n$ and $\bar{E}^n$ are defined similarly to Δ^n and E^n, then we accept $\bar{\mathbf{p}}^n$ as the next estimate of the parameter vector. Then, setting $\mathbf{p}^{n+1} = \bar{\mathbf{p}}^n$ gives,

$$\psi_i(\mathbf{p}^{n+1}) \leq \psi_i(\mathbf{p}^n), \quad \text{for } i = 1, 2, 3 \tag{26}$$

and

$$\Gamma \subset \Gamma^{n+1} \subset \Gamma^n \tag{27}$$

so that the boundary of the set in which the parameters are located has been moved towards the admissible set, or rarely, has remained unaltered. The process of finding the optimal solution is terminated only when both Δ_n and E^n cannot be reduced any further. But the process of finding an admissible parameter vector $\mathbf{p}$ can be terminated when

$$\Delta^n \leq 1, \tag{28}$$

ie., when the boundaries of Γ^n have converged to the boundaries of Γ. If the Δ^n persists in being larger than 1, this may be taken as an indication that the performance criteria may be inconsistent, whilst their magnitude gives some measure of how closely it is possible to approach the objectives. In this case, some of the performance criteria should be relaxed until they are satisfied. From a practical viewpoint, the approximate optimal solution is also useful if the optimal solution is not achievable. GA have been used in multi-objective optimisation and have provided better results over conventional search methods [4], [6]. Here, we combine GA with that of least squares in deriving the estimation algorithm.

NUMERICAL ALGORITHM

The steps of the learning algorithm to be executed for the GA implementation are as follows:

Step 1: Chromosomal representation
Each chromosome in the population consists of an K_0-bit binary model code $\mathbf{c}$ and a real number basis function parameter vector $\mathbf{D}$, where K_0 is the total number of the basis functions in the class of models. For example, for the VPBF model the $\mathbf{D}$ is null and for the GRBF model the vector $\mathbf{D}$ contains all basis function centres $\mathbf{d}_k$ $(k = 1, 2, ..., K_0)$

Step 2: Generation of the initial population
The L chromosomes $[\mathbf{c}, \mathbf{D}]$ for the initial population are randomly generated, where L is an odd number.

Step 3: Evaluation of the performance functions
Given the j-th binary model code $\mathbf{c}_j$ and basis function parameter vector $\mathbf{D}_j$, then the j-th nonlinear model is known. Using the least squares algorithm, compute the j-th weight vector $\mathbf{w}_j$ for each of the models. Then evaluate the normalised performance functions $\psi_i(\mathbf{p}_j)$ $(i = 1, 2, 3)$, where $\mathbf{p}_j = [\mathbf{w}_j, \mathbf{c}_j, \mathbf{D}_j]$,

$$\Delta_j = \max_{i=1,2,3} \psi_i(\mathbf{p}_j), \tag{29}$$

$$E_j = \sum_{i=1}^{3} \psi_i(\mathbf{p}_j), \tag{30}$$

Complete the above computations for all L sets of chromosomes, *ie.*, $j = 1, 2, ..., L$.

Step 4: Selection
According to the fitness of the performance functions for each chromosome, delete the $(L-1)/2$ weaker members of the population and reorder the chromosomes. The fitness of the performance functions is,

$$F_j = \frac{1}{\Delta_j}, \quad \text{for } j = 1, 2, \ldots, L. \tag{31}$$

Step 5: Crossover
Offspring binary model codes are produced from two parent binary model codes so that their first half elements are preserved. The second half elements in each parent are exchanged. The average crossover operator is used to produce offspring basis function parameter vectors and is defined as,

$$\frac{\mathbf{D}_{j+1} + \mathbf{D}_j}{2}, \quad \text{for } j = 1, 2, \ldots, (L-1)/2. \tag{32}$$

Then the $(L-1)/2$ offsprings are produced.

Step 6: Mutation
A mutation operator, called a creep, is used. For the binary model codes, it randomly replaces one bit in each offspring binary model code with a random number 1 or 0. For the offspring basis function parameter vectors, the mutation operation is,

$$\mathbf{D}_j + \beta\xi_j\mathbf{I}, \quad \text{for } j = 1, 2, \ldots, (L-1)/2, \tag{33}$$

where β is the maximum value adaptable and $\xi_j \in [-1, 1]$ is a random variable with zero mean and $\mathbf{I}$ is the identity matrix.

Step 7: Elitism
The elitist strategy copies the best chromosome into the succeeding generation to prevent it being lost in the next generation. The best chromosome is defined as the one satisfying

$$E_b = \min_{l \in \{1,2,\ldots,L\}} \{ \; E_l : E_l \leq E_m - \alpha(\Delta_l - \Delta_m) \quad \text{and} \quad \Delta_l \leq \Delta_m + \delta\} \tag{34}$$

where,

$$\Delta_m = \min_{j=1,2,\ldots,L} \{\Delta_j\}, \tag{35}$$

E_m and E_l are corresponding to Δ_m and Δ_l, which are defined in equations (30) and (29), $\alpha > 1$ and δ is a positive number, specified by the designer (*eg.*, $\alpha = 1.1$ and $\delta = 0.1$).

Step 8: New offsprings
Add the $(L-1)/2$ new offsprings to the population which are generated in a random fashion. Actually, the new offsprings are formed by replacing randomly some elements of the best binary model code and

mutating the best basis function parameter vector.

Step 9: Stop check
Continue the cycle initiated in Step 3 until convergence is achieved. The population is considered to have converged when

$$\Delta_j - \Delta_b \leq \varepsilon \quad \text{for } j = 1, 2, \ldots, (L-1)/2, \tag{36}$$

where Δ_b corresponds to E_b, and $\varepsilon > 0$.

EXAMPLE: SYSTEM IDENTIFICATION

In the example, the task is to identify a system based on the data generated by a large pilot scale liquid level nonlinear system with zero mean Gaussian input signal [5]. The data consists of 1000 pairs of input-output data, of which the first 500 were used for estimation and the remaining 500 for validation. Using the nonlinear autoregressive modelling with exogenous input (NARX) formulation [2], the system can be represented by,

$$y(t) = f^*(\mathbf{x}(t)) + \eta \tag{37}$$

with $\mathbf{x}$ formed from the delayed input and output variables. The VPBF and GRBF models were subjected to this identification task. The parameters used in the example are given in Table 1.

TABLE 1 Design parameter values.

Parameter	VPBF	GRBF
K_0	45	10
L	21	21
$\mathbf{x}$	$\begin{bmatrix} y(t-1) \\ y(t-2) \\ y(t-3) \\ y(t-4) \\ u(t-1) \\ u(t-2) \\ u(t-3) \\ u(t-4) \end{bmatrix}$	$\begin{bmatrix} y(t-1) \\ y(t-2) \\ u(t-1) \\ u(t-2) \end{bmatrix}$
$\varepsilon_1/10^{-3}$	1.5	1.5
ε_2	0.3	0.3
ε_3	7	7

The VPBF Model: With the maximum number of the model terms being 45, there are 2^{45} possible models for selections. But, after 210 generations a sub-optimal model (optimality cannot be guaranteed unless exhaustive search is done) has been found by the algorithm. The best model performance functions are

$$\phi_1(\mathbf{p}) = 1.8 \times 10^{-3}, \quad \phi_2(\mathbf{p}) = 0.3965, \quad \phi_3(\mathbf{p}) = 3. \tag{38}$$

The GRBF Model: Although the maximum number of the model terms is only 10 (*ie.*, 1024 possible models for selection), the search dimension of the basis function centre parameters is 40 in real number space (*ie.*, infinite possibilities for selection). After 700 generations the performance criteria are almost satisfied. In order to obtain better performance, the basis function parameter vector was searched for another 100 generations using the algorithm with the fixed number of the model terms, *ie.*, let $\phi_3(\mathbf{p}) = 5$. The best model performance functions are

$$\phi_1(\mathbf{p}) = 1.3 \times 10^{-3}, \quad \phi_2(\mathbf{p}) = 0.1724, \quad \phi_3(\mathbf{p}) = 5. \tag{39}$$

The convergence of the performance functions with respect to generations are given in Figures 1 and 2.

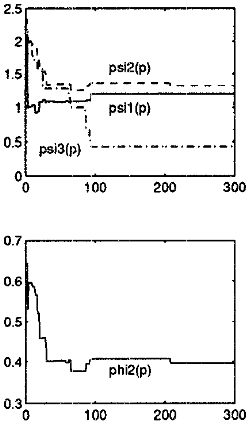

Figure 1. Convergence of performance functions – VPBF.

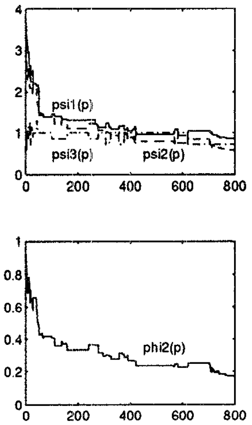

Figure 2. Convergence of performance functions – GRBF.

The measured and estimated outputs, and estimation error of the system on the validation data for the model identified with the VPBF is illustrated in Figure 3 and with the GRBF is shown in Figure 4.

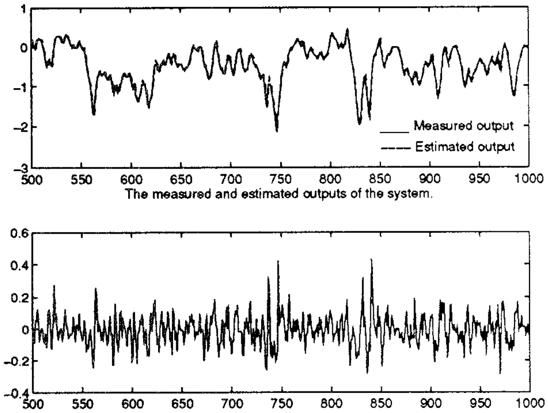

Figure 3. The validation results for VPBF model.

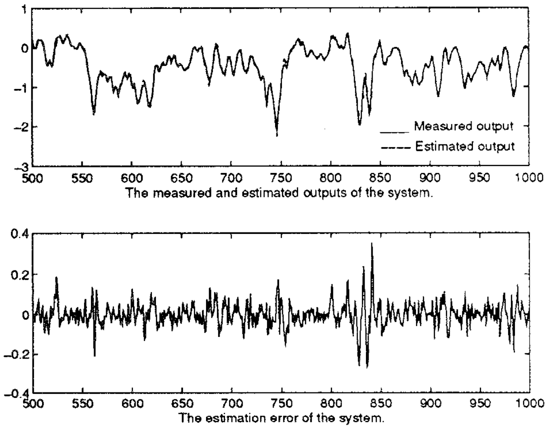

Figure 4. The validation results for GRBF model.

The results show that while the GBRF model is more complex than the VPBF model, it provides a better approximation to the underlying system.

CONCLUSIONS

A learning algorithm with multi-objective criteria for approximation with neural networks has been provided. The set of performance functions chosen measure the approximation accuracy based on L_2- and L_∞ norms and the model complexity based on the number of basis functions in a model. This method incorporates a search for model selection amongst a large number of models formed by the various combinations of basis functions. The optimisation algorithm for model parameter estimation and selection is derived from the method of inequalities and the genetic algorithm. The algorithm is demonstrated on the Volterra polynomial basis function and the Gaussian radial basis function models in a liquild level nonlinear system identification task.

ACKNOWLEDGEMENTS

The authors would like to thank Professor S.A. Billings for his assistance and are grateful for the support of EPSRC of UK under the grant GR/J46661.

REFERENCES

[1] Broomhead, D. S. and Lowe, D. B., 1989, "Multivariable functional interpolation and adaptive networks", Complex Systems, 2, 321-355.

[2] Chen, S. and Billings, S. A., 1989, " Representations of non-linear systems: the NARMAX model", Int. J. Control, 49, 1013-1032.

[3] Cybenko, G., 1989, "Approximations by superpositions of a sigmoidal function", Complex Systems, 5, 603-643.

[4] Davis, L. (ed.), 1991, Handbook of genetic algorithms, Van Nostrand Reinhold, New York.

[5] Fonseca, C. M., Mendes, E. M., Fleming, P. J. and Billings, S. A., 1993, "Nonlinear model term selection with genetic algorithms", Proc. IEE/IEEE Workshop on Natural Algorithms for Signal Processing, 27/1-27/8.

[6] Hajela, P. and Lin, C. Y., 1992, "Genetic search strategies in multicriterion optimal design", Structural Optimization, 4, 99-107.

[7] Kadirkamanathan, V., 1991, Sequential learning in artificial neural networks, PhD Thesis, University of Cambridge, UK.

[8] Kadirkamanathan, V., 1995, "Bayesian inference for basis function selection in nonlinear system identification using genetic algorithms", In Skilling, J. and Sibisi, S., (eds.) Maximum entropy and Bayesian methods, Kluwer.

[9] MacKay, D. J. C., 1992, "Bayesian interpolation", Neural Computation, 4, 415-447.

[10] Powell, M. J. D., 1981, Approximation Theory and Methods, Cambridge University Press, Cambridge.

[11] Powell, M. J. D., 1987, "Radial basis functions for multivariable interpolation: A review", In Mason, J. C. and Cox, M. G., (eds.), Algorithms for Approximation, Oxford University Press, Oxford, 143-167.

[12] Rissanen, J., 1989, Stochastic Complexity in Statistical Inquiry, World Scientific, Singapore.

[13] Schaffer, J. D., and Whitley, D. (eds.), 1992, Combinations of genetic algorithms and neural networks, IEEE Computer Society Press, CA: Los Alamitos.

[14] Schetzen, M., 1980, The Volterra and Wiener theories of nonlinear systems, Wiley, New York.

[15] Whidborne, J. F. and Liu, G. P., 1993, Critical Control Systems: Theory, Design, Applications, Research Studies Press Limited, UK.