

Genetic Evolution of Radial Basis Function Coverage Using Orthogonal Niches

Bruce A. Whitehead

The author is with the University of Tennessee Space Institute, Tullahoma, TN 37388, USA. E-mail:
bwhitehe@utsi.edu

April 25, 1996

Abstract

A well-performing set of radial basis functions (RBF's) can emerge from genetic competition among individual RBF's. Genetic selection of the individual RBF's is based on credit sharing which localizes competition within orthogonal niches. These orthogonal niches are derived using singular value decomposition and are used to apportion credit for the overall performance of the RBF network among individual non-orthogonal RBF's. Niche-based credit apportionment facilitates competition to fill each niche and hence to cover the training data. The resulting genetic algorithm yields RBF networks with better prediction performance on the Mackey-Glass chaotic time series than RBF networks produced by the Orthogonal Least Squares method and by k -means clustering.

I. INTRODUCTION

A so-called “ $1\frac{1}{2}$ - layer” neural network attempts to approximate an unknown function $f : \mathfrak{R}^N \rightarrow \mathfrak{R}$ by a function g drawn from the linear span of a set Φ of basis functions $\phi_i : \mathfrak{R}^N \rightarrow \mathfrak{R}$ for $i = 1, \dots, m$. Given a training set of known values of f , an interesting optimization problem is to find the “best” set of basis functions for approximating these training examples. More precisely, given

- a training set $\{(\mathbf{x}_k, f(\mathbf{x}_k))\}$, $k = 1, \dots, p$ of known values of f ;
- some family Ψ of permissible basis functions;
- some norm $\|\cdot\|$ defined at least on $\Psi \cup \{f\}$; and
- m , the desired number of basis functions to be drawn from Ψ to approximate f ;

then this optimization problem can be stated as follows:

Find the subset (of size m) $\Phi \subset \Psi$ which minimizes $\|f - g_\Phi\|$, where g_Φ is the function closest to f in the linear span of Φ

in which “closest” is defined in terms of this same norm $\|\cdot\|$. In general, the norm above might be chosen to reflect regularization criteria [1], [2]. We consider only the commonly-used rms error criterion, however, and accordingly use a Euclidean norm over the finite basis $\{\delta_k\}$, $k = 1, \dots, p$, where $\delta_k(\mathbf{x})$ is 1 at $\mathbf{x} = \mathbf{x}_k$ and 0 elsewhere. In this basis, $\|f - g_\Phi\|$ is simply the rms training error, and minimizing it is equivalent to maximizing $\|g_\Phi\|$, the length of the projection of f onto the space spanned by Φ . For convenience, since the norm is Euclidean, we will take $\|g_\Phi\|^2$ as the objective function to be maximized in our optimization problem.

If Ψ is small enough to actually evaluate all possible subsets Φ of size m , then the optimization problem becomes trivial. More typically, however, Ψ is taken to be a space such as the space of all possible affine transformations of a given prototype basis function. In such cases, optimization methods often involve procedures to (i) *generate* candidate sets Φ , and (ii) *evaluate* each candidate set Φ by computing $\|g_\Phi\|^2$ or its equivalent. A population-based optimization method, such as a conventional genetic algorithm (GA) [3], typically generates a population P consisting of many candidate sets and evaluates each candidate set Φ in this population. Such a population defines one generation of the genetic algorithm. The candidates belonging to this generation are typically evaluated, selected, perturbed, and recombined to yield a new population of candidates for the next generation.

In such a conventional GA method, each candidate set Φ in the population P would represent a separate, complete solution to the original problem (the problem of approximating f). Each candidate set Φ would try to solve this problem by itself, and would do better or worse according to how close its linear span could get to f . Note that each candidate set Φ would be evaluated as a whole. The evaluation of Φ would depend on its constituent basis functions, but these individual basis functions would never receive individual evaluations. The only evaluation would be of sets of basis functions.

We propose a different genetic approach in which each individual basis function ϕ_i receives an individual evaluation. The basis functions ϕ_i are individually evaluated, selected, perturbed, and recombined. Our hope is that a population P of these individually evolved basis functions will emerge to cooperatively cover the domain of f to be approximated. In other words, we are trying to engineer an evolutionary process at the level of individual basis functions, hoping that a set of basis functions which work well together will emerge at the population level.

Since our genetic population P contains just one set Φ of basis functions, only one neural network is evaluated in a given generation of the GA. This could potentially yield an evolutionary process which is computationally faster than evaluating many competing sets of basis functions in each generation. But with only one set Φ per generation, we have only one number, $\|g_\Phi\|^2$, to guide the GA in producing the next generation. To make

the GA work, we need to apportion this single number $\|g_\Phi\|^2$ into a separate evaluation of each basis function ϕ_i in Φ . Such an evaluation must capture the value of the contribution of that particular basis function to the overall performance of the set of basis functions.

II. CREDIT APPORTIONMENT PROBLEM

If the basis functions ϕ_i were mutually orthogonal, then an obvious way to apportion credit would be to assign each basis function ϕ_i a credit of $(\phi_i \cdot f)^2 / \|\phi_i\|^2$ where \cdot denotes inner product in the finite Euclidean basis $\{\delta_k\}$, $k = 1, \dots, p$ given above. If the functions f and ϕ_i were each linearly rescaled to have a mean of 0 over the set of training examples, then in the terminology of linear regression, each basis function would receive credit for the proportion of the variance in f it could account for.

It is not as clear, however, how to apportion credit among a set of *non-orthogonal* basis functions. One possible approach is to order the non-orthogonal basis functions in some well-defined sequence ϕ_1, \dots, ϕ_m , and to apply “winner-take-all” credit assignment at each step in the sequence, as in the following orthogonal least squares (OLS) algorithm [4]:

At each step $k + 1$ in the sequence, we have already chosen the first k basis functions in the sequence and have already defined an orthonormal basis $U^{(k)}$ which spans these k basis functions. Position $k + 1$ in the sequence is then filled by holding a competition among the remaining $m - k$ basis functions. Each of these $m - k$ basis functions is projected onto the subspace orthogonal to $U^{(k)}$, and each such projection is normalized to unit length. The squared inner product of each such normalized projection with f is computed. The projected basis function with the highest squared inner product (yielding the maximum incremental reduction in rms training error) wins the competition and is chosen to fill position $k + 1$ in the sequence. Its projection orthogonal to $U^{(k)}$ then provides the additional orthogonal basis vector to form $U^{(k+1)}$.

Given a finite set $\overline{\Phi} \subset \Psi$ of basis functions sufficient but larger than necessary to approximate f to the desired accuracy (e.g., radial basis functions centered on each training example), OLS rank ordering of $\overline{\Phi}$ selects the subset $\Phi \subset \overline{\Phi}$ of size m which maximizes $\|g_\Phi\|^2$. [4]

Intuitively, if some portion of the variance in f can be accounted for by either ϕ_i or ϕ_j , then whichever basis function wins earlier in the OLS sequence receives all the credit for accounting for this portion of the variance. Some other neural network models [5]–[7] with other ways of evolving new basis functions in a sequence also employ winner-take-all credit

assignment at each step in the sequence.

In the remainder of this paper, we propose an alternative to this sequential winner-take-all approach. Our objective is to apportion credit among m non-orthogonal basis functions such that the credit assigned to any given basis function is *independent of any ordering* among the basis functions. Intuitively, if some portion of the variance in f can be accounted for by either ϕ_i or ϕ_j , then credit for this portion of the variance must be shared between the two basis functions in a manner which does not depend on one preceding or following the other in some rank ordering. Our approach might therefore be termed an “unordered credit-sharing” approach.

Section III develops the proposed credit-sharing method. Section IV experimentally compares its generalization performance with that of two alternatives: “winner-take-all” credit apportionment using the OLS algorithm of [4], and k -means clustering [8].

III. CREDIT SHARING METHOD

Since the basis functions ϕ_i in Φ are not constrained to be orthogonal, our strategy will be to apportion credit for $\|g_\Phi\|^2$ into orthogonal components and then to reassemble these orthogonal components into the non-orthogonal basis functions ϕ_i .

g_Φ can be computed by singular value decomposition (SVD) of the matrix A whose elements are $a_{ki} = \phi_i(\mathbf{x}_k)$. Let us express this SVD as $A = U\Sigma V^T$ where the columns $\mathbf{u}_1, \dots, \mathbf{u}_n$ of the matrix U form an orthonormal basis for the space spanned by Φ ; where Σ is a diagonal matrix of n non-zero singular values $\sigma_1, \dots, \sigma_n$; and where the orthonormal matrix V has entries v_{ij} . For convenience, define $f_j = \mathbf{u}_j \cdot f$ where \cdot denotes inner product in the finite Euclidean basis $\{\delta_k\}$, $k = 1, \dots, p$ given above.

The best least-squares approximation to f in the linear span of Φ , using SVD, is the weighted sum $g_\Phi = \sum_i w_i \phi_i$ where each weight is $w_i = \sum_{j'} v_{ij'} f_{j'} / \sigma_{j'}$ and where each basis function ϕ_i can be expressed in the SVD’s orthonormal basis as $\phi_i = \sum_j \sigma_j v_{ij} \mathbf{u}_j$. Let us represent each weighted basis function by defining $\hat{\phi}_i = w_i \phi_i = w_i \sum_j \sigma_j v_{ij} \mathbf{u}_j = \sum_j w_i \sigma_j v_{ij} \mathbf{u}_j = \sum_j \hat{\phi}_{ij}$ where this last step expresses each weighted basis function $\hat{\phi}_i$ as the sum of orthogonal components $\hat{\phi}_{ij}$, $j = 1, \dots, n$ defined by

$$\hat{\phi}_{ij} = w_i \sigma_j v_{ij} \mathbf{u}_j = \left(\sum_{j'} \frac{1}{\sigma_{j'}} v_{ij'} f_{j'} \right) \sigma_j v_{ij} \mathbf{u}_j. \quad (1)$$

Our original SVD fit can now be expressed as $g_{\Phi} = \sum_i \sum_j \hat{\phi}_{ij}$. This is equivalent to $g_{\Phi} = \sum_j g_j$, where for each j we define $g_j = \sum_i \hat{\phi}_{ij}$, in order to collect together those components $\hat{\phi}_{ij}$ that are collinear with \mathbf{u}_j . This definition of g_j reduces to $g_j = f_j \mathbf{u}_j$ using equation 1 and the property that columns j and j' of the orthonormal matrix V have an inner product of 1 if $j = j'$ or 0 otherwise.

The least-squares fit g_{Φ} is thus the sum of orthogonal components g_j , $j = 1, \dots, n$, and each g_j is in turn the sum of collinear pieces $\hat{\phi}_{ij}$, $i = 1, \dots, m$. The total credit to be apportioned, $\|g_{\Phi}\|^2$, can therefore be decomposed as $\|g_{\Phi}\|^2 = \sum_j \|g_j\|^2 = \sum_j f_j^2$. The total credit available in each orthogonal dimension \mathbf{u}_j is f_j^2 , the squared magnitude of the projection of f onto that dimension. This credit should be apportioned among those pieces $\hat{\phi}_{ij}$, $i = 1, \dots, m$ which are collinear with \mathbf{u}_j . This is because these are the pieces of the original basis functions ϕ_i which are responsible for the approximation g_j to f along the \mathbf{u}_j dimension, and which in fact sum to g_j . We can apportion the total credit f_j^2 available in dimension \mathbf{u}_j among these collinear pieces in proportion to their coefficients in equation 1 by setting

$$\text{credit}(\hat{\phi}_{ij}) = \hat{\phi}_{ij} \cdot f = \left(\sum_{j'} \frac{1}{\sigma_{j'}} v_{ij'} f_{j'} \right) \sigma_j v_{ij} f_j \quad (2)$$

which forces the credit given to the pieces $\hat{\phi}_{ij}$ in dimension \mathbf{u}_j to sum to $\sum_i \hat{\phi}_{ij} \cdot f = g_j \cdot f = (f_j \mathbf{u}_j) \cdot f = f_j^2$ as desired. Reassembling these pieces $\hat{\phi}_{ij}$ into the original non-orthogonal basis functions ϕ_i , we see that the credit assigned to each ϕ_i should be

$$\text{credit}(\phi_i) = \sum_j \text{credit}(\hat{\phi}_{ij}) = \sum_{j'} \frac{1}{\sigma_{j'}} v_{ij'} f_{j'} \sum_j \sigma_j v_{ij} f_j. \quad (3)$$

This defines how credit for $\|g_{\Phi}\|^2$ is shared among non-orthogonal basis functions. As desired, the credit assigned to any given basis function is independent of any ordering among the basis functions. The orthonormal basis $\{\mathbf{u}_j\}$ is only an intermediate step in apportioning credit among these non-orthogonal basis functions.

Intuitive expectations for the behavior of this credit apportionment method can most easily be explained in terms of the concept of niche sharing in genetic algorithms [9]–[11]. In this terminology, each basis vector \mathbf{u}_j represents an orthogonal niche. Instead of the global competition of “winner-take-all” credit apportionment, we now have a separate local

competition within each niche. To recapitulate the development above in these terms,

1. The projection of f onto the space spanned by Φ has been expressed as the sum of non-orthogonal weighted basis functions $\hat{\phi}_i$.
2. Each $\hat{\phi}_i$ has been decomposed into components $\hat{\phi}_{ij}$ belonging to different niches \mathbf{u}_j .
3. Equation 2 means that components arising from different basis functions which belong to the *same* orthogonal niche \mathbf{u}_j will compete for the fixed amount of credit f_j^2 available in that niche.
4. Components arising from different basis functions which belong to *different* orthogonal niches \mathbf{u}_j and $\mathbf{u}_{j'}$ will *not* be competing with each other, since they are competing for different pools of credit belonging to different orthogonal niches. Different niches do not compete with each other since the total credit available in each niche \mathbf{u}_j is fixed (by the projection of f onto that niche), and accordingly not subject to competition from other orthogonal niches.

This last point is the reason we do not expect competition in the GA to force most members of the population Φ to converge toward the same optimum (as in a typical GA), which would produce fit individuals, but not a co-adapted population. Instead, proportional selection using equation 3 will reward the spread of basis functions into less-filled niches while providing less reward (and less chance of reproduction) for basis functions which crowd into an already-filled niche to share the limited credit available in that niche [3], [9], [10]. By promoting competition within each niche, but not between niches, we expect a GA using equation 3 to spread its basis functions over different niches in accordance with the credit available in each niche. *This is the purpose of the credit sharing given by equation 3.* We expect the overall result to be a good fit to f resulting from a good fit within each niche resulting from competition within that niche.

IV. SIMULATION RESULTS

A genetic algorithm (GA) with proportional selection governed by equation 3 was simulated to determine whether selection at the level of individual basis functions, based on credit sharing, could produce a co-adapted population—a population of local basis functions cooperating to cover the domain of f to yield a good approximation. Each genetic string in the population consists of a binary encoding of the vector center and scalar

width of one Gaussian RBF ϕ_i . The initial population of m genetic strings is generated randomly. Each successive generation of the GA replaces $\frac{1}{4}$ of the population using selection probabilities proportional to the credit assigned by equation 3. To preclude nearly singular combinations of RBF's (which would not be expected to generalize well) each σ in equation 3 is mapped to the nearest value within the range $[\.033, 30]$, to maintain a ratio of less than 1000 between largest and smallest singular values. The selected genetic strings are perturbed by genetic recombination, mutation, and creep operators the same as described in previous work studying non-orthogonal niches [12].

RBF placements evolved by this genetic algorithm were compared with those produced by the OLS algorithm [4] and by k -means clustering [8], using the neural network benchmark task of predicting the Mackey-Glass chaotic time series. The time series equation, parameters, data sets, procedures followed, k -means algorithms, and RBF width determination were as described in [12]–[13], following previous studies [8], [14]–[16]. (The data and the RBF placements produced by all three methods are available by anonymous ftp in `microlab.utsi.edu:/pub/whitehead/ortho_niche_data`). Each GA run consisted of 2000 generations, requiring approximately 2.5 hours of CPU time on a 36-MHz microsparc processor for a population size of 100. Since only one RBF network is evaluated in each generation, this computation is comparable to just a few generations of a conventional GA evaluating many RBF networks per generation.

The RBF placements produced by each of the three methods were fit to the training data using least-squares SVD. Each resulting linear combination of RBF's was then applied to the test data to predict values of the Mackey-Glass time series subsequent to the training data, as in [13]. Figure 1 shows the normalized prediction error obtained for each method for populations ranging from 25 to 150 RBF units. Each point plotted for the GA averages four GA runs using different random seeds. Since the sampling error of the GA's proportional selection would decrease as the population size increases, it is not surprising that the GA's advantage appears more pronounced for larger populations.

These initial results appear to confirm that high performance at the RBF population level can emerge from genetic evolution at the individual RBF level, based on credit sharing along orthogonal dimensions. Since the credit sharing derived in Section III is not limited

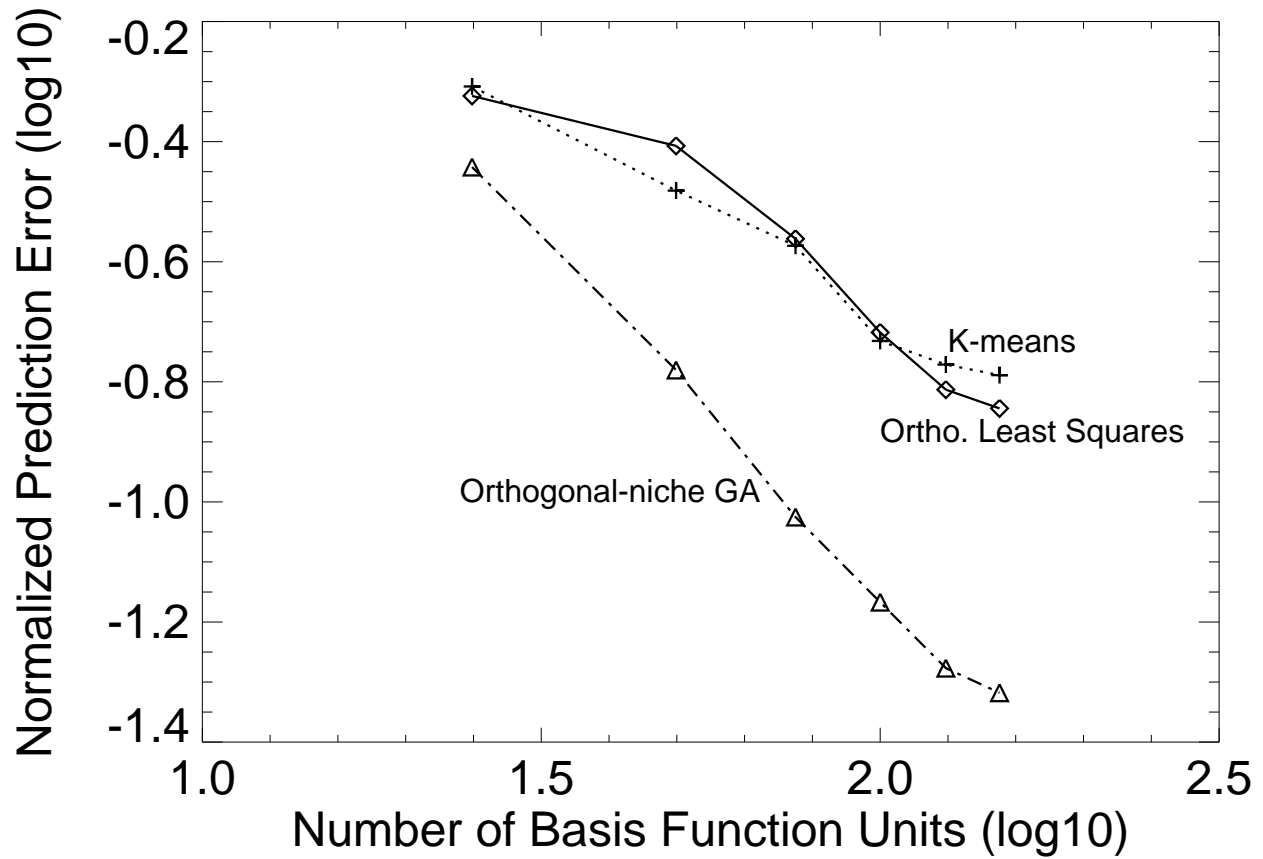


Fig. 1. Generalization performance of Gaussian RBF networks evolved by the orthogonal-niche genetic algorithm ($-\cdot-\Delta-\cdot-$), compared with those produced by the Orthogonal Least Squares [4] method ($-\diamond-$) and by k -means clustering ($\cdots+\cdots$). The performance of each method, using 25, 50, 75, 100, 125, and 150 RBF's, is measured as the normalized prediction error on test data later than the training data in the Mackey-Glass time series.

to Gaussian RBF's, its behavior using other basis functions merits further investigation.

V. ACKNOWLEDGMENTS

The simulation software derived from GA software by Tim Choate. Singular value methods used LAPACK. The contrast with OLS was pointed out by an anonymous reviewer.

REFERENCES

- [1] C. Bishop, "Improving the generalization properties of radial basis function neural networks," *Neural Computation*, vol. 3, no. 4, pp. 579–588, 1991.
- [2] F. Girosi, "Some extensions of radial basis functions and their applications in artificial intelligence," *Computers Math. Applic.*, vol. 24, no. 12, pp. 61–80, 1992.
- [3] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: The University of Michigan Press, 1975.
- [4] 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, no. 2, pp. 302–309, 1991.
- [5] S. E. Fahlman and C. Lebiere, "The cascade-correlation learning architecture," in *Advances in Neural Information Processing Systems 2* (R. P. Lippmann, J. E. Moody, and D. S. Touretzky, eds.), pp. 524–532, San Francisco: Morgan Kaufmann, 1991.
- [6] N. Karunanithi, R. Das, and D. Whitley, "Genetic cascade learning for neural networks," in *Combinations of Genetic Algorithms and Neural Networks* (L. D. Whitley and J. D. Schaffer, eds.), pp. 134–145, Los Alamitos, CA: IEEE Computer Society Press, June 1992.
- [7] M. A. Potter, "A genetic cascade-correlation learning algorithm," in *Combinations of Genetic Algorithms and Neural Networks* (L. D. Whitley and J. D. Schaffer, eds.), pp. 123–133, Los Alamitos, CA: IEEE Computer Society Press, June 1992.
- [8] J. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computation*, vol. 1, pp. 281–294, 1989.
- [9] K. Deb and D. E. Goldberg, "An investigation of niche and species formation in genetic function optimization," in *Proceedings of the Third International Conference on Genetic Algorithms*, pp. 42–50, San Francisco: Morgan Kaufmann, June 1989.
- [10] M. E. Palmer and S. J. Smith, "Improved evolutionary optimization of difficult landscapes: Control of premature convergence through scheduled sharing," *Complex Systems*, vol. 5, no. 5, pp. 443–458, 1991.
- [11] J. Horn, D. E. Goldberg, and K. Deb, "Implicit niching in a learning classifier system: Nature's way," *Evolutionary Computation*, vol. 2, no. 1, pp. 37–66, 1994.
- [12] B. A. Whitehead and T. D. Choate, "Cooperative - competitive genetic evolution of radial basis function centers and widths for time series prediction," *IEEE Transactions on Neural Networks*, in press, 1996. (Anonymous ftp from `microlab.utsi.edu:/pub/whitehead/papers/non_ortho_niche.ps`).
- [13] B. A. Whitehead and T. D. Choate, "Evolving space-filling curves to distribute radial basis functions over an input space," *IEEE Transactions on Neural Networks*, vol. 5, no. 1, pp. 15–23, 1994.
- [14] J. D. Farmer and J. J. Sidorowich, "Predicting chaotic time series," *Physical Review Letters*, vol. 59, no. 8, pp. 845–848, 1987.

- [15] M. F. Tenorio, "Self-organizing network for optimum supervised learning," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 100–110, 1990.
- [16] J. Platt, "A resource-allocating network for function interpolation," *Neural Computation*, vol. 3, pp. 213–225, 1991.