On the Construction and Training of Reformulated Radial Basis Function Neural Networks

Nicolaos B. Karayiannis, *Senior Member, IEEE,* and Mary M. Randolph-Gips

***Abstract*—This paper presents a systematic approach for constructing reformulated radial basis function (RBF) neural networks, which was developed to facilitate their training by supervised learning algorithms based on gradient descent. This approach reduces the construction of radial basis function models to the selection of admissible generator functions. The selection of generator functions relies on the concept of the blind spot, which is introduced in this paper. This paper also introduces a new family of reformulated radial basis function neural networks, which are referred to as cosine radial basis functions. Cosine radial basis functions are constructed by linear generator functions of a special form and their use as similarity measures in radial basis function models is justified by their geometric interpretation. A set of experiments on a variety of datasets indicate that cosine radial basis functions outperform considerably conventional radial basis function neural networks with Gaussian radial basis functions. Cosine radial basis functions are also strong competitors to existing reformulated radial basis function models trained by gradient descent and feedforward neural networks with sigmoid hidden units.**

***Index Terms*—Absolute sensitivity, active region, blind spot, cosine radial basis function, generator function, gradient descent learning, radial basis function (RBF) neural network, reformulation.**

I. Introduction

RADIAL basis function (RBF) neural networks are function approximation models that can be trained by examples to implement a desired input–output mapping [1], [6]. In fact, radial basis function models are closely related to function approximation models used to perform interpolation [16]. Under certain mild conditions on the radial basis functions, the RBF neural networks are capable of approximating arbitrarily well any function [5]. The performance of an radial basis function neural network depends on the number and centers of the radial basis functions, their shapes, and the method used for learning the input–output mapping. Broomhead and Lowe [2] suggested that the centers of the radial basis functions can either be distributed uniformly within the region of the input space for which there is data, or chosen to be a subset of the training vectors by analogy with strict interpolation. Moody and Darken [17] proposed a hybrid learning process for training radial basis function neural networks with Gaussian radial basis functions, which employs a supervised scheme for updating the output weights, i.e., the weights that connect the radial basis functions with the output units, and an unsupervised clustering algorithm for determining the centers of the radial basis functions. The centers of the radial basis functions are often determined by the k-means (or c-means) clustering algorithm [15]. Poggio and Girosi [18] proposed a supervised approach for training radial basis function neural networks with Gaussian radial basis functions, which updates the radial basis function centers together with the output weights. Chen *et al.* [4], proposed a learning procedure for radial basis function neural networks based on the orthogonal least squares (OLS) method, which is used as a forward regression procedure to select a suitable set of radial basis function centers. Cha and Kassam [3] proposed a stochastic gradient training algorithm for radial basis function neural networks with Gaussian radial basis functions, which uses gradient descent to update all their free parameters (radial basis function centers, widths of the Gaussian radial basis functions, and output weights). Whitehead and Choate [19] proposed an evolutionary training algorithm for radial basis function neural networks. In this approach, the centers of the radial basis functions are governed by space-filling curves whose parameters evolve genetically.

The training of radial basis function neural networks using gradient descent offers a solution to the tradeoff between performance and training speed and can make radial basis function neural networks serious competitors to feedforward neural networks (FFNNs) with sigmoid hidden units [8]–[10], [12]–[14]. The convergence of gradient descent learning and the performance of the trained radial basis function neural networks are both affected rather strongly by the choice of radial basis functions. The search for admissible radial basis functions other than the Gaussian function motivated the development of an axiomatic approach for constructing reformulated radial basis function neural networks suitable for gradient descent learning [8]–[10], [12]–[14]. This approach reduces the development of reformulated radial basis function models to the selection of admissible generator functions that determine the form of the radial basis functions.

This paper presents new results on the construction and training of reformulated radial basis function neural networks. The results of the analysis presented in this paper can be used for selecting generator functions according to their suitability for gradient descent learning. This paper also introduces a new family of reformulated radial basis function neural networks, which are constructed by linear generator functions of a special form. These radial basis functions are referred to as cosine radial basis functions because of an interesting geometric interpretation of their responses. Cosine radial basis functions are trained by gradient descent to perform a variety of classification

Manuscript received April 2, 2002; revised October 21, 2002.

N. B. Karayiannis is with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77204-4005 USA.

M. M. Randolph-Gips is with the Department of Computer Engineering, University of Houston-Clear Lake, Houston, TX 77058 USA.

Digital Object Identifier 10.1109/TNN.2003.813841

tasks and their performance is compared with that of alternative radial basis function models and conventional FFNNs with sigmoid hidden units.

II. Reformulated Radial Basis Function Neural Networks

Consider the $\mathbb{R}^n \rightarrow \mathbb{R}^p$ mapping implemented by the model

$$\hat{y}_i = f\left(w_{i0} + \sum_{j=1}^{c} w_{ij} g_j\left(\|\mathbf{x}-\mathbf{v}_j\|^2\right)\right), 1 \leq i \leq p \quad (1)$$

where $f(\cdot)$ is a nondecreasing, continuous and differentiable function. The model (1) describes an radial basis function neural network with inputs from $\mathbb{R}^n$, c radial basis functions, and p output units if $g_j(x^2) = \phi_j(x)$, and $\phi_j(x)$ are radial basis functions. In such a case, the response of the radial basis function neural network to the input vector $\mathbf{x}_k$ is $\hat{y}_{i,k} = f(\sum_{j=0}^{c} w_{ij} h_{j,k})$, $1 \leq i \leq p$, where $h_{0,k} = 1, \forall k$, and $h_{j,k}$ represents the response of the radial basis function located at the jth prototype $\mathbf{v}_j$ to the input vector $\mathbf{x}_k$, that is, $h_{j,k} = g_j(\|\mathbf{x}_k - \mathbf{v}_j\|^2)$, $1 \leq j \leq c$.

A. Axiomatic Requirements

Reformulated radial basis function neural networks were developed to facilitate the training of radial basis function models by learning algorithms based on gradient descent [8]–[10], [12]–[14]. This was attempted by including the centers of the radial basis functions in the adjustable model parameters and searching for radial basis functions that improve the effectiveness of gradient descent learning. The construction of reformulated radial basis function neural networks relied on a set of intuitively appealing axioms, which guarantee that the resulting radial basis functions have some desirable properties [8]–[10]. The main motivation behind these axioms was the preservation of the localized nature of radial basis function models. This can be accomplished by constructing radial basis function models containing locally tuned radial basis functions, that is, radial basis functions sensitive to input vectors from certain regions of the input space.

In order for the model (1) to satisfy the desired properties mentioned above, any admissible radial basis function $\phi_j(x) = g_j(x^2)$ must satisfy the following three basic axiomatic requirements [8]–[10]:

Axiom 1: $g_j(\|\mathbf{x}-\mathbf{v}\|^2) > 0$ for all $\mathbf{x}, \mathbf{v} \in \mathbb{R}^n$.

Axiom 2: $g_j(\|\mathbf{x}-\mathbf{v}\|^2) > g_j(\|\mathbf{y}-\mathbf{v}\|^2)$ for all $\mathbf{x}, \mathbf{y}, \mathbf{v} \in \mathbb{R}^n$ such that $\|\mathbf{x}-\mathbf{v}\|^2 < \|\mathbf{y}-\mathbf{v}\|^2$.

Axiom 3: If $\nabla_{\mathbf{x}} g_j \doteq \nabla_{\mathbf{x}} g_j(\|\mathbf{x}-\mathbf{v}\|^2)$ denotes the gradient with respect to $\mathbf{x}$ of $g_j(\|\mathbf{x}-\mathbf{v}\|^2)$ at $\mathbf{x}$, then

$$\frac{\|\nabla_{\mathbf{x}} g_j\|^2}{\|\mathbf{x}-\mathbf{v}\|^2} > \frac{\|\nabla_{\mathbf{y}} g_j\|^2}{\|\mathbf{y}-\mathbf{v}\|^2} \quad (2)$$

for all $\mathbf{x}, \mathbf{y}, \mathbf{v} \in \mathbb{R}^n$ such that $\|\mathbf{x}-\mathbf{v}\|^2 < \|\mathbf{y}-\mathbf{v}\|^2$. Consider the model described by (1) and let $\mathcal{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c\}$ be the set of prototypes that determine the centers of the radial basis functions. Axiom 1 requires that the response $h_j = g_j(\|\mathbf{x}-\mathbf{v}_j\|^2)$ of an admissible radial basis function to any $\mathbf{x} \in \mathbb{R}^n$ be always positive. The second axiom guarantees that the response of an admissible radial basis function be localized by requiring that $h_j = g_j(\|\mathbf{x}-\mathbf{v}_j\|^2)$ be a monotonically decreasing function of $\|\mathbf{x}-\mathbf{v}_j\|^2$ for any $\mathbf{x} \in \mathbb{R}^n$. Axiom 3 requires that $\|\nabla_{\mathbf{x}} h_j\|^2 / \|\mathbf{x}-\mathbf{v}_j\|^2$ be a monotonically decreasing function of $\|\mathbf{x}-\mathbf{v}_j\|^2$. In a sense, the third axiom reinforces the localized nature of radial basis functions during the learning process by guaranteeing that there exists a region around any prototype $\mathbf{v}_j$ where the norm $\|\nabla_{\mathbf{x}} h_j\|^2$ of the gradient $\nabla_{\mathbf{x}} h_j$ diminishes as $\|\mathbf{x}-\mathbf{v}_j\|^2$ increases.

The three basic axiomatic requirements impose some rather mild mathematical restrictions on the search for admissible radial basis functions that can lead to reformulated radial basis function neural-network models [10]. In addition to these axiomatic requirements, the search for radial basis functions can be restricted even further by imposing other desirable conditions. For example, it is reasonable to require that the responses of all radial basis functions to all inputs be bounded. This suggests the following complementary axiomatic requirement for radial basis functions:

Axiom 4: $g_j(\|\mathbf{x}-\mathbf{v}\|^2) < \infty$ for all $\mathbf{x}, \mathbf{v} \in \mathbb{R}^n$.

B. Admissibility Conditions for Radial Basis Functions

The selection of admissible radial basis functions can be facilitated by extending the theorem proposed in [10].

Theorem 1: The model described by (1) represents an radial basis function neural network in accordance with all four axiomatic requirements if $g_j(x)$ are continuous functions on $(0, \infty)$ with continuous first-order derivatives $g_j'(x)$ such that:

1) $g_j(x) > 0, \forall x \in (0, \infty)$;
2) $g_j'(x) < 0, \forall x \in (0, \infty)$;
3) $g_j''(x) > 0, \forall x \in (0, \infty)$;
4) $\lim_{x \rightarrow 0_+} g_j(x) = L_j$, where L_j are finite numbers.

Proof: The proof of conditions 1–3 can be found in [10]. The proof of condition 4 is based on the fourth axiom, which requires that $g_j(\|\mathbf{x}-\mathbf{v}\|^2) < \infty$ for all $\mathbf{x}, \mathbf{v} \in \mathbb{R}^n$. This condition is satisfied if $g_j(x)$ is bounded for all $x \in (0, \infty)$. Since $g_j(x)$ is a monotonically decreasing function of $x \in (0, \infty)$, $g_j(x)$ is bounded for all $x \in (0, \infty)$ if $\lim_{x \rightarrow 0_+} g_j(x) = L_j < \infty$.

A radial basis function is said to be *admissible in the wide sense* if it satisfies the three basic axiomatic requirements, that is, the first three conditions of Theorem 1 [8]–[10]. If a radial basis function satisfies all four axiomatic requirements, that is, all four conditions of Theorem 1, then it is said to be *admissible in the strict sense*.

Theorem 1 verifies the strong link between radial basis function neural networks and function approximation models used to perform interpolation. Such function approximation models attempt to determine a surface in a Euclidean space $\mathbb{R}^n$ that provides the best fit for the data $(\mathbf{x}_k, y_k)$, $1 \leq k \leq M$, where

$\mathbf{x}_k \in \mathcal{X} \subset \mathbb{R}^n$ and $y_k \in \mathbb{R}$ for all $k = 1, 2, \ldots, M$. Micchelli [16] considered the solution of the interpolation problem $s(\mathbf{x}_k) = y_k$, $1 \leq k \leq M$, by functions $s : \mathbb{R}^n \rightarrow \mathbb{R}$ of the form

$$s(\mathbf{x}) = \sum_{k=1}^{M} w_k g\left(\|\mathbf{x} - \mathbf{x}_k\|^2\right). \tag{3}$$

This formulation treats interpolation as a function approximation problem, with the function $s(\cdot)$ generated by the fitting procedure as the best approximation to this mapping. Given the form of the basis function $g(x)$, the function approximation problem described by $s(\mathbf{x}_k) = y_k$, $1 \leq k \leq M$, reduces to determining the weights w_k, $1 \leq k \leq M$, associated with the model (3). Micchelli [16] showed that the model described by eqn (3) is admissible for interpolation if the basis function $g(x)$ is *completely monotonic* on $(0, \infty)$. A function $g(x)$ is called completely monotonic on $(0, \infty)$ if it is continuous on $(0, \infty)$ and its ℓth order derivatives $g^{(\ell)}(x)$ satisfy $(-1)^\ell g^{(\ell)}(x) \geq 0$, $\forall x \in (0, \infty)$, for $\ell = 0, 1, 2, \ldots$.

Theorem 1 requires that any wide-sense admissible function $g_j(x)$ be continuous on $(0, \infty)$ and its derivatives satisfy $(-1)^\ell g_j^{(\ell)}(x) \geq 0$, $\forall x \in (0, \infty)$, for $\ell = 0, 1, 2$. Theorem 1 is less restrictive than Micchelli's interpolation theorem in terms of the conditions imposed on the selection of functions $g_j(x)$ that are admissible in the wide sense. However, Theorem 1 is more restrictive than Micchelli's interpolation theorem if it is used to select functions $g_j(x)$ that are admissible in the strict sense.

C. Admissibility Conditions for Generator Functions

The search for admissible radial basis functions can be simplified by considering basis functions of the form $\phi_j(x) = g_j(x^2)$, with each $g_j(x)$ defined in terms of a *generator function* $g_{j0}(x)$ as $g_j(x) = (g_{j0}(x))^{1/(1-m)}$, $m \neq 1$ [8]–[10]. The selection of generator functions that lead to admissible radial basis functions can be facilitated by extending the theorem proposed in [10]:

Theorem 2: Consider the model (1) and let each $g_j(x)$ be defined as $g_j(x) = (g_{j0}(x))^{1/(1-m)}$, $m \neq 1$, where $g_{j0}(x)$ is a generator function that is continuous on $(0, \infty)$ and has continuous first- and second-order derivatives. If $m > 1$, then this model represents an radial basis function neural network in accordance with all four axiomatic requirements if:

1) $g_{j0}(x) > 0$, $\forall x \in (0, \infty)$;
2) $g'_{j0}(x) > 0$, $\forall x \in (0, \infty)$;
3) $r_{j0}(x) = [m/m-1](g'_{j0}(x))^2 - g_{j0}(x)g''_{j0}(x) > 0$, $\forall x \in (0, \infty)$;
4) $\lim_{x \rightarrow 0_+} g_{j0}(x) = L_{1j}$, where $L_{1j} \in (0, \infty)$.

If $m < 1$, then this model represents an radial basis function neural network in accordance with all four axiomatic requirements if:

1) $g_{j0}(x) > 0$, $\forall x \in (0, \infty)$;
2) $g'_{j0}(x) < 0$, $\forall x \in (0, \infty)$;
3) $r_{j0}(x) = [m/m-1](g'_{j0}(x))^2 - g_{j0}(x)g''_{j0}(x) < 0$, $\forall x \in (0, \infty)$;
4) $\lim_{x \rightarrow 0_+} g_{j0}(x) = L_{2j}$, where $L_{2j} \in (0, \infty)$.

Proof: The proof of conditions 1–3 can be found in [10]. The proof of condition 4 is outlined below. If $m > 1$ $(1 - m < 0)$, the condition $\lim_{x \rightarrow 0_+} g_j(x) = \lim_{x \rightarrow 0_+} (g_{j0}(x))^{1/(1-m)} = L_j < \infty$ of Theorem 1 is satisfied if $\lim_{x \rightarrow 0_+} g_{j0}(x) \neq 0$. Since $g_{j0}(x) > 0$, $x \in (0, \infty)$, the condition $\lim_{x \rightarrow 0_+} g_{j0}(x) \neq 0$ implies that $\lim_{x \rightarrow 0_+} g_{j0}(x) = L_{1j} > 0$. If $m < 1$ $(1 - m > 0)$, the condition $\lim_{x \rightarrow 0_+} g_j(x) = \lim_{x \rightarrow 0_+} (g_{j0}(x))^{1/1-m} = L_j < \infty$ of Theorem 1 is satisfied if there exists a positive number L_{2j} so that $\lim_{x \rightarrow 0_+} g_{j0}(x) = L_{2j} < \infty$.

Any generator function that satisfies the first three conditions of Theorem 2 leads to admissible radial basis functions in the wide sense [8]–[10]. Admissible radial basis functions in the strict sense can be obtained from generator functions that satisfy all four conditions of Theorem 2.

D. Constructing Admissible Generator Functions

Theorem 2 essentially reduces the construction of admissible radial basis function models to the search for admissible generator functions. A broad variety of admissible generator functions can be determined by a constructive approach based on the admissibility conditions of Theorem 2. The construction of wide-sense admissible generator functions can be attempted by assuming that $g'_{j0}(x) = p_j(g_{j0}(x))$, where $p_j(\cdot)$ satisfies certain conditions in accordance with Theorem 2. Theorem 2 requires that $g'_{j0}(x) > 0$, $\forall x \in (0, \infty)$, for $m > 1$ and $g'_{j0}(x) < 0$, $\forall x \in (0, \infty)$, for $m < 1$. Since it is also required by Theorem 2 that $g_{j0}(x) > 0$, $\forall x \in (0, \infty)$, the function $p_j(\cdot)$ must be selected so that $p_j(x) > 0$, $\forall x \in (0, \infty)$, if $m > 1$ and $p_j(x) < 0$, $\forall x \in (0, \infty)$, if $m < 1$. For such functions, the admissibility conditions of Theorem 2 are satisfied by all solutions $g_{j0}(x) > 0$, $\forall x \in (0, \infty)$, of the differential equation $g'_{j0}(x) = p_j(g_{j0}(x))$ that satisfy the conditions $r_{j0}(x) > 0$, $\forall x \in (0, \infty)$, for $m > 1$ and $r_{j0}(x) < 0$, $\forall x \in (0, \infty)$, for $m < 1$. Generator functions admissible in the strict sense can be obtained by determining the subset of the resulting wide-sense admissible generator functions that satisfy the fourth condition of Theorem 2. The constructive approach outlined above is employed here to produce *increasing* generator functions that can be used for $m > 1$. The same constructive approach can be extended to produce *decreasing* generator functions that can be used for $m < 1$.

Assume that $m > 1$ and let the function $p_j(x)$ be of the form $p_j(x) = k_j x^n$, $k_j > 0$. The function $g_{j0}(x)$ can be obtained in this case by solving the differential equation

$$g'_{j0}(x) = k_j \left(g_{j0}(x)\right)^n, \; k_j > 0. \tag{4}$$

According to (4), $g''_{j0}(x) = k_j n (g_{j0}(x))^{n-1} g'_{j0}(x) = k_j^2 n (g_{j0}(x))^{2n-1}$. In this case

$$r_{j0}(x) = \left(g'_{j0}(x)\right)^2 \left(\frac{m}{m-1} - n\right). \tag{5}$$

If $m > 1$, it is required that $r_{j0}(x) > 0$, $\forall x \in (0, \infty)$, which holds for all $m/(m-1) > n$. For $m > 1$, $m/(m-1) > 1$ and the inequality $m/(m-1) > n$ holds for all $n < 1$. For

$n = 1$, $m/(m-1) - n = 1/(m-1) > 0$. Thus, the condition $r_{j0}(x) > 0$, $\forall x \in (0, \infty)$, is satisfied for all $n \leq 1$.

For $n = 1$, $p_j(x) = k_j x$ and the solutions of (4) are $g_{j0}(x) = \theta_j \exp(\beta_j x)$, where $\theta_j > 0$ and $\beta_j = k_j/\theta_j > 0$. These generator functions also satisfy the fourth axiomatic requirement since $\lim_{x \to 0_+} g_{j0}(x) = \theta_j > 0$. The exponential generator functions $g_{j0}(x) = \exp(\beta_j x)$, $\beta_j > 0$, obtained for $\theta_j = 1$ correspond to $g_j(x) = \exp(\beta_j x/(1-m))$, which lead to Gaussian radial basis functions $\phi_j(x) = g_j(x^2) = \exp(-x^2/\sigma_j^2)$, with $\sigma_j^2 = (m-1)/\beta_j$.

For $n < 1$, the solutions of (4) are of the form $g_{j0}(x) = (a_j x + b_j)^{1/(1-n)}$, where $a_j = k_j(1-n) > 0$ and $b_j \geq 0$. For $n = 0$, $p_j(x) = k_j$ and (4) leads to linear generator functions $g_{j0}(x) = a_j x + b_j$, $a_j > 0$, $b_j \geq 0$. For $g_{j0}(x) = a_j x + b_j$, the fourth axiomatic requirement is satisfied only if $\lim_{x \to 0_+} g_{j0}(x) = b_j > 0$. Thus, the fourth axiomatic requirement excludes generator functions of the form $g_{j0}(x) = a_j x$. Linear generator functions produce radial basis functions of the form $\phi_j(x) = g_j(x^2) = (a_j x^2 + b_j)^{1/(1-m)}$, with $m > 1$. If $a_j = 1$ and $b_j = \gamma_j^2$, then the linear generator function $g_{j0}(x) = a_j x + b_j$ becomes $g_{j0}(x) = x + \gamma_j^2$ and $g_j(x) = (x + \gamma_j^2)^{1/(1-m)}$. If $m = 3$, $g_j(x) = (x + \gamma_j^2)^{-1/2}$ corresponds to the inverse multiquadratic radial basis function $\phi_j^q(x) = g_j(x^2) = 1/(x^2 + \gamma_j^2)^{1/2}$ [6], [8], [10]. Another useful generator function for practical applications can be obtained from $g_{j0}(x) = a_j x + b_j$ by selecting $b_j = 1$ and $a_j = \delta_j > 0$. For $g_{j0}(x) = 1 + \delta_j x$, $\lim_{x \to 0_+} g_j(x) = \lim_{x \to 0_+} g_{j0}(x) = 1$. For this choice of parameters, the corresponding radial basis function $\phi_j(x) = g_j(x^2)$ is bounded by 1, which is also the bound of the Gaussian radial basis function.

III. Selecting Generator Functions

The axiomatic requirements impose conditions on the response $h_j = g_j(\|\mathbf{x} - \mathbf{v}_j\|^2)$ of the jth radial basis function to the input vector $\mathbf{x}$ and the norm $\|\nabla_{\mathbf{x}} h_j\|^2$ of the gradient $\nabla_{\mathbf{x}} h_j$. Thus, the selection of generator functions suitable for gradient descent learning can be accomplished by focusing on:

1) the response $h_j = g_j(\|\mathbf{x} - \mathbf{v}_j\|^2)$ of the jth radial basis function to an input vector $\mathbf{x}$;
2) the *absolute sensitivity* of h_j with respect to $\mathbf{x}$, defined as $S_j^A = \|\nabla_{\mathbf{x}} h_j\|^2$.

Since $h_j = g_j(\|\mathbf{x} - \mathbf{v}_j\|^2)$, the gradient $\nabla_{\mathbf{x}} h_j$ can be obtained as $\nabla_{\mathbf{x}} h_j = -\alpha_j(\mathbf{x} - \mathbf{v}_j)$, where $\alpha_j = -2g_j'(\|\mathbf{x} - \mathbf{v}_j\|^2)$ and $g_j'(\|\mathbf{x} - \mathbf{v}_j\|^2)$ denotes the derivative of $g_j(\|\mathbf{x} - \mathbf{v}_j\|^2)$ with respect to $\|\mathbf{x} - \mathbf{v}_j\|^2$. The norm $\|\nabla_{\mathbf{x}} h_j\|^2$ of the gradient $\nabla_{\mathbf{x}} h_j$ is

$$\|\nabla_{\mathbf{x}} h_j\|^2 = \|\mathbf{x} - \mathbf{v}_j\|^2 \alpha_j^2. \tag{6}$$

For $g_j(x) = (g_{j0}(x))^{1/(1-m)}$, $g_j'(x) = [1/(1-m)](g_j(x))^m g_{j0}'(x)$. Since $h_j = g_j(\|\mathbf{x} - \mathbf{v}_j\|^2)$, α_j is given by

$$\alpha_j = \frac{2}{m-1} (h_j)^m g_{j0}' \left(\|\mathbf{x} - \mathbf{v}_j\|^2 \right). \tag{7}$$

For linear generator functions of the form $g_{j0}(x) = 1 + \delta_j x$, $\alpha_j = [2\delta_j/(m-1)](h_j)^m$. For increasing exponential generator functions $g_{j0}(x) = \exp(\beta_j x)$, $\alpha_j = [2\beta_j/(m-1)]h_j$.

A. Blind Spot

Axiom 3 requires that $\|\nabla_{\mathbf{x}} h_j\|^2 / \|\mathbf{x} - \mathbf{v}_j\|^2$ be a monotonically decreasing function of $\|\mathbf{x} - \mathbf{v}_j\|^2$. However, the condition in (2) does not necessarily guarantee that the inequality

$$\|\nabla_{\mathbf{x}_k} h_{j,k}\|^2 > \|\nabla_{\mathbf{x}_\ell} h_{j,\ell}\|^2 \tag{8}$$

is valid if $\|\mathbf{x}_k - \mathbf{v}_j\|^2 < \|\mathbf{x}_\ell - \mathbf{v}_j\|^2$. This can easily be seen from the condition in (2) by considering that $\mathbf{y}$ is fixed and $\mathbf{x}$ approaches $\mathbf{v}$. This observation implies that there may exist a region $\|\mathbf{x} - \mathbf{v}_j\|^2 \in (0, B_j)$ around the prototype $\mathbf{v}_j$ where the absolute sensitivity $S_j^A = \|\nabla_{\mathbf{x}} h_j\|^2$ is an increasing function of $\|\mathbf{x} - \mathbf{v}_j\|^2$ even if Axiom 3 is satisfied. For all input vectors $\mathbf{x}$ that satisfy $\|\mathbf{x} - \mathbf{v}_j\|^2 < B_j$, the norm of the gradient $\nabla_{\mathbf{x}} h_j$ corresponding to the jth radial basis function decreases as $\mathbf{x}$ approaches its center that is located at the prototype $\mathbf{v}_j$. As far as gradient descent learning is concerned, the input vectors in the region $\|\mathbf{x} - \mathbf{v}_j\|^2 \in (0, B_j)$ are not "visible" by the radial basis function centered at the prototype $\mathbf{v}_j$. Accordingly, the *blind spot* of the radial basis function centered at the prototype $\mathbf{v}_j$ is the hypersphere $\mathcal{R}_j^{\text{bl}}$, defined as

$$\mathcal{R}_j^{\text{bl}} = \left\{ \mathbf{x} \in \mathbb{R}^n : \|\mathbf{x} - \mathbf{v}_j\|^2 \in (0, B_j) \right\}. \tag{9}$$

According to (6), the absolute sensitivity $S_j^A = \|\nabla_{\mathbf{x}} h_j\|^2$ can be written as $S_j^A = 4t_j(\|\mathbf{x} - \mathbf{v}_j\|^2)$, where $t_j(x) = x[g_j'(x)]^2$. The absolute sensitivity S_j^A is monotonically decreasing in the interval (B_j, ∞) if there exists a $B_j \geq 0$ such that $t_j'(x) < 0$, $\forall x \in (B_j, \infty)$. Since $t_j(x) = x[g_j'(x)]^2$, $t_j'(x) = g_j'(x) d_j(x)$, where $d_j(x) = g_j'(x) + 2x g_j''(x)$. The absolute sensitivity S_j^A is monotonically decreasing in the interval (B_j, ∞) if the function $t_j(x) = x[g_j'(x)]^2$ has a maximum at $x = B_j$. In such a case, B_j can be obtained as the solution of the equation $t_j'(x) = 0$. Theorem 1 requires that $g_j(x)$ be a decreasing function of $x \in (0, \infty)$, which implies that $g_j'(x) < 0$, $\forall x \in (0, \infty)$. Since $t_j'(x) = g_j'(x) d_j(x)$, B_j can also be obtained as the solution of

$$d_j(x) = g_j'(x) + 2x g_j''(x) = 0. \tag{10}$$

For linear generator functions $g_{j0}(x) = 1 + \delta_j x$, the absolute sensitivity $S_j^A = \|\nabla_{\mathbf{x}} h_j\|^2$ can be obtained using (6) and (7) as $S_j^A = [2\delta_j/(m-1)]^2 (h_j)^{2m} \|\mathbf{x} - \mathbf{v}_j\|^2$. If $g_j(x) = (g_{j0}(x))^{1/(1-m)}$, the blind spot $\mathcal{R}_j^{\text{bl}}$ corresponding to $g_{j0}(x) = 1 + \delta_j x$ is determined by B_j, which can be obtained by solving $d_j(x) = 0$ as

$$B_j = \frac{m-1}{m+1} \frac{1}{\delta_j}. \tag{11}$$

Fig. 1(a) shows the response $h_j = g_j(\|\mathbf{x} - \mathbf{v}_j\|^2)$ and the absolute sensitivity $S_j^A = \|\nabla_{\mathbf{x}} h_j\|^2$ plotted as functions of $\|\mathbf{x} - \mathbf{v}_j\|^2$ for $g_j(x) = (g_{j0}(x))^{1/(1-m)}$, with $g_{j0}(x) = 1 + \delta_j x$, $m = 3$ and $\delta_j = 10$. In accordance with the analysis, the absolute sensitivity S_j^A increases monotonically as $\|\mathbf{x} - \mathbf{v}_j\|^2$ increases from 0 to $B_j = 1/(2\delta_j)$ and decreases monotonically as $\|\mathbf{x} - \mathbf{v}_j\|^2$ increases above B_j. For a fixed value of $m > 1$, B_j decreases as δ_j increases. Thus, increasing the value of δ_j shrinks the blind spot of the corresponding radial basis function. Moreover, the peak value $(S_j^A)_{\max} = [1/(m^2 - 1)][(m +$

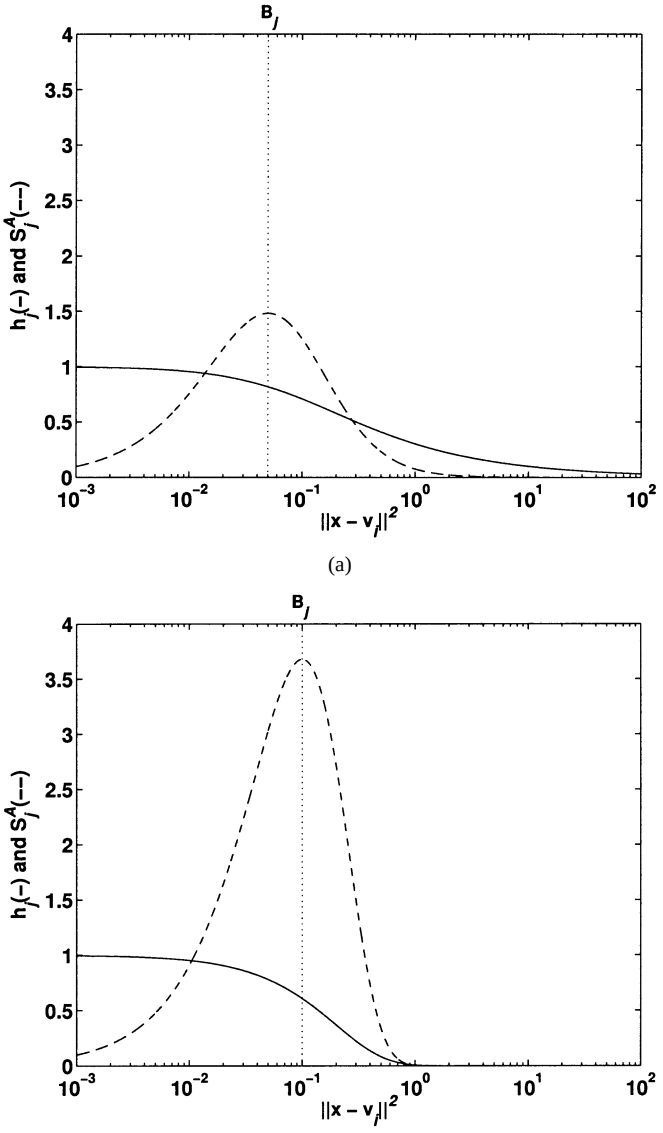

Fig. 1. Response $h_j = g_j(\|\mathbf{x} - \mathbf{v}_j\|^2)$ of the jth radial basis function (solid line) plotted as a function of $\|\mathbf{x} - \mathbf{v}_j\|^2$ together with the absolute sensitivity $S_j^A = \|\nabla_{\mathbf{x}} h_j\|^2$ (dotted line) for (a) $g_j(x) = (g_{j0}(x))^{1/(1-m)}$, with $g_{j0}(x) = 1 + \delta_j x$, $m = 3$, $\delta_j = 10$, and (b) $g_{j0}(x) = \exp(\beta_j x)$, $m = 3$, $\beta_j = 10$.

$1)/(2m)]^{2m/(m-1)}\delta_j$ of the absolute sensitivity increases as δ_j increases. However, as the value of δ_j increases, the absolute sensitivity S_j^A diminishes faster as $\|\mathbf{x} - \mathbf{v}_j\|^2$ increases above B_j.

For exponential generator functions $g_{j0}(x) = \exp(\beta_j x)$, the absolute sensitivity $S_j^A = \|\nabla_{\mathbf{x}} h_j\|^2$ can be obtained using (6) and (7) as $S_j^A = [2\beta_j/(m-1)]^2 (h_j)^2 \|\mathbf{x} - \mathbf{v}_j\|^2$. If $g_j(x) = (g_{j0}(x))^{1/(1-m)}$, the blind spot $\mathcal{R}_j^{\text{bl}}$ corresponding to $g_{j0}(x) = \exp(\beta_j x)$ is determined by B_j, which can be obtained by solving $d_j(x) = 0$ as

$$B_j = \frac{m-1}{2\beta_j}. \tag{12}$$

Fig. 1(b) shows the response $h_j = g_j(\|\mathbf{x} - \mathbf{v}_j\|^2)$ and the absolute sensitivity $S_j^A = \|\nabla_{\mathbf{x}} h_j\|^2$ plotted as functions of $\|\mathbf{x} - \mathbf{v}_j\|^2$ for $g_j(x) = (g_{j0}(x))^{1/(1-m)}$, with $g_{j0}(x) = \exp(\beta_j x)$, $m = 3$ and $\beta_j = 10$. The absolute sensitivity S_j^A increases monotonically as $\|\mathbf{x} - \mathbf{v}_j\|^2$ increases from 0 to $B_j = 1/\beta_j$ and decreases monotonically as $\|\mathbf{x} - \mathbf{v}_j\|^2$ increases above B_j. As the value of β_j increases, B_j decreases and the blind spot shrinks. Increasing the value of β_j increases the peak value $(S_j^A)_{max} = [(2e^{-1})/(m-1)]\beta_j$ of the absolute sensitivity S_j^A but the values of the response and the absolute sensitivity diminish faster for values of $\|\mathbf{x} - \mathbf{v}_j\|^2$ above B_j.

B. Linear Versus Exponential Generator Functions

Linear and exponential generator functions can be compared and evaluated in terms of their suitability for gradient descent learning by employing a criterion based on the concept of the blind spot presented above. The update of the prototypes by gradient descent is mostly affected by neighboring training vectors located outside the blind spots of their corresponding radial basis functions. Training radial basis function neural networks by a learning procedure based on gradient descent requires that the response h_j and the absolute sensitivity $S_j^A = \|\nabla_{\mathbf{x}} h_j\|^2$ take substantial values outside the blind spot before they approach zero. In order to guarantee a certain degree of overlapping between the radial basis functions, it is also required that the response h_j be sizable outside the blind spot even after the values of the absolute sensitivity S_j^A become negligible. The application of this criterion relies on the peak value attained by the absolute sensitivity $S_j^A = S_j^A(\|\mathbf{x} - \mathbf{v}_j\|^2)$ at $\|\mathbf{x} - \mathbf{v}_j\|^2 = B_j$ and the rate at which h_j and S_j^A decrease as $\|\mathbf{x} - \mathbf{v}_j\|^2$ increases above B_j. Comparison of Fig. 1(a) and (b) indicates that the absolute sensitivity corresponding to exponential generator functions $g_{j0}(x) = \exp(\beta_j x)$ reaches a higher peak value than that corresponding to linear generator functions $g_{j0}(x) = 1 + \delta_j x$ when the values of β_j and δ_j are the same. However, the absolute sensitivity corresponding to linear generator functions decreases at a much lower rate outside the blind spot compared with that corresponding to exponential generator functions. According to Fig. 1(a) and (b), the response of radial basis functions corresponding to linear generator functions is sizable outside the blind spot even after the values of the absolute sensitivity become negligible. In contrast, the response of radial basis functions corresponding to exponential generator functions becomes negligible even before the absolute sensitivity approaches practically zero values.

C. Estimating the Free Parameters of Radial Basis Functions

Reformulated radial basis function neural networks can be trained by fixing the values of the free parameters that determine the shapes of the radial basis functions and updating only the output weights and the prototypes using gradient descent. In such a case, the free parameters of the radial basis functions can be estimated in the beginning of the learning process in such a way that all training vectors lie outside the blind spots of the radial basis functions. The training vectors $\mathbf{x}_k \in \mathcal{X}$ lie outside the blind spot $\mathcal{R}_j^{\text{bl}} = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x} - \mathbf{v}_j\|^2 \in (0, B_j)\}$ of the radial basis function centered at the prototype $\mathbf{v}_j$ if

$$B_j \leq \min_{\mathbf{x}_k \in \mathcal{X}} \left\{ \|\mathbf{x}_k - \mathbf{v}_j\|^2 \right\}. \tag{13}$$

For linear generator functions of the form $g_{j0}(x) = 1 + \delta_j x$, $B_j = (m-1)/[(m+1)\delta_j]$ and condition (13) holds as an equality if

$$\delta_j = \frac{m-1}{m+1} \frac{1}{\min_{\mathbf{x}_k \in \mathcal{X}} \left\{ \|\mathbf{x}_k - \mathbf{v}_j\|^2 \right\}}. \tag{14}$$

For exponential generator functions of the form $g_{j0}(x) = \exp(\beta_j x)$, $B_j = (m-1)/(2\beta_j)$ and condition (13) holds as an equality if

$$\beta_j = \frac{m-1}{2} \frac{1}{\min_{\mathbf{x}_k \in \mathcal{X}} \left\{ \|\mathbf{x}_k - \mathbf{v}_j\|^2 \right\}}. \tag{15}$$

The shapes of the radial basis functions can be determined in practice by fixing m to an integer value between 2 and 4 and computing the values of their free parameters $\{\delta_j\}$ or $\{\beta_j\}$ according to (14) or (15), respectively.

IV. Cosine Radial Basis Functions

Consider that $g_j(x) = (g_{j0}(x))^{1/(1-m)}$, where $g_{j0}(x)$ is the linear generator function $g_{j0}(x) = 1 + \delta_j x$, $0 < \delta_j < \infty$. If $m = 3$, then $g_j(x) = (1 + \delta_j x)^{-1/2}$. If $\delta_j = 1/a_j^2$, with $a_j \neq 0$, then

$$g_j(x) = \frac{a_j}{\left(x + a_j^2\right)^{1/2}}. \tag{16}$$

The corresponding radial basis function $\phi_j(x) = g_j(x^2)$ can be obtained from (16) as

$$\phi_j(x) = \frac{a_j}{\left(x^2 + a_j^2\right)^{1/2}}. \tag{17}$$

The resulting radial basis function (17) has some resemblance with the inverse multiquadratic radial basis function $\phi_j^q(x) = 1/(x^2 + \gamma_j^2)^{1/2}$. However, the basis function defined in (17) satisfies $\phi_j(0) = 1$. In contrast, the value of the multiquadratic radial basis function at the origin is a function of γ_j, as indicated by $\phi_j^q(0) = \gamma_j^{-1}$. According to the definition of the multiquadratic function

$$\frac{\phi_j^q(x)}{\phi_j^q(0)} = \frac{\gamma_j}{\left(x^2 + \gamma_j^2\right)^{1/2}}. \tag{18}$$

Thus, the proposed radial basis function (17) can also be seen as a normalized version of the multiquadratic radial basis function. The response $h_{j,k} = g_j(\|\mathbf{x}_k - \mathbf{v}_j\|^2)$ of the radial basis function centered at the prototype $\mathbf{v}_j$ to the input $\mathbf{x}_k$ can be obtained according to (16) as

$$h_{j,k} = \frac{a_j}{\left(\|\mathbf{x}_k - \mathbf{v}_j\|^2 + a_j^2\right)^{1/2}}. \tag{19}$$

According to (19), $h_{j,k}$ is a monotonically decreasing function of $\|\mathbf{x}_k - \mathbf{v}_j\|^2$ for any nonzero value of a_j. In fact, $h_{j,k}$ approaches 0 as $\|\mathbf{x}_k - \mathbf{v}_j\|^2$ approaches infinity while $h_{j,k} = 1$ if $\mathbf{x}_k$ coincides with $\mathbf{v}_j$. For a fixed value of $\|\mathbf{x}_k - \mathbf{v}_j\|^2$, $h_{j,k}$ is an increasing function of a_j. As a_j approaches zero, h_{jk} also approaches zero. As a_j increases and approaches ∞, $h_{j,k}$ increases and approaches 1 regardless of the value of $\|\mathbf{x}_k - \mathbf{v}_j\|^2$. The parameters $\{a_j\}$ play a critical role in making the responses $\{h_{j,k}\}$ defined in (19) an effective similarity measure between the training vectors and the prototype $\mathbf{v}_j$. The analysis which follows presents a geometric interpretation of the radial basis function introduced above.

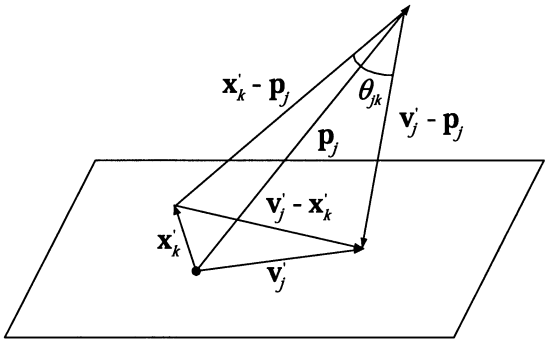

Fig. 2. Geometric interpretation of cosine radial basis functions for $\mathbf{x}_k \in \mathbb{R}^2$ and $\mathbf{v}_j \in \mathbb{R}^2$.

Let $\mathbf{x}_k = [x_{1k}, x_{2k}, \ldots, x_{nk}]^T \in \mathbb{R}^n$ be a training vector and $\mathbf{v}_j = [v_{1j}, v_{2j}, \ldots, v_{nj}]^T \in \mathbb{R}^n$ be a prototype. Define a new vector $\mathbf{x}_k' \in \mathbb{R}^{n+1}$ obtained by augmenting $\mathbf{x}_k$ as $\mathbf{x}_k' = [\mathbf{x}_k^T, 0]^T = [x_{1k}, x_{2k}, \ldots, x_{nk}, 0]^T \in \mathbb{R}^{n+1}$. Augmenting the prototype $\mathbf{v}_j$ in a similar manner gives $\mathbf{v}_j' = [\mathbf{v}_j^T, 0]^T = [v_{1j}, v_{2j}, \ldots, v_{nj}, 0]^T \in \mathbb{R}^{n+1}$. Let $\mathbf{p}_j$ be a vector obtained by augmenting the prototype $\mathbf{v}_j$ as $\mathbf{p}_j = [\mathbf{v}_j^T, a_j]^T = [v_{1j}, v_{2j}, \ldots, v_{nj}, a_j]^T \in \mathbb{R}^{n+1}$ with $a_j \in \mathbb{R} - \{0\}$. For $n = 2$, $\mathbf{x}_k \in \mathbb{R}^2$ and $\mathbf{v}_j \in \mathbb{R}^2$ lie in a plane. The vectors $\mathbf{x}_k'$ and $\mathbf{v}_j'$ obtained by augmenting $\mathbf{x}_k$ and $\mathbf{v}_j$ respectively, are defined in $\mathbb{R}^3$. For $a_j \neq 0$, $\mathbf{p}_j$ defines a vector $\mathbf{v}_j' - \mathbf{p}_j$ which is orthogonal to the plane that contains $\mathbf{x}_k \in \mathbb{R}^2$ and $\mathbf{v}_j \in \mathbb{R}^2$ (see Fig. 2). Using the definitions of $\mathbf{p}_j$, $\mathbf{v}_j'$, and $\mathbf{x}_k'$, $\|\mathbf{x}_k - \mathbf{v}_j\|^2 + a_j^2 = \|\mathbf{x}_k' - \mathbf{p}_j\|^2$ and the response $h_{j,k}$ defined in (19) can be written as

$$h_{j,k} = \frac{a_j}{\|\mathbf{x}_k' - \mathbf{p}_j\|}. \tag{20}$$

For $\mathbf{x}_k \in \mathbb{R}^2$ and $\mathbf{v}_j \in \mathbb{R}^2$, Fig. 2 indicates that $\|\mathbf{v}_j' - \mathbf{x}_k'\| = \|\mathbf{v}_j - \mathbf{x}_k\|$ and $\|\mathbf{v}_j' - \mathbf{p}_j\| = a_j$ are the lengths of the legs of a right triangle, $\|\mathbf{x}_k' - \mathbf{p}_j\|$ is the length of its hypotenuse, and

$$h_{j,k} = \cos(\theta_{jk}) \tag{21}$$

where θ_{jk} is the angle between $\mathbf{v}_j' - \mathbf{p}_j$ and $\mathbf{x}_k' - \mathbf{p}_j$. In general, the response of a radial basis function centered at the prototype $\mathbf{v}_j$ to an input vector $\mathbf{x}_k$ measures the similarity between this input vector and the prototype $\mathbf{v}_j$. The similarity between $\mathbf{v}_j$ and $\mathbf{x}_k$ is measured in this case by using $\mathbf{p}_j$ as a reference for computing the cosine of the angle θ_{jk} between $\mathbf{v}_j' - \mathbf{p}_j$ and $\mathbf{x}_k' - \mathbf{p}_j$, which can also be obtained as

$$\cos(\theta_{jk}) = \frac{\left(\mathbf{v}_j' - \mathbf{p}_j\right)^T \left(\mathbf{x}_k' - \mathbf{p}_j\right)}{\left\|\mathbf{v}_j' - \mathbf{p}_j\right\| \left\|\mathbf{x}_k' - \mathbf{p}_j\right\|}. \tag{22}$$

According to this interpretation, a_j represents the distance of the reference point from the corresponding prototype $\mathbf{v}_j$ or, simply, the *reference distance*.

A. Sensitivity Analysis of Cosine Radial Basis Functions

The mapping realized by cosine radial basis functions depends on the prototypes $\{\mathbf{v}_j\}$ and the reference distances $\{a_j\}$. Thus, a learning scheme relying on gradient descent would mainly depend on the gradient of h_j with respect to $\mathbf{v}_j$, which can be measured by $\|\nabla_{\mathbf{v}_j} h_j\|^2$, and the rate at which h_j changes with a_j, which can be measured by $\partial h_j/\partial a_j$. Since $\|\nabla_{\mathbf{v}_j} h_j\|^2 = \|\nabla_{\mathbf{x}} h_j\|^2$, the suitability of cosine radial basis functions for gradient descent learning can be investigated by focusing on the absolute sensitivity $S_j^A = \|\nabla_{\mathbf{x}} h_j\|^2$ of h_j with respect to inputs $\mathbf{x} \in \mathbb{R}^n$. This investigation can be facilitated by the concept of the blind spot, which was introduced in Section III.

The absolute sensitivity $S_j^A = \|\nabla_{\mathbf{x}} h_j\|^2$ of cosine radial basis functions can be obtained as

$$S_j^A = a_j^2 \frac{\|\mathbf{x}-\mathbf{v}_j\|^2}{\left(\|\mathbf{x}-\mathbf{v}_j\|^2 + a_j^2\right)^3} = \frac{h_j^4}{a_j^2}\left(1-h_j^2\right). \tag{23}$$

The value of $\|\mathbf{x}-\mathbf{v}_j\|^2 = B_j$ for which $S_j^A = S_j^A(\|\mathbf{x}-\mathbf{v}_j\|^2)$ reaches its maximum value can be determined as $B_j = a_j^2/2$. This implies that the blind spot of the cosine radial basis function centered at $\mathbf{v}_j$ shrinks as a_j decreases. However, reducing a_j increases the rate at which the response $h_j = h_j(\|\mathbf{x}-\mathbf{v}_j\|^2)$ diminishes when $\|\mathbf{x}-\mathbf{v}_j\|^2$ increases. The behavior of $\|\nabla_{\mathbf{v}_j} h_j\|^2$ in a neighborhood centered at $\mathbf{v}_j$ is shown in Fig. 3(a), which plots $S_j^A = \|\nabla_{\mathbf{x}} h_j\|^2$ together with $h_j = g_j(\|\mathbf{x}-\mathbf{v}_j\|^2)$ as functions of $\|\mathbf{x}-\mathbf{v}_j\|^2$ for $a_j = 0.5$.

If $h_j = g_j(\|\mathbf{x}-\mathbf{v}_j\|^2)$ and $g_j(\cdot)$ is defined in (16), then it can be shown that

$$\frac{\partial h_j}{\partial a_j} = \frac{\|\mathbf{x}-\mathbf{v}_j\|^2}{\left(\|\mathbf{x}-\mathbf{v}_j\|^2 + a_j^2\right)^{3/2}} = \frac{h_j}{a_j}\left(1-h_j^2\right). \tag{24}$$

According to (24), $\partial h_j/\partial a_j$ is an increasing function of $\|\mathbf{x}-\mathbf{v}_j\|^2$ for $\|\mathbf{x}-\mathbf{v}_j\|^2 \in (0, 2a_j^2)$ and a decreasing function of $\|\mathbf{x}-\mathbf{v}_j\|^2$ for $\|\mathbf{x}-\mathbf{v}_j\|^2 > 2a_j^2$. The behavior of $\partial h_j/\partial a_j$ in a neighborhood centered at $\mathbf{v}_j$ is shown in Fig. 3(b), which plots $\partial h_j/\partial a_j$ together with $h_j = g_j(\|\mathbf{x}-\mathbf{v}_j\|^2)$ as functions of $\|\mathbf{x}-\mathbf{v}_j\|^2$ for $a_j = 0.5$. Fig. 3 indicates that there always exists a region of the input space where $\|\nabla_{\mathbf{v}_j} h_j\|^2$ is a decreasing function of $\|\mathbf{x}-\mathbf{v}_j\|^2$ and $\partial h_j/\partial a_j$ takes substantial values, including its maximum value. Thus, training cosine radial basis function models by gradient descent implies that the adjustable parameters of the jth radial basis function will be mostly sensitive to input vectors $\mathbf{x}$ for which $\|\mathbf{x}-\mathbf{v}_j\|^2 \in (a_j^2/2, 2a_j^2)$. The *active region* of the cosine radial basis function centered at the prototype $\mathbf{v}_j$ is defined as

$$\mathcal{R}_j^{\text{act}} = \left\{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}-\mathbf{v}_j\|^2 \in \left(\frac{a_j^2}{2}, 2a_j^2\right)\right\}. \tag{25}$$

The active region of the jth radial basis function depends exclusively on a_j; this reveals the critical role of the reference distance in the implementation of the desired input-output mapping.

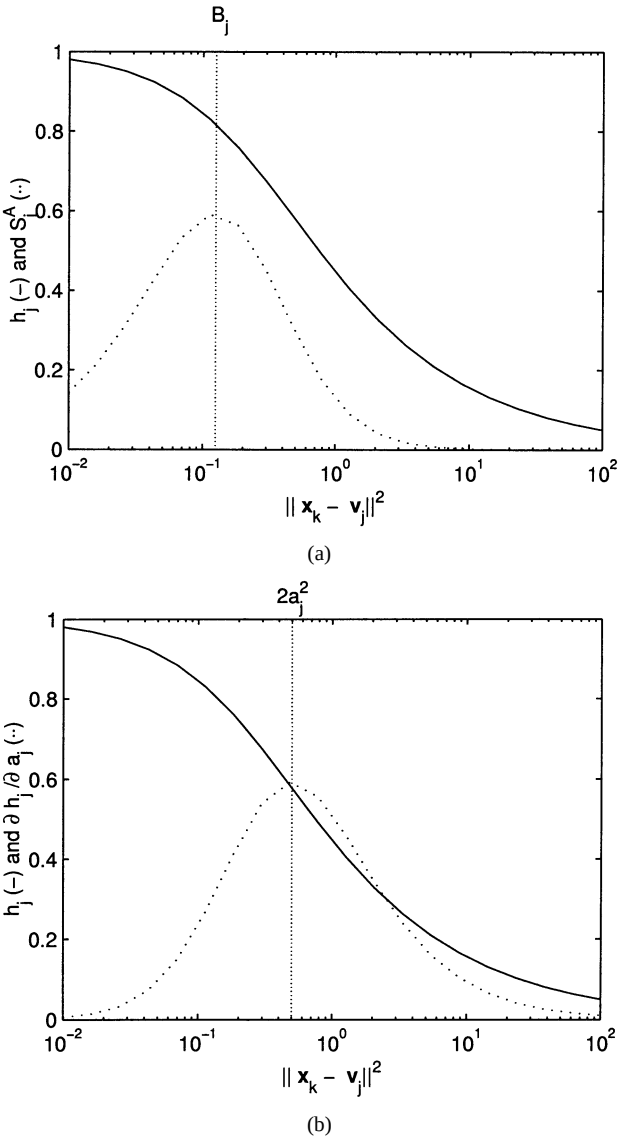

Fig. 3. Response $h_j = \phi_j(\|\mathbf{x}-\mathbf{v}_j\|)$ of the jth radial basis function (solid line) plotted as a function of $\|\mathbf{x}-\mathbf{v}_j\|^2$ together with (a) the absolute sensitivity $S_j^A = \|\nabla_{\mathbf{x}} h_j\|^2$ (dotted line) and (b) $\partial h_j/\partial a_j$ (dotted line). radial basis functions: $\phi_j(x) = a_j/(x^2+a_j^2)^{1/2}$ with $a_j = 0.5$.

Training of a cosine radial basis function neural network by gradient descent requires that the active regions of the available radial basis functions cover completely the input space. The definition of the active region in (25) indicates that the likelihood of complete coverage of the input space by active regions of radial basis functions improves considerably as the values of $\{a_j\}$ increase. Thus, the training process can be facilitated by updating the reference distances $\{a_j\}$ based on the requirements of the desired input–output mapping. It is expected that updating $\{a_j\}$ during the learning process would allow the implementation of a desired input–output mapping by cosine radial basis function neural networks with a relatively small number of radial basis functions. This is due to the fact that the reduction of the number of radial basis functions can be compensated during learning by increasing the values of $\{a_j\}$, which is expected to expand the active regions of the corresponding radial basis functions. Reducing the number of radial basis functions is expected to im-

prove the generalization ability of the corresponding cosine radial basis function neural network.

V. Training Reformulated Radial Basis Function Neural Networks

radial basis function neural networks can be trained to map $\mathbf{x}_k \in \mathbb{R}^n$ into $\mathbf{y}_k = [y_{1,k} \quad y_{2,k} \quad \dots \quad y_{p,k}]^T \in \mathbb{R}^p$, where the vector pairs $(\mathbf{x}_k, \mathbf{y}_k)$, $1 \leq k \leq M$, form the training set. If $\mathbf{x}_k \in \mathbb{R}^n$ is the input to an radial basis function neural network, its response is $\hat{\mathbf{y}}_k = [\hat{y}_{1,k} \quad \hat{y}_{2,k} \quad \dots \quad \hat{y}_{p,k}]^T$, where $\hat{y}_{i,k}$ is the actual response of the ith output unit to $\mathbf{x}_k$ given by $\hat{y}_{i,k} = f(\bar{y}_{i,k}) = f(\mathbf{w}_i^T \mathbf{h}_k)$, where $\mathbf{h}_k = [h_{0,k} \quad h_{1,k} \quad \dots \quad h_{c,k}]^T$, $h_{0,k} = 1$, $1 \leq k \leq M$, $h_{j,k} = g_j(\|\mathbf{x}_k - \mathbf{v}_j\|^2)$, $1 \leq j \leq c$, with $g_j(x) = (g_{j0}(x))^{1/(1-m)}$, and $\mathbf{w}_i = [w_{i,0} \quad w_{i,1} \quad \dots \quad w_{i,c}]^T$.

A. Batch Learning Algorithms

Reformulated radial basis function neural networks can be trained by *batch* learning algorithms, which can be developed by using gradient descent to minimize

$$E = \frac{1}{2} \sum_{k=1}^{M} \sum_{i=1}^{p} (y_{i,k} - \hat{y}_{i,k})^2 . \tag{26}$$

An radial basis function neural network can be trained by a gradient descent algorithm in a sequence of *adaptation cycles*, where an adaptation cycle involves the update of all adjustable parameters of the network. An adaptation cycle begins by incrementing each weight vector $\mathbf{w}_i$, $1 \leq i \leq p$, by the amount $\Delta \mathbf{w}_i = -\alpha \nabla_{\mathbf{w}_i} E$ as [10]

$$\mathbf{w}_i \leftarrow \mathbf{w}_i + \Delta \mathbf{w}_i = \mathbf{w}_i + \alpha \sum_{k=1}^{M} \varepsilon_{i,k}^o \mathbf{h}_k \tag{27}$$

where α is the learning rate and $\varepsilon_{i,k}^o$ is the *output unit error*, given as

$$\varepsilon_{i,k}^o = f'(\bar{y}_{i,k})(y_{i,k} - \hat{y}_{i,k}) . \tag{28}$$

Following the update of these weight vectors, each prototype $\mathbf{v}_j$, $1 \leq j \leq c$, is incremented by an amount $\Delta \mathbf{v}_j = -\alpha \nabla_{\mathbf{v}_j} E$ as [10]

$$\mathbf{v}_j \leftarrow \mathbf{v}_j + \Delta \mathbf{v}_j = \mathbf{v}_j + \alpha \sum_{k=1}^{M} \varepsilon_{j,k}^h (\mathbf{x}_k - \mathbf{v}_j) \tag{29}$$

where α is the learning rate and $\varepsilon_{j,k}^h$ is the *hidden unit error*, defined as

$$\varepsilon_{j,k}^h = \frac{2}{m-1} (h_{j,k})^m g'_{j0} \left(\|\mathbf{x}_k - \mathbf{v}_j\|^2 \right) \sum_{i=1}^{p} \varepsilon_{i,k}^o w_{ij} . \tag{30}$$

If $g_{j0}(x) = 1 + \delta_j x$, then $g'_{j0}(\|\mathbf{x}_k - \mathbf{v}_j\|^2) = \delta_j$. The hidden unit error corresponding to cosine radial basis function neural networks can be obtained from (30) for $m = 3$ and $\delta_j = 1/a_j^2$ as

$$\varepsilon_{j,k}^h = \left(\frac{h_{j,k}^3}{a_j^2} \right) \sum_{i=1}^{p} \varepsilon_{i,k}^o w_{ij} . \tag{31}$$

Training of cosine radial basis function neural networks also involves updates of the reference distances a_j, $1 \leq j \leq c$, which can be incremented by an amount $\Delta a_j = -\eta \partial E / \partial a_j$ as

$$a_j \leftarrow a_j + \Delta a_j = a_j + \left(\frac{\eta}{a_j} \right) \sum_{k=1}^{M} h_{j,k} \left(1 - h_{j,k}^2\right) \varepsilon_{j,k}^h \tag{32}$$

where η is the learning rate and $\varepsilon_{j,k}^h$ is defined in (31).

B. Sequential Learning Algorithms

Reformulated radial basis function neural networks can also be trained "on-line" by *sequential* learning algorithms, which can be developed by using gradient descent to minimize

$$E_k = \frac{1}{2} \sum_{i=1}^{p} (y_{i,k} - \hat{y}_{i,k})^2 \tag{33}$$

for $k = 1, 2, \dots, M$. After an example $(\mathbf{x}_k, \mathbf{y}_k)$, $1 \leq k \leq M$, is presented to the radial basis function neural network, the new estimate $\mathbf{w}_{i,k}$ of each weight vector $\mathbf{w}_i$, $1 \leq i \leq p$, is obtained by incrementing its current estimate $\mathbf{w}_{i,k-1}$ by an amount $\Delta \mathbf{w}_{i,k} = -\alpha \nabla_{\mathbf{w}_i} E_k$ as

$$\mathbf{w}_{i,k} = \mathbf{w}_{i,k-1} + \Delta \mathbf{w}_{i,k} = \mathbf{w}_{i,k-1} + \alpha \varepsilon_{i,k}^o \mathbf{h}_k \tag{34}$$

where α is the learning rate and $\varepsilon_{i,k}^o$ is the output unit error defined in (28). Following the update of all the weight vectors $\mathbf{w}_i$, $1 \leq i \leq p$, the new estimate $\mathbf{v}_{j,k}$ of each prototype $\mathbf{v}_j$, $1 \leq j \leq c$, can be obtained by incrementing its current estimate $\mathbf{v}_{j,k-1}$ by the amount $\Delta \mathbf{v}_{j,k} = -\alpha \nabla_{\mathbf{v}_j} E_k$ as

$$\mathbf{v}_{j,k} = \mathbf{v}_{j,k-1} + \Delta \mathbf{v}_{j,k} = \mathbf{v}_{j,k-1} + \alpha \varepsilon_{j,k}^h (\mathbf{x}_k - \mathbf{v}_{j,k-1}) \tag{35}$$

where α is the learning rate and $\varepsilon_{j,k}^h$ is the hidden unit error defined in (30). The hidden unit error for cosine radial basis function neural networks is given in (31). Finally, the new estimate $a_{j,k}$ of each reference distance a_j, $1 \leq j \leq c$, can be obtained by incrementing its current estimate $a_{j,k-1}$ by the amount $\Delta a_{j,k} = -\eta \partial E_k / \partial a_j$ as

$$\begin{aligned} a_{j,k} &= a_{j,k-1} + \Delta a_{j,k} \\ &= a_{j,k-1} + \left(\frac{\eta}{a_{j,k-1}} \right) h_{j,k} \left(1 - h_{j,k}^2\right) \varepsilon_{j,k}^h . \end{aligned} \tag{36}$$

An adaptation cycle is completed in this case after the sequential presentation to the radial basis function neural network of all the examples included in the training set.

VI. Experimental Results

The performance of cosine radial basis function neural networks was evaluated and compared with that of FFNNs with sigmoid hidden units, conventional radial basis function neural networks with Gaussian radial basis functions, and reformulated radial basis function neural networks constructed by linear generator functions $g_{j0}(x) = 1 + \delta_j x$, $\delta_j > 0$. Conventional radial basis function neural networks were trained by a hybrid learning scheme similar to that proposed by Moody and Darken [17]. The centers of the radial basis functions were determined according to an unsupervised procedure relying on the k-means

algorithm. The output weights connecting the radial basis functions with the output units were updated according to a supervised procedure based on gradient descent. The widths of the Gaussian radial basis functions were computed according to the nearest prototype heuristic [15]. The centers of the radial basis functions were fixed during the supervised learning process. Reformulated radial basis function neural networks were trained by a fully supervised procedure based on gradient descent [12]. This procedure involved the update of the output weights and the centers $\{\mathbf{v}_j\}$ of the radial basis functions. The free parameters $\{\delta_j\}$ were computed according to (14) in accordance with the procedure described in Section III. These parameters were kept fixed during the learning process to avoid substantial fluctuations of the error (26) that were observed in the early stages of the learning process when $\{\delta_j\}$ were included in the adjustable parameters of the network. Such fluctuations are undesirable since they may cause a premature termination of the learning process. Cosine radial basis functions were trained by the sequential gradient descent algorithm described in this paper. The learning rate η used to update the reference distances $\{a_j\}$ was one order of magnitude lower than the learning rate α used to update the output weights and the prototypes. All neural networks were trained using a normalized version of the input data produced by replacing each feature sample x by $\bar{x} = (x - \mu_x)/\sigma_x$, where μ_x and σ_x denote the sample mean and standard deviation of this feature over the entire dataset, respectively. The training of all neural-network models utilized in this experimental study was terminated according to the stopping criterion described below: Each adaptation cycle was followed by the calculation of the average classification error on the testing set over the previous W adaptation cycles. The training was terminated when the average classification error on the testing set began to increase. The trained neural network was selected to be that obtained after the adaptation cycle corresponding to the lowest classification error within the window of length W. The length of the window employed in this experimental study was $W = 100$. Note that the stopping criterion outlined above allows for some fluctuations of the classification error during the learning process. In fact, this stopping criterion becomes increasingly tolerant to fluctuations of the classification error as the length W of the window increases. Finally, the performance of neural-network classifiers trained by employing this stopping criterion is influenced by both training and testing sets.

A. Pima Data

This database[1] is the result of a diagnostic study of diabetes in the adult women in the Pima Indian tribe [7]. There are eight features and two classes. A sample belongs to the first class if diabetes is present and to the second class if it is not. The labeled feature vectors from this dataset were used to form a training set and a testing set, each containing 384 randomly selected samples.

The training set formed from the Pima data was used to train conventional radial basis function neural networks, reformulated radial basis function neural networks constructed by $g_{j0}(x) = 1 + \delta_j x$, and cosine radial basis function neural networks trained by the sequential gradient descent algorithm described in this paper. The neural networks trained in these experiments consisted of eight inputs and one binary output unit, which is 1 if diabetes is present and 0 if it is not. All three radial basis function models were trained with two, four, six, and eight radial basis functions. The same data was also used to train conventional FFNNs with two, four, six, and eight sigmoid hidden units. The results of these experiments are summarized in Table I, which shows the number of adaptation cycles required on average in ten trials for training the neural networks mentioned above, and the percentage of classification errors produced on average by the trained neural networks on the training and testing sets. According to Table I, cosine radial basis function neural networks produced the smallest percentage of classification errors on both training and testing sets among all radial basis function models tested in the experiments. The performance differences among radial basis function models became more significant as the number of radial basis functions decreased. In fact, cosine radial basis function neural networks were strong competitors to conventional FFNNs, which outperformed considerably conventional radial basis function neural networks and reformulated radial basis function neural networks trained by gradient descent.

TABLE I
AVERAGE NUMBER N OF ADAPTATION CYCLES REQUIRED FOR TRAINING VARIOUS RADIAL BASIS FUNCTION NEURAL NETWORKS WITH c RADIAL BASIS FUNCTIONS AND FFNNS WITH n_h HIDDEN UNITS TO CLASSIFY THE PIMA DATA IN TEN TRIALS, AND PERCENTAGE OF CLASSIFICATION ERRORS PRODUCED ON AVERAGE ON THE TRAINING SET (E_{tr}) AND THE TESTING SET (E_{te}). THE NUMBERS IN PARENTHESES REPRESENT THE STANDARD DEVIATION

Conventional RBF NNs				
c	2	4	6	8
N	106.2	108.9	109.1	107.1
E_{tr} (σ_{tr})	33.98 (1.73)	31.17 (1.35)	29.77 (1.43)	29.38 (1.26)
E_{te} (σ_{te})	34.51 (1.28)	31.80 (1.46)	31.35 (1.92)	31.67 (1.51)
Reformulated RBF NNs				
c	2	4	6	8
N	399.2	352.4	331.9	319.5
E_{tr} (σ_{tr})	29.02 (1.52)	26.92 (1.39)	26.57 (0.67)	26.42 (1.12)
E_{te} (σ_{te})	29.49 (2.85)	27.21 (2.48)	24.95 (0.69)	25.74 (1.83)
Cosine RBF NNs				
c	2	4	6	8
N	686.4	524.3	454.3	395.3
E_{tr} (σ_{tr})	23.75 (0.79)	23.13 (0.95)	22.86 (0.79)	22.94 (0.68)
E_{te} (σ_{te})	22.27 (1.35)	21.69 (1.09)	21.12 (0.59)	21.33 (0.53)
Feedforward NNs				
n_h	2	4	6	8
N	124.8	145.3	159.5	148.9
E_{tr} (σ_{tr})	20.86 (0.53)	20.73 (1.26)	21.02 (1.39)	22.40 (1.68)
E_{te} (σ_{te})	21.61 (0.57)	20.96 (0.42)	20.99 (0.63)	21.02 (0.64)

B. Glass Data

This dataset contains information on the chemical composition of various types of glass, represented by nine numeric features [7]. This set of experiments focused on the differentiation between

[1]The Pima, Glass, and Dermatology datasets were acquired from the University of California at Irvine website at http://www.ics.uci.edu/~mlearn/MLRepository.html.

float-processed glass and glass that was not float-processed. This is the problem often considered by users of this dataset. The labeled feature vectors from the Glass dataset were used to form a training set, containing 79 randomly selected samples, and a testing set, containing 84 randomly selected samples.

The training set formed from the Glass data was used for training conventional radial basis function neural networks, reformulated radial basis function neural networks constructed by linear generator functions $g_{j0}(x) = 1 + \delta_j x$, and cosine radial basis function neural networks. The neural networks trained in these experiments consisted of nine inputs and one binary output unit, which is 1 if the glass is float-processed and 0 if it is not. The number of radial basis functions varied in the experiments from two to five. The same dataset was also used to test the performance of conventional FFNNs with two, three, four, and five sigmoid hidden units. Table II shows the number of adaptation cycles required on average in ten trials to train the neural networks mentioned above and the average percentage of feature vectors from the training and testing sets classified incorrectly by the trained neural networks. The reformulated radial basis function neural networks and the proposed cosine radial basis function neural networks exhibited comparable performance on this dataset. In fact, their performance was very satisfactory even when they contained few radial basis functions. Moreover, reformulated radial basis function neural networks and cosine radial basis function models outperformed considerably conventional radial basis function neural networks of the same size and conventional FFNNs.

TABLE II
AVERAGE NUMBER N OF ADAPTATION CYCLES REQUIRED FOR TRAINING VARIOUS RADIAL BASIS FUNCTION NEURAL NETWORKS WITH c RADIAL BASIS FUNCTIONS AND FFNNS WITH n_h HIDDEN UNITS TO CLASSIFY THE GLASS DATA IN TEN TRIALS, AND PERCENTAGE OF CLASSIFICATION ERRORS PRODUCED ON AVERAGE ON THE TRAINING SET (E_{tr}) AND THE TESTING SET (E_{te}). THE NUMBERS IN PARENTHESES REPRESENT THE STANDARD DEVIATION

Conventional RBF NNs				
c	2	3	4	5
N	219	216	213.7	215.6
E_{tr} (σ_{tr})	29.87 (3.68)	33.16 (2.32)	32.28 (4.01)	29.75 (4.78)
E_{te} (σ_{te})	30.48 (3.16)	32.14 (1.51)	30.71 (3.23)	28.81 (4.64)
Reformulated RBF NNs				
c	2	3	4	5
N	776.6	758.9	712	615.4
E_{tr} (σ_{tr})	21.77 (2.18)	21.39 (2.00)	21.39 (2.36)	20.76 (1.62)
E_{te} (σ_{te})	23.81 (1.99)	22.98 (0.93)	23.69 (1.64)	22.86 (1.17)
Cosine RBF NNs				
c	2	3	4	5
N	617.1	609.6	587.4	576.4
E_{tr} (σ_{tr})	25.82 (2.21)	23.42 (1.98)	23.92 (1.65)	22.28 (1.41)
E_{te} (σ_{te})	24.40 (3.16)	23.33 (0.95)	23.21 (2.68)	22.50 (1.80)
Feedforward NNs				
n_h	2	3	4	5
N	298.5	276.8	266	260.3
E_{tr} (σ_{tr})	25.95 (1.90)	27.34 (3.16)	27.47 (2.89)	28.10 (1.86)
E_{te} (σ_{te})	32.02 (1.80)	33.21 (2.52)	32.98 (1.85)	33.57 (2.05)

C. Dermatology Data

The Dermatology dataset provides the basis for determining the type of eryhemato-squamous disease based on 34 clinical and histopathological features. The samples of this dataset belong to six classes, which represent the following diseases: 1) psoriasis, 2) seboreic dermatitis, 3) lichen planus, 4) pityriasis rosea, 5) chronic dermatitis, and 6) pityriasis rubra pilaris. Each feature vector of this dataset contains 34 features. The first 11 features are clinical features. Each clinical feature takes a value in the range 0 to 3, which represents the degree to which this feature was present. More specifically, 0 indicates that the feature was not present, 3 indicates the highest degree of confidence that the feature was present, while 1 and 2 indicate relative intermediate values. The only exception is the family history, which takes the value of 1 if any of these diseases has been observed in the family and 0 otherwise. The next 22 features are histopathological attributes, which were determined by an analysis of the samples under a microscope. The histopathological features take the values 0, 1, 2, and 3. The last feature is the patient's age. After removing the samples with missing values, this dataset contained 358 labeled feature vectors that were used to form the training and testing sets.

The training set was used to train conventional radial basis function neural networks, reformulated radial basis function neural networks generated by linear generator functions $g_{j0}(x) = 1 + \delta_j x$, and cosine radial basis function neural networks. All radial basis function models were trained with two, four, six, and eight radial basis functions in ten trials. The same classification task was performed by training conventional FFNNs with two, four, six, and eight sigmoid hidden units. The neural networks trained in these experiments consisted of 34 inputs and six binary output units, each representing one of the six classes. Each input vector was assigned to the class represented by the output unit of the trained neural network with the maximum response. The results of these experiments are summarized in Table III, which shows the number of adaptation cycles required on average for training the neural-network models in ten trials and the percentage of classification errors produced on the training and testing sets by the trained neural networks mentioned above. The performance of all radial basis function neural networks tested improved considerably as the number of radial basis functions increased. The proposed cosine radial basis function neural networks outperformed considerably all radial basis function neural networks tested in the experiments. It is remarkable the cosine radial basis function neural networks achieved very satisfactory performance when the number of radial basis functions was as low as $c = 4$. Among all radial basis function models tested, only cosine radial basis function neural networks were comparable with FFNNs in terms of their performance on both the training and testing sets. On average, cosine radial basis function neural networks converged in fewer adaptation cycles than FFNNs.

VII. CONCLUSION

This paper presented a systematic approach developed for constructing reformulated radial basis function neural networks suitable for gradient descent learning. This approach reduces the development of admissible radial basis function

TABLE III
AVERAGE NUMBER N OF ADAPTATION CYCLES REQUIRED FOR TRAINING VARIOUS RADIAL BASIS FUNCTION NEURAL NETWORKS WITH c RADIAL BASIS FUNCTIONS AND FFNNS WITH n_h HIDDEN UNITS TO CLASSIFY THE DERMATOLOGY DATA IN TEN TRIALS, AND PERCENTAGE OF CLASSIFICATION ERRORS PRODUCED ON AVERAGE ON THE TRAINING SET (E_{tr}) AND THE TESTING SET (E_{te}). THE NUMBERS IN PARENTHESES REPRESENT THE STANDARD DEVIATION

Conventional RBF NNs				
c	2	4	6	8
N	606.6	689.9	762.4	738.6
E_{tr} (σ_{tr})	41.20 (4.75)	20.00 (5.72)	12.09 (3.34)	8.61 (4.85)
E_{te} (σ_{te})	43.05 (8.90)	23.10 (5.41)	15.95 (4.91)	13.85 (5.16)
Reformulated RBF NNs				
c	2	4	6	8
N	2942.1	6720	14282.5	9077.7
E_{tr} (σ_{tr})	32.09 (7.84)	15.76 (6.74)	5.76 (6.82)	2.59 (4.33)
E_{te} (σ_{te})	36.65 (8.14)	22.10 (5.06)	14.00 (5.73)	11.70 (3.98)
Cosine RBF NNs				
c	2	4	6	8
N	3334.5	2726.4	2135.4	1860.2
E_{tr} (σ_{tr})	18.73 (2.92)	5.25 (6.10)	0.00 (0.00)	0.06 (0.19)
E_{te} (σ_{te})	24.20 (2.37)	11.70 (6.59)	8.10 (1.37)	9.00 (0.55)
Feedforward NNs				
n_h	2	4	6	8
N	936.2	5639.5	853.9	7230.1
E_{tr} (σ_{tr})	5.63 (0.72)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
E_{te} (σ_{te})	19.35 (1.00)	8.35 (1.00)	8.00 (0.50)	7.75 (0.56)

models to the selection of admissible generator functions. The criteria proposed in this paper for selecting generator functions indicated that linear generator functions provide an attractive alternative to exponential generator functions when reformulated radial basis function neural networks are trained by gradient descent. Given that exponential generator functions lead to Gaussian radial basis functions, the comparison of linear and exponential generator functions indicated that Gaussian radial basis functions are not the only, and perhaps not the best, choice for constructing radial basis function neural models.

This paper also introduced a new family of reformulated radial basis function neural networks constructed by cosine radial basis functions and investigated their training by gradient descent. The experiments evaluated and benchmarked cosine radial basis function neural networks on a variety of datasets, which were selected to differ considerably in terms of their dimensionality and structure. The outcome of this experimental study reinforced some widely popular beliefs regarding neural-network models. As an example, the experiments indicated that FFNNs achieved considerably higher accuracy than conventional radial basis function neural networks, which were trained fast but lagged all other models when tested on all datasets used in this study. FFNNs outperformed reformulated radial basis function neural networks when tested on two of the datasets, namely, the Pima and Dermatology data. On the other hand, reformulated radial basis function and cosine radial basis function neural networks outperformed FFNNs when tested on the Glass data. It is also remarkable that the proposed cosine radial basis function neural networks were strong competitors to the best performing model tested on all datasets employed in this study. In many cases, the performance differences between the best performing model and cosine radial basis function neural networks were not significant. As an example, FFNNs classified incorrectly 7.75% of the feature vectors from the Dermatology data compared with 8.10% of the same feature vectors classified incorrectly by the best cosine radial basis function neural network. The experiments indicated that the performance of cosine radial basis function neural networks remained satisfactory when the dimensionality of the feature space increased. Moreover, the consistent performance of cosine radial basis function neural networks when they were tested on a variety of structurally different datasets revealed the versatility of the proposed radial basis function model and the associated learning procedure. Another interesting observation is that the performance differences between cosine radial basis function neural networks and the other two radial basis function models became increasingly profound as the number of radial basis functions decreased. This verifies the results of the analysis and indicates that the use of the proposed cosine radial basis functions allows the implementation of a given input-output mapping by an radial basis function model of smaller size. In fact, the results of the sensitivity analysis presented in this paper can be used to develop procedures for determining the initial locations of cosine radial basis functions and the number of cosine radial basis functions required to implement a desired input-output mapping. This can be done by establishing criteria for eliminating unnecessary radial basis functions and/or creating necessary radial basis functions. As an example, the criteria used for the elimination and/or creation of cosine radial basis functions can be established by requiring that all input vectors from the training set belong to the active region of at least one of the cosine radial basis functions.

The versatility of cosine radial basis function neural networks was also revealed by their evaluation on applications other than pattern classification. These applications provided the basis for evaluating the ability of cosine radial basis function neural networks to perform function approximation. Such an example is a recent study, which evaluated cosine radial basis function neural networks and FFNNs on electric power load forecasting [13]. This was attempted by training FFNNs and cosine radial basis function neural networks to predict future power demand based on past power load data and weather conditions. This comparison indicated that both neural-network models exhibit comparable performance when tested on the training data but cosine radial basis function neural networks generalize better since they outperform considerably FFNNs on the testing data.

REFERENCES

[1] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Oxford Univ. Press, 1995.
[2] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Systems*, vol. 2, pp. 321–355, 1988.
[3] I. Cha and S. A. Kassam, "Interference cancellation using radial basis function networks," *Signal Processing*, vol. 47, pp. 247–268, 1995.
[4] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. Neural Networks*, vol. 2, pp. 302–309, Mar. 1991.
[5] T. Chen and H. Chen, "Approximation capability to functions of several variables, nonlinear functionals, and operators by radial basis function neural networks," *IEEE Trans. Neural Networks*, vol. 6, pp. 904–910, July 1995.

[6] V. Cherkassky and F. Mulier, *Learning From Data: Concepts, Theory, and Methods*. New York: Wiley, 1998.
[7] D. Heinke and F. H. Hamker, "Comparing neural networks: A benchmark on growing neural gas, growing cell structures, and fuzzy ARTMAP," *IEEE Trans. Neural Networks*, vol. 9, pp. 1279–1291, Nov. 1998.
[8] N. B. Karayiannis, "Gradient descent learning of radial basis neural networks," in *Proc. 1997 IEEE Int. Conf. Neural Networks*, vol. 3, Houston, TX, June 9–12, 1997, pp. 1815–1820.
[9] ——, "Learning algorithms for reformulated radial basis neural networks," in *Proc. 1998 Int. Joint Conf. Neural Networks*, Anchorage, AK, 1998, pp. 2230–2235.
[10] ——, "Reformulated radial basis neural networks trained by gradient descent," *IEEE Trans. Neural Networks*, vol. 10, pp. 657–671, May 1999.
[11] ——, "An axiomatic approach to soft learning vector quantization and clustering," *IEEE Trans. Neural Networks*, vol. 10, pp. 1153–1165, Sept. 1999.
[12] ——, "New developments in the theory and training of reformulated radial basis neural networks," in *Proc. 2000 Int. Joint Conf. Neural Networks*, vol. 3, Como, Italy, July 24–27, 2000, pp. 614–619.
[13] N. B. Karayiannis, M. Balasubramanian, and H. Malki, "Evaluation of cosine radial basis function neural networks on electric power load forecasting," presented at the *2003 Int. Joint Conf. Neural Networks*, Portland, OR, July 20–24, 2003.
[14] N. B. Karayiannis and S. Behnke, "New radial basis neural networks and their application in a large-scale handwritten digit recognition problem," in *Recent Advances in Artificial Neural Networks: Design and Applications*, L. C. Jain and A. M. Fanelli, Eds. Boca Raton, FL: CRC, 2000, pp. 39–94.
[15] N. B. Karayiannis and W. Mi, "Growing radial basis neural networks: Merging supervised and unsupervised learning with network growth techniques," *IEEE Trans. Neural Networks*, vol. 8, pp. 1492–1506, Nov. 1997.
[16] C. A. Micchelli, "Interpolation of scattered data: Distance matrices and conditionally positive definite functions," *Constructive Approximation*, vol. 2, pp. 11–22, 1986.
[17] J. E. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Comput.*, vol. 1, pp. 281–294, 1989.
[18] T. Poggio and F. Girosi, "Regularization algorithms for learning that are equivalent to multilayer networks," *Science*, vol. 247, pp. 978–982, 1990.
[19] B. A. Whitehead and T. D. Choate, "Evolving space-filling curves to distribute radial basis functions over an input space," *IEEE Trans. Neural Networks*, vol. 5, pp. 15–23, Jan. 1994.

Nicolaos B. Karayiannis (S'85–M'86–SM'01) was born in Greece on January 1, 1960. He received the Diploma degree in electrical engineering from the National Technical University of Athens, Athens, Greece, in 1983, and the M.A.Sc. and Ph.D. degrees in electrical engineering from the University of Toronto, Toronto, ON, Canada, in 1987 and 1991, respectively.

From 1984 to 1991, he worked as a Research and Teaching Assistant at the University of Toronto. From 1983 to 1984, he was a Research and Teaching Assistant at the Nuclear Research Center "Democritos," Athens, Greece, where he was engaged in research on multidimensional signal processing. He is currently an Associate Professor in the Department of Electrical and Computer Engineering, University of Houston, Houston, TX. He has published more than 100 papers, including 43 in technical journals, and is the coauthor of the book *Artificial Neural Networks: Learning Algorithms, Performance Evaluation, and Applications* (Boston, MA: Kluwer, 1993).

Dr. Karayiannis is a member of the International Neural Network Society (INNS) and the Technicl Chamber of Greece. He is the recipient of the W. T. Kittinger Outstanding Teacher Award (1994) and the University of Houston El Paso Energy Foundation Faculty Acheivement Award (2000). He is also a corecipient of a Theoretical Development Award for a paper presented at the Artificial Neural Networks in Engineering '94 Conference. He is an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS and the IEEE TRANSACTIONS ON FUZZY SYSTEMS. He also served as the General Chair of the 1997 International Conference on Neural Networks (ICNN'97), held in Houston, TX, on June 9–12, 1997. His current research interests include supervised and unisupervised learning, learning vector quantization and neuro-fuzzy systems, biomedical imaging and video, networking, and wireless communications.

Mary M. Randolph-Gips received the B.S. degree in electrical engineering and the B.S. degree in engineering phyics from the University of Kansas, Lawrence, in 1990. She received the M.S. degree in electrooptics from the University of Houston at Clear Lake, Houston, TX, in 1995. She received the Ph.D. degree in electrical and computer engineering from the University of Houston in 2002.

From 1990 to 1997, she worked as a Space Shuttle Flight Controller in Payload Operations for the United Space Alliance Corporation, Houston, TX. She is a Research Assistant Professor at the University of Houston at Clear Lake, where she is enagaged in research on in intelligent control systems for hypersonic motors. Her research interests include neural networks, data mining, and fuzzy control.