ON RADIAL BASIS FUNCTION NEURAL NETWORKS

L.N.M.Tawfiq [*] **Q.H. *Eqhaar*** [**]
College of Education (Ibn Al-Haitham) College of Science
University of Baghdad University of Al-Qadisiya

1. ABSTRACT

Artificial Neural Networks (ANN's) are largely used in applications involving classification or functions approximation. It has been proved that several classes of ANN such as Multilayer Radial-Basis Function Networks (RBFN) are universal function approximators . Therefore, they are widely used for function approximation .

In this paper ,we examine the similarities and differences between RBFNNs compare the performance of learning with each representation applied to the interpolation problem. Nonetheless, this paper should help the reader to understand which basis function and which efficient method should be employed for particular reconstruction problem. It should also encourage the reader to consult the literature pointed out in the bibliography for further studying.

2. INTRODUCTION

A radial basis function network is a neural network approached by viewing the design as a curve-fitting (approximation) problem in a high dimensional space. Learning is equivalent to finding a multidimensional function that provides a best fit to the training data, with the criterion for “best fit” being measured in some statistical sense .Correspondingly, regularization is equivalent to the use of this multidimensional surface to interpolate the test data. This viewpoint is the real motivation behind the RBF method in the sense that it draws upon research work on traditional strict interpolations in a multidimensional space. In a neural network, the hidden units form a set of "functions" that compose a random "basis" for the input patterns (vectors). These functions are called radial basis functions, [1].

Radial basis functions were first introduced by Powell to solve the real multivariate interpolation problem . This problem is currently one of the principal fields of research in numerical analysis. In the field of neural networks, radial basis functions were first used by Broomhead and Lowe . Other major contributions to the theory, design, and applications of RBFNs can be found in papers by Moody and Darken, Renals, and Poggio and Girosi . The paper by Poggio and Girosi[2] explains the use of regularization theory applied to this class of neural networks as a method for improved generalization to new data .

The design of a RBFN in its most basic form consists of three separate layers. The input layer is the set of source nodes . The second layer is a hidden layer of high dimension. The output layer gives the response of the network to the activation patterns applied to the input layer. The transformation from the input space to the hidden-unit space is nonlinear. On the other hand, the transformation from the hidden space to the output space is linear [3].

3. RADIAL FUNCTIONS [4],[5]

Let X be a normed linear space .A function $f : X \to R$ is said to be radial if there exists a function $h : R^+ \to R$ such that $f(x) = h(\|x\|)$ for all $x \in X$.

A radial basis function is any translate of f; that is a function of the form $g(x) = f(x-\theta) = h(\|x - \theta\|)$, where θ is any prescribed point of X. In other word , Radial functions are a special class of functions show the characteristic feature that their response decreases or increases monotonically with distance from a central point .

4. INTERPOLATION PROBLEM

This section talks about the interpolation problem let us consider a feed forward network with an input layer, a single hidden layer, and an output layer having a single unit. The network can be designed to perform a nonlinear mapping from the input space to the hidden space, and a linear mapping from the hidden space to the output space. The network represents a map from p-dimensional input space to the single dimensional output space, expressed as $\mathbf{F} : R^p \rightarrow R$.

The theory of multivariable interpolation in high-dimensional space has a long history starting with Davis . The interpolation problem, in its strict sense can be stated as follows:

Given set of N different points$\{\mathbf{x}_i \in R^p \mid i = 1, 2,...., N\}$and a corresponding set of N real numbers$\{d_i \in R \mid i =1,2,....,N\}$find function $\mathbf{F}:R^P \rightarrow R$ that satisfies the interpolation condition:

$$\mathbf{F}(x_i) = d_i \ , \ i = 1,2,...., N \qquad \text{. (1)}$$

The interpolating surface (i.e. function **F**) has to pass through all the training data points . The radial basis function technique consists of choosing a function that has the following form given by Powell ,[6].

$$\mathbf{F}(x) = \sum_{i=1}^{N} w_i \, \varphi(\|x- x_i\|) \qquad \text{. . . . .(2)}$$

where $\{\varphi(\|x- x_i\|) \mid i = 1,2,\ldots N\}$ is a set of N random (usually nonlinear) functions, known as radial basis functions, and || . || represents a norm that is generally Euclidean. The known data points $x_i \in R^p$, i =1,2,…N are the centers of radial basis functions, [7].

If the interpolation conditions equation (1) is inserted in (2), the following set of simultaneous linear equations can be obtained for the unknown coefficients (weights) of the expansion $\{w_i\}$:

$$\begin{bmatrix} \varphi_{11}\varphi_{12}\cdots\varphi_{1N} \\ \varphi_{21}\varphi_{22}\cdots\varphi_{2N} \\ \vdots \quad \vdots \\ \varphi_{N1}\varphi_{N2}\cdots\varphi_{NN} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix} \qquad \text{. . . . .(3)}$$

Where

$\varphi_{ji} = \varphi(\|x_j - x_i\|)$ j,i= 1,2,…,N . . . . (4)

Let $\mathbf{d} = \lfloor d_1, d_2, \cdots, d_N \rfloor$ . . . . (5)

$\mathbf{w} = \lfloor w_1, w_2, \cdots, w_N \rfloor$ . . . . (6)

The vectors **d** and **w** represent the desired response vector and linear weight vector respectively. Let Φ denote an NxN matrix with elements φ_{ji} :

Φ={φ_{ji} \ j , i =1,2,…,N} . . . . (7)

The matrix Φ is called the interpolation matrix. Equation (3) can be written in the compact form: Φ **w** = **d** . . . . (8)

Light gives a remarkable property for a class of radial basis functions which obtains a positive definite interpolation matrix Φ . Because a positive definite matrix has always an inverse, this specific class of radial basis functions will always solve the Interpolation problem.

Powell declares that theoretical investigation and practical results show that the type of nonlinearity φ(·) is not vital to the performance of RBFNs . Considering Light's Theorem, it is noted that if the data points are all distinct, the interpolation matrix Φ is positive definite, and the weight vector **w** can be formed as follows:

$\mathbf{w} = \Phi^{-1}\,\mathbf{d}$ . . . . (9)

Even though in theory a solution to the strict interpolation problem exists, in practice equation (8) cannot be solved when the matrix Φ is arbitrarily close to singular. Regularization theory can solve this problem by perturbating the matrix Φ to Φ + λI, as described in Section 6. This problem leads us to examine the solving of an ill-posed hypersurface reconstruction problem because this inverse problem can become an ill-posed problem. The next section talks about how an ill-posed problem can be solved.

5. SUPERVISED LEARNING AS AN ILL-POSED HYPERSURFACE RECONSTRUCTION PROBLEM

The strict interpolation procedure as defined above may not be a good method for the training of RBF networks for certain classes of tasks because of poor generalization to new data for the following reason. When the number of data points in the training set is much larger than the number of degrees of freedom of the underlying physical process, and the network cannot have radial basis functions greater than the number of data points, the problem is over determined.

The design of a neural network trained to have an output pattern when presented with an input pattern is equivalent to learning a hypersurface (i.e. multidimensional mapping) that defines the output in terms of the input. That is, learning is considered as a hypersurface reconstruction problem, given a set of data that may be sparse. Therefore, the hypersurface reconstruction or approximation problem belongs to a “generic” class of problems called inverse problems .

An inverse problem can be well-posed or ill-posed. The term “well-posed” has been used in applied mathematics since the time of Hadamard in the early 1900s. Haykin gives an explanation about an inverse problem as follows:

Assume that we have a domain X and a range Y taken to be metric spaces, and that are related by a fixed but unknown mapping **F**. The problem of reconstructing the mapping **F** is said to be well posed if three conditions are satisfied:

1. **Existence**. For every input vector $x \in X$, there exist an output $y = \mathbf{F}(x)$, where $y \in Y$,
2. **Uniqueness.** For any pair of input vectors $x, t \in X$. We have $\mathbf{F}(x) = \mathbf{F}(t)$, if and only if, $x = t$
3. **Continuity**. The mapping is continuous, that is, for any $\varepsilon>0$, there exists $\delta = \delta(\varepsilon)$ such that the condition $\rho_x(x, t) < \delta$ implies that $\rho_Y(F(x), F(t)) < \varepsilon$, where $\rho(\cdot,\cdot)$ is the symbol for distance between the two arguments in their respective spaces.

If these conditions are not satisfied, the inverse problem is said to be ill-posed.

Haykin supports reasons why learning is an ill-posed inverse problem when learning is viewed as a hypersurface reconstruction problem. First, there is insufficient information in the training data in order to reconstruct the input-output mapping uniquely, thus the uniqueness criterion is not satisfied. Second, the existence of noise and imprecision in the input data adds uncertainty to the reconstructed input-output mapping. Particularly, if the noise level in the input is too high, the neural network may produce an output outside of the range **Y** for a specified input x in the domain **X**; hence the continuity criterion is not satisfied .

Poggio and Girosi state that some form of prior information about the input-output mapping is necessary to make the learning problem well-posed so that generalization to the new data can be accomplished. In other words, the process responsible for generation of input-output examples used to train a neural network must show redundancy in an information-theoretic sense to establish the prior information about the input-output mapping. This necessity is, indeed, satisfied by the physical processes such as speech, pictures, radar, and sonar signals in practice. These processes are all redundant by their nature. Moreover, the generator of the data is generally smooth. As long as the data generation is smooth, small changes in the input can cause large changes in the output and can still be approximated successfully. Haykin notes that the smoothness of data generation is a fundamental form of functional redundancy .

As was said in Section 4, the interpolation problem can not be solved when the matrix Φ is arbitrarily close to singular or ill-posed. The next section gives a solution to this problem using regularization theory.

6. REGULARIZATION THEORY

Regularization Theory was first introduced by Tikhonov in (1963) . The fundamental idea of regularization is to stabilize the solution in terms of some auxiliary nonnegative functional that embeds prior information, e.g., smoothness constraints on the input-output mapping, and make an ill-posed problem into a well-posed one .

Let the set of input-output data available for approximation be described by

Input signal : $x_i \in R^p$, $i = 1, 2, ..., N$

Desired signal: $d_i \in R$, $i = 1, 2, ..., N$

The dimensionality of the output is chosen as one. This choice does not limit the general applicability of the regularization theory in any way. The approximation function is denoted by **F**(x). The weight factor **w** of the network is omitted from the argument of the function **F** for convenience of presentation.

According to Tikhonov's regularization theory, the function **F** is obtained by minimizing a cost functional ξ(F) that maps functions to the real line. Haykin expresses the cost functional using two terms of regularization as follows:

$$\xi(\mathbf{F}) = \xi_s(\mathbf{F}) + \lambda\xi_c(\mathbf{F}) \qquad \text{. (10)}$$

where ξ_s(**F**) is standard error term that measures the standard error (distance) between the desired response d_i and the actual response y_i training samples i = 1, 2, .., N. The term ξ_c(**F**) is regularization term that depends on the geometric properties of the approximation function **F**(x). The symbol λ is a positive real number called regularization parameter. The main aim of the regularization is to minimize the cost functional ξ(**F**). The cost functional can be written in terms of the desired response d_i, the actual response y_i, and the regularization parameter λ as follows:

$$\xi(\mathbf{F}) = \frac{1}{2}\sum_{i=1}^{N}\left[d_i - \mathbf{F}(x_i)\right]^2 + \frac{1}{2}\lambda\left\|P\mathbf{F}\right\|^2 \qquad \text{. . . . .(11)}$$

where P is a linear (pseudo) differential operator that contains the prior information about the form of the solution. Haykin refers to P as a stabilizer in the sense that stabilizes the solution **F** making it smooth and therefore continuous.

The regularization parameter, λ, is considered as indicator of the sufficiency of the given data set as examples that specify the solution **F**(x). If λ → 0, the problem is unconstrained and the solution **F**(x) can be completely determined from the examples.

On the other hand, if, λ→ ∞, the priori smoothness constraint is sufficient to specify the solution **F**(x); that is, the examples are unreliable. In practice, λ is assigned a value somewhere between 0 and ∞, so that both the sample data and the priori information contribute to the solution **F**(x). Therefore, the regularizing term ξ_c (**F**) represents a model complexity penalty function, the influence of which on the final solution is controlled by the regularization parameter λ .

This section gave broad explanation of the cost functional ξ(**F**) and its parameters. The next section gives detailed mathematical review of the solution of the regularization problem minimizing the cost functional ξ(**F**).

6.1. Solution to the Regularization Problem

The principle of regularization is to find the function **F**(x) that minimizes the cost functional ξ(**F**), defined by equation (10) [8]. To manage the minimization of the cost functional ξ(**F**), an evaluation of the differential of ξ(**F**) is necessary. The Frechet differential can be employed to do the minimization. The Frechet differential has the following form:

$$d\xi(F,h)=\left[\frac{d}{d\beta}\xi(F+\beta h)\right]_{\beta=0} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (12)$$

where h(x) is a fixed function of the vector x, and β is a multi index. A multi index $\beta=(\beta_1,\beta_2, ..., \beta_n)$ of order $|\beta|=\sum_{i=1}^{n}\beta_i$ is a set of whole numbers used to abrreviate the following notations :

1. $x=x_1^{\beta_1}x_2^{\beta_2}\ldots x_n^{\beta_n}$ for $x\in R^n$

2. $x=\dfrac{\partial^{|\beta|}f}{\partial x_1^{\beta_1}\partial x_2^{\beta_2}\ldots\partial x_n^{\beta_n}}$ for $f:R^n\rightarrow R$

A necessary condition for the function **F**(x) to be a relative extremum of the functional ξ(**F**) is that the Frechet differential $d\xi$(**F**,h) be zero at **F**(x) for all $h\in$ H, as expressed by

$$d\xi(\mathbf{F},h) = d\xi_s(\mathbf{F},h) + \lambda d\xi_c(\mathbf{F},h) = 0 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (13)$$

where $d\xi_s(\mathbf{F},h)$ and $d\xi_c(\mathbf{F},h)$ are the Frechet differentials of the functionals $\xi_s(\mathbf{F})$ and $\xi_c(\mathbf{F})$, respectively. After the evaluation of the Frechet differential, the standard error term $\xi_s(\mathbf{F},h)$ is expressed as follows:

$$d\xi_S(F,h) = \left[\frac{d}{d\beta}\xi_s(F+\beta h)\right]_{\beta=0}$$

$$= -\sum_{i=1}^{N}\left[d_i - F(x_i)\right]h(x_i) \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (14)$$

Similarly, the regularizing term $\xi_c(\mathbf{F})$ can be expressed by the following equation:

$$d\xi_c(F,h) = \frac{d}{d\beta}\xi_c(F+\beta h)\Big|_{\beta=0}$$

$$= (Ph, P\mathbf{F})_H \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (15)$$

where H represents Hilbert space and the symbol $(\cdot,\cdot)_{\mathbf{H}}$ represents the inner product in H space. Hilbert space is a normed vector space of functions. These functions are rapidly decreasing, infinitely continuously differentiable functions. Considering the definition of an adjoint differential operator, equation (15) can be written as:

$$d\xi_c(\mathbf{F},h) = (h, P^{*}P\mathbf{F})_{\mathbf{H}} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (16)$$

where P* is the adjoint of the differential operator P.

Substituting the Frechet differentials of equation (14), and (16), in the equation (13), Haykin states that the Frechet differential $d\xi(\mathbf{F},h)$ is zero for every h(x) in **H** space if and only if the following condition is satisfied:

$$P^{*}P\mathbf{F} - \frac{1}{\lambda}\sum_{i=1}^{N}(d_i - \mathbf{F})\delta_{x_i} = 0$$

or, equivalently, $P*P\mathbf{F}(x) = \frac{1}{\lambda}\sum_{i=1}^{N}\left[d_i - \mathbf{F}(x_i)\right]\delta(x - x_i)$ . . . .(17)

where $\delta(x - x_i)$ or (δ_{x_i}) is a delta function located at x = x_i .

Poggio and Grosi refer to equation (17) as the Euler-Lagrange Equation for the cost functional $\xi(\mathbf{F})$ expressed in equation (11) . They also declare that the equation (17) symbolizes a partial pseudo differential equation in **F**. The solution of this equation can be obtained by applying the integral transformation of the right hand side of the equation with a kernel given by the influence function or Green's Function for the self-adjoint differential operator P*P. The role of the Green's function in a linear differential equation is the same as the role that an inverse matrix plays in a matrix equation.

Let $G(x;x_i)$ be a Green's function centered at x_i. A Green's function $G(x;x_i)$ is any function that satisfies the partial differential equation

$P*PG(x;x_i) = 0$

everywhere other than at the point $x = x_i$, where the Green's function has a singularity. Haykin gives the solution **F**(x) for the differential equation (17) after some mathematical steps resulting with the following equation:

$$\mathbf{F}(x) = \frac{1}{\lambda}\sum_{i=1}^{N}\left[d_i - F(x_i)\right]G(x;x_i) \quad \text{. (18)}$$

Equation (18) shows that the minimizing solution **F**(x) to the regularization problem is a linear superposition of N Green's functions. In this equation, the x_i symbolize the centers of the expansion , and the weights

$[[d_i - \mathbf{F}(x_i)] / \lambda]$ symbolize the coefficients of the expansion. Haykin declares that the solution of the regularization problem lies in an N-dimensional subspace of the space of smooth functions, and the set of Green's functions $G(x ; x_i)$ centered at x_i , i =1, 2,…, N, builds a basis for this subspace.

Since we have the solution **F**(x) to the regularization problem, our next step is to determine the unknown coefficients in the equation (18). Let us denote:

$$w_i = \frac{1}{\lambda}\left[d_i - \mathbf{F}(x_i)\right],\ i=1,2,\ldots,N \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (19)$$

The minimizing solution can be symbolized by the following equation

$$\mathbf{F}(x) = \sum_{i=1}^{N} w_i G(x;x_i) \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (20)$$

After evaluation of the equation (20) at x_j, j =1, 2,.., N, the equation can be expanded to

$$\mathbf{F}(x_j) = \sum_{i=1}^{N} w_i G(x_j;x_i),\ j=1,2,\ldots,N \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (21)$$

The definitions needed to be introduced can be given as follows;

$$\mathbf{F}=[\mathbf{F}(X_1),\mathbf{F}(X_2),\ldots,\mathbf{F}(X_N)]^T \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

(22)

$$\mathbf{d}=[d_1,d_2,\ldots.,d_N]^T \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

(23)

and ,

$$\mathbf{G} = \begin{bmatrix} G(x_1;x_1)G(x_1;x_2)\cdots G(x_1;x_N) \\ G(x_2;x_1)G(x_2;x_2)\cdots G(x_2;x_N) \\ \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ G(x_N;x_1)G(x_N;x_2)\cdots G(x_N;x_N) \end{bmatrix} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (24)$$

$$\mathbf{w} = [w_1, w_2, \ldots w_N]^T \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (25)$$

Now, the equation (19) and equation (21) can be rewritten in matrix form as follows, respectively:

$$\mathbf{w} = \frac{1}{\lambda}(d - F) \quad \text{. (26)}$$

$$\mathbf{F} = \mathbf{G}\mathbf{w} \quad \text{. (27)}$$

When F is eliminated between equations (26) and (27), the following equation can be obtained: $(\mathbf{G} + \lambda\mathbf{I})\mathbf{w} = \mathbf{d}$ . . . . (28)

The matrix **G** is called the Green's Matrix. Since the combined operator P*P in equation (17) is self-adjoint, the associated Green's function $\mathbf{G}(x;x_i)$ is a symmetric function, as shown by Courant and Hilbert.

$$\mathbf{G}(x_i;x_j) = \mathbf{G}(x_j;x_i) \quad \text{for all i and j} \quad \text{. (29)}$$

Similarly, the Green's matrix G defined in equation (24) is a symmetric matrix;

$$\mathbf{G}^T = \mathbf{G} \quad \text{. (30)}$$

After revisiting Light's theorem, defined in Section 4 in the context of the interpolation matrix Φ, Haykin notes that Green's matrix, **G** has a role in regularization theory similar to the role that of Φ in RBF interpolation theory. Both **G** and Φ are NxN symmetric matrices. Haykin also states that the matrix **G** is positive definite for certain classes of Green's functions if the data points x_1, $x_2,\ldots,x_N$ are distinct.

Multiquadrics and Gaussian functions are the classes of Green's functions covered by Light's theorem. In practice, λ can be chosen sufficiently large enough to suffice so that (**G** + λ**I**) is positive definite, and thus, invertible. Poggio and Girosi give a unique solution of the linear system of equations (28) as follows:

$$\mathbf{w} = (\mathbf{G} + \lambda\mathbf{I})^{-1}\ \mathbf{d} \quad \text{. (31)}$$

Haykin concludes that the solution to the regularization problem is expressed by the equation

$$\mathbf{F}\,(\mathrm{x}) = \sum_{i=1}^{N} \mathrm{w}_i\, \mathbf{G}(\mathrm{x};\, \mathrm{x}_i) \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (32)$$

where $G(x_i;x_j)$ is the Green's function for the self adjoint differential operator P*P, and w_i is the ith element weight vector w. He also states that if the Green's functions in equation (32) are radial basis functions, the solution can be rewritten as follows:

$$\mathbf{F}(\mathrm{x}) = \sum_{i=1}^{N} \mathrm{w}_i \mathbf{G}(\|\mathrm{x}-\mathrm{x}_i\|) \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (33)$$

There is one more step left to reach the final solution of the regularization problem. We need to define the radial basis function in equation (33). The following functions can be employed in equation (33):

$$\mathrm{G}(\mathrm{x};\mathrm{x}_i) = \exp(-\frac{1}{2\sigma_i^2}\|\mathrm{x}-\mathrm{x}_i\|^2) \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (34)$$

where $\mathbf{G}(x;\, x_i)$ is a multivariate Gaussian function characterized by a mean vector x_i and common variance σ^2, except for a scaling factor that can be put in the weight w_i. Now we can present our final solution to the regularization problem as follows:

$$\mathbf{F}(\mathrm{x}) = \sum_{i=1}^{N} \mathrm{w}_i \exp(-\frac{1}{2\sigma^2}\|\mathrm{x}-\mathrm{x}_i\|^2) \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (35)$$

which contains a linear superposition of multivariate Gaussian basis functions with centers x_i (located at the data points) and widths σ_i. In section 6, we discussed regularization theory and we declared the solution of the regularization problem in terms of radial basis functions. Now, we need to define our network structure that supports the solution we found in this section. The next section talks about regularization networks that are built by the radial basis functions defined above.

Remark

1. If the regularization parameter λ approaches zero , the weight vector **w** converges to the pseudo inverse solution , which is the optimal solution of the overdetermined least-squares data fitting problem where $m_1 < N$

$$\mathbf{w} = G^{+}\mathbf{d} = (G^{T}G)^{-1}G^{T}\mathbf{d}$$

2. The value of the regularization parameter does not affect much the performance if $\lambda \geq 0.1$.
3. Increasing the number of centers (radial – basis functions) from 20 to 100 improves the performance by about 45%.

7. REGULARIZATION NETWORKS

A regularization network, introduced by Poggio and Girosi because it uses the solution to the regularization problem expressed in Section 6. The network has three layers. The first layer of the network consists of input nodes whose number is equal to the dimension p of the input vector x (i.e., the number of independent variables of the problem). The second layer is a hidden layer, made up of nonlinear units that are connected directly to all of the nodes in the input layer. There is one hidden unit for each data vector x_i , i = 1, 2,...., N, where N is the number of training samples. The activation function of the individual hidden units are described by the Green's functions. Correspondingly, **G**(x; x_i) represents the output of the ith hidden unit. The output layer has one single linear unit which is fully connected to the hidden layer. The term “linearity” is introduced because the output of the network is a linearly weighted sum of the outputs of the hidden units. The weights of the output layer are the unknown coefficients of the expression described in equation (31) in terms of the Green’s functions **G**(x; x_i) and the regularization parameter λ. Obviously, such a network structure can be readily extended to have any number of outputs desired [9].

The Green’s function **G**(x; x_i) is assumed to be positive definite for all i in the regularization network. Haykin states that if this condition is satisfied, which is true in the case where the Green’s functions **G**(x; x_i) have the form of Gaussian functions, then this network will produce an “optimal” interpolant solution in the sense that it minimizes the functional ξ(**F**). Furthermore, Poggio and Girosi give three properties of the regularization network from the viewpoint of approximation theory as follows:

1. The regularization network is a universal approximator in that it can approximate arbitrarily well any multivariate continuous function on a compact subset of R^p , given a sufficiently large number of hidden units.
2. Since the approximation scheme derived from regularization theory is linear in the unknown coefficients, it follows that the regularization network has the best approximation property. This means that given an unknown nonlinear function **F**, there always exists a choice of coefficients that approximates **F** better than all other possible choices.
3. The solution computed by the regularization network is optimal. Optimality here means that the regularization network minimizes a functional that measures how much the solution deviates from its true value as represented by training data.

A regularization network is prohibitively expensive to build in computational terms for large N because there is a one-to-one correspondence between the training input data x_i and Green's function **G**(x; x_i) for i = 1, 2,...., N. In particular, the inversion of NxN matrix is necessary to calculate the linear weights of the network (i.e., the coefficients of the expression in equation (32)). When N gets to large, the computational complexity will be very high. Moreover, the probability of ill conditioning is higher for larger matrices. Because of these reasons, we need a generalization of the solution in some sense. The next section introduces a generalization of the solution defining a new type of network structure which is called generalized radial basis function network.

8. PROPERTIES OF RBF

Over the past decades radial basis functions or, more generally, (conditionally) positive definite kernels have very successfully been used for reconstructing multivariate functions from scattered data. This success is mainly based upon the following facts:

(i) Radial basis functions can be used in any space dimension.

(ii) They work for arbitrarily scattered data, bearing no regularity at all.

(iii) They allow interpolants of arbitrary smoothness.

(iv) The interpolants have a simple structure, which makes RBFs in particular interesting to users outside mathematics.

However, these positive properties do not come for free. For example, building a smooth interpolant using a smooth basis function leads also to an ill-conditioned linear system that has to be solved. Moreover, since most basis functions are globally supported, a large number of interpolation points leads to an unacceptable complexity concerning both space and time.

For these reasons recent research concentrated on resolving these problems. Fast methods for evaluating and computing an RBF interpolant have been developed and thoroughly investigated. Smoothing techniques have been employed to regularize ill-conditioned systems and to smooth out measurement errors.

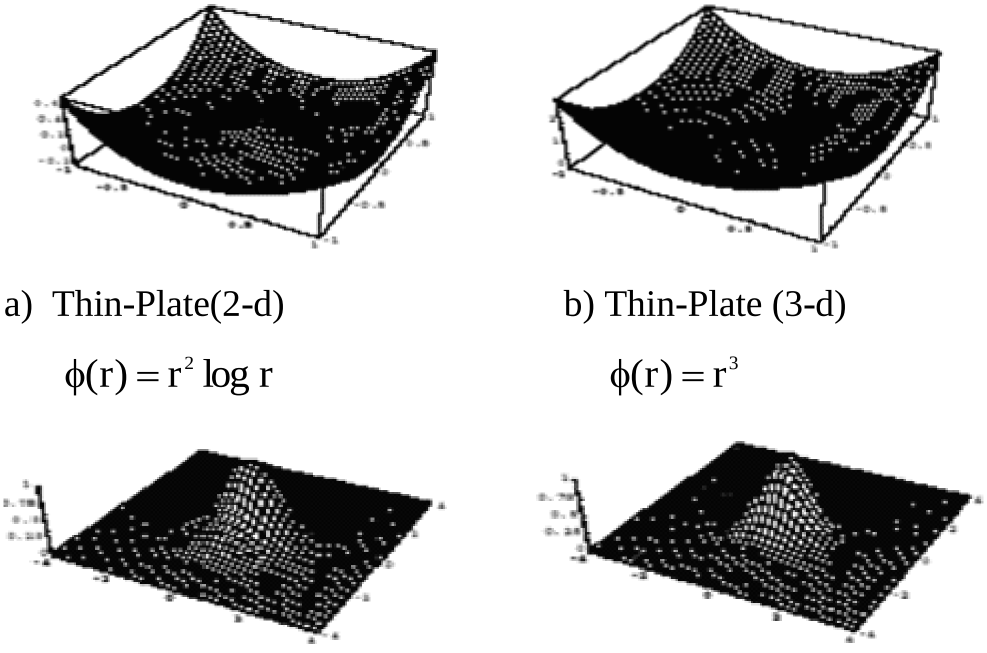
9. CHANGE OF BASIS

So far, we have learned that smoothing is an adequate choice in the situation of highly non-uniform data sets. It also helps in the case of quasi-uniform data sets and infinitely smooth basis functions, like Gaussians and (inverse) multiquadrics, since their associated interpolation matrices are already highly ill-conditioned in that particular situation for moderate separation distances. Unfortunately there exists no theoretical converage of error estimates in that situation, even if numerical test show promising results.

Our final task, for basis functions of finite smoothness, is to deal with the case of really dense data sets. TPS is piecewise smooth RBFs but G, MQ ,IMQ are infinitely smooth RBFs. We recall that G, IMQ(inverse multiquadrics) and W_2(wendland compactly supported .i.e. $\phi(r\ )= (1\text{-}r)^4 + (4r+1)$ are positive definite (PD), i.e. the corresponding collocation matrix A is positive definite for every choice of the (distinct) interpolation nodes, while TPS and MQ are conditionally positive definite (CPD). (note: Where A defined in linear system Ac=f (interpolation equations)and is symmetric matrix, usually termed collocation matrix of the RBF (see figure 1) several forms of ϕ □are used for RBF models.

Amongst these, the Gaussian is probably the most popular basis function because it has attractive mathematical properties of universal and best approximation and its hill-like shape is easy to control with the parameter σ.

Also Gaussian basis Functions are quasi-orthognal , the product of two basis functions, whose centers far away from each other with respect to their spreads , is almost zero.



a) Thin-Plate(2-d)

$\phi(r) = r^2 \log r$

b) Thin-Plate (3-d)

$\phi(r) = r^3$

c) Gaussian

$\phi(r) = e^{r^2/\sigma^2}$

d) Compactly Supported

$\phi(r) = (1-r)^4 + (4r+1)$

Figure (1) Comparison of different radial basis functions .

While the thin-plate spline embedding function does indeed minimize bending energy , it has the following drawbacks in computation and usefulness for user interaction :

1. $O(n^2)$ computation in required to build the system of equations .
2. $O(n^2)$ storage is required (for the nearly-full matrix) to represent the system.
3. $O(n^2)$ computation is required to solve the system of equations.
4. $O(n)$ computation is required per evaluation
5. Because every known point affects the result, a small change in even one constraint is felt throughout the entire resulting interpolated surface ,an undesirable property for shape modeling.

Figure (2) illustrates using both thin-plate and compactly-supported radial basis functions to compute embedding functions. The constraint points consist of 36 points in an ovoid shape with 36 normal (positive valued) constraints placed just inside (2.a). A thin-plate radial basis function produces a globally –smooth embedding function(2.b). A compactly –supported radial basis function produces an embedding function that does not have global smoothness but is as smooth as the thin –plate spline interpolation in a narrow band surrounding the shape both inside and out.

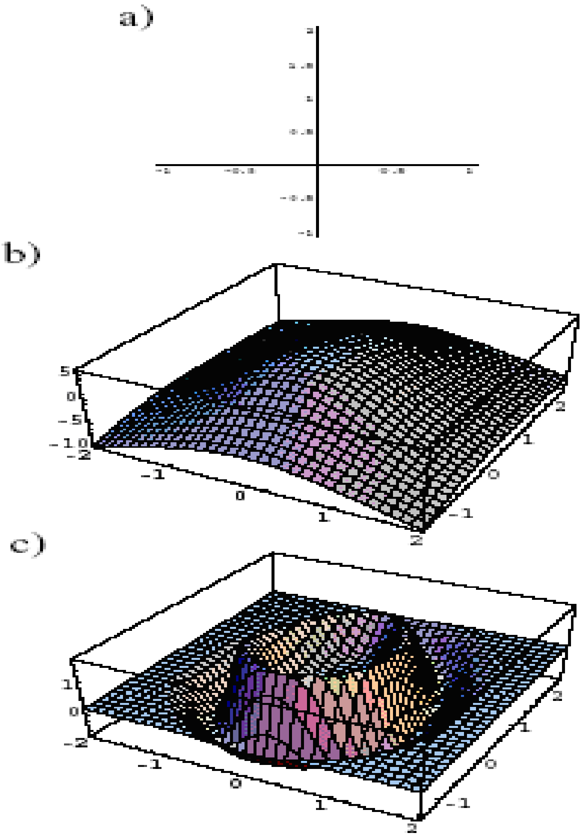


Figure (2). A simple 36-point ovoid (a)interpolated using thin-plate (b) and compactly –supported (c) radial basis functions.

The previous analyses have shown that the MultiQuadrics (MQ) and Thin Plate Spline (TPS) give the most accurate results for scattered data approximations . However, the accuracy of the MQ method depends on a shape parameter and as yet there is no mathematical theory about how to choose its optimal value . Hence, most applications of the MQ use experimental tuning parameters or expensive optimization techniques to evaluate the optimum shape parameter .While the TPS method gives good agreement without requiring such additional parameters and based on sound mathematical theory .

An mth order TPS is defined as $\phi\ (x, x_j) = \phi(r_j) = r_j^{2m}\ \log(r_j)$, $m = 1, 2,3, \ldots$ where $r_j = \|x- x_j\|$ is the Euclidean norm. Since ϕ is C^{2m-1} continuous, a higher-order TPS must be used, for higher-order partial differential operators. The advection-diffusion equation is of second-order, $m = 2$ is used to ensure at least C^2 continuity for **F**. Our numerical results confirm such observation and coincide with the result given in [7].

References

[1] G.P.Jaya Prakash and TRBstaff Representative,"**Use of Artificial Neural Networks in Geomechanical and Pavement System**", TRANSPORTATION RESEARCH CIRCULAR,Number E-co12 ,December (1999).

[2] T.Poggio and F.Girosi, "A Theory of Networks for Approximation and Learning ,"MASSACHUSETTS INSTITUTE OF TECHNOLOGY ARTIFICIAL INTELLIGENCE LABORATORY", A.I.Memo No.1140, C.B.I.P Paper No.31,July (1989).

[3] B.Fornberg,E.Larsson and G.Wright ,"**A New Class of Scillatory Radial Basis Functions**",NSF grants DMS-9810751(VIGRE) and DMS-030980,3.

[4] J.Zhang ,W.Baqai and A.Knoll,"**A comparative Study of B-spline Fuzzy Controller and RBFN**",In proceedings of the Fourth European Workshop on Fuzzy Decision Analysis and Recognition Technology ,Dortmund,(1999).

[5] W.A.LIGHT and E.W.CHENEY, "**Interpolation by Periodic Radial Basis Functions**", JOURNAL OF MATHEMATICAL ANALYSIS AND APPLICATIONS ,No.168 ,pp.111-130,(1992).

[6] M.J.D.Powell,"**The Theory of Radial Basis Function Approximation in 1990**", University of Cambridge ,Numerical Analysis Reports,DAMTP 1990/NA11,December (1990).

[7] I.Boztosun,A.Charafi,M.Zerroukat and K.Djidjeli, "**Thin-Plate Spline Radial Basis Function Scheme for Advection-Diffusion Problems**",Electonic Journal of Boundary Elements,Vol.BETEQ 2001,No.2,pp.267-282,(2002).

[8] T.Poggio and F.Girosi and M.Jone , "**Regularization Theory and Neural Networks Architecturess**", J.of Neural Comp.7 , pp.219-269,(1995).

[9] D. Simon ,"**Training Radial Basis Neural Networks with The Extended Kalman Filter**", Neurocomputing 000,pp.1-21.(col,fig.inil),(2001).

حول الشبكات العصبية الصناعية ذات دوال الاساس الشعاعية

أ.م.د.لمى ناجي محمد توفيق & قصي حاتم عكار الكفاري

المستخلص

الشبكات العصبية الصناعية لها تطبيقات في مجالات واسعة منها مجال التصنيف أو تقريب الدوال وثبت إن أنواع عديدة من الشبكات الصناعية العصبية ذات التغذية التقدمية متعددة الطبقات ذات دوال الأساس الصلبة ودوال الأساس الشعاعية أستخدمت في نطاق واسع في مجال تقريب الدوال بشكل عام.

في هذا البحث تم اختبار ودراسة التشابه والفروقات بين أنواع من الشبكات الصناعية ذات دوال الأساس الشعاعية من حيث الأداء والتعلم الخاصة بمسائل الاندراج .

كما إن هذا البحث يساعد على اختيار دوال الأساس المناسبة الكفوءة لمعالجة مسألة معينة .