On some extensions of Radial Basis Functions and their applications in Artificial Intelligence

Federico Girosi

Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139, USA

1 Introduction

In recent years approximation theory has found interesting applications in the fields of Artificial Intelligence and Computer Science. For instance, a problem that fits very naturally in the framework of approximation theory is the problem of learning to perform a particular task from a set of examples. The examples are sparse data points in a multidimensional space, and learning means to reconstruct a surface that fits the data. From this perspective, the so popular approach of Neural Networks to this problem (Rumelhart et al., 1986; Hertz et al., 1991) is nothing else than the implementation of a particular kind of nonlinear approximation scheme. However, despite the great popularity, very little is known about the properties of neural networks. For this reason we started considering the same class of problems, but in a more classical framework. In particular, since the problem of approximating a surface from sparse data points is ill-posed, regularization theory (Tikhonov, 1963; Tikhonov and Arsenin, 1977; Morozov, 1984; Bertero, 1986) seemed to be an ideal framework. Regularization theory leads naturally to the formulation of variational principle, from which it is possible to derive some well known approximation schemes (Wahba, 1990), such as the multivariate splines previously introduced by Duchon (1977) and Meinguet (1979) and the more general Radial Basis Functions technique (Powell, 1987; Franke, 1982; Micchelli, 1986; Kansa, 1990a,b; Madych and Nelson, 1991; Dyn, 1991). Due to the characteristics of the problems of machine learning, Radial Basis Functions seemed to be a very appropriate technique. Moreover, this method has a simple interpretation in terms of a "network" whose architecture is similar to the one of the multilayer perceptrons, and therefore retains all the advantages of this architecture, such as the high degree of parallelizability. Unfortunately, in many practical cases the Radial Basis Functions method cannot be applied in a straightforward manner, because does not take in account some features that are typical of problems in Artificial Intelligence. Goal of this paper is to review some of these aspects, show possible solutions in the framework of Radial Basis Functions, and point out some open problems.

In the next section we briefly define what we mean by learning, and show how natural is its embedding in the framework of approximation theory. In section 3 we review the classical variational approach to surface reconstruction, and in section 4 we show some extensions that are needed, in order to cope with practical problems. In section 5 we present two examples and in section 6 we draw some conclusions and show future directions of investigation. Other aspects and extensions of this approach to learning are discussed in (Poggio and Girosi, 1990).

2 Learning and Approximation Theory

The problem of learning is at the very core of the problem of intelligence. But learning is a very broad term and there are many different forms of learning that must be distinguished. In this paper we will consider a specific form of the problem of learning from examples. This is clearly only one small part of the larger problem of learning, but – we believe – an interesting place from where to start a rigorous theory and from where to develop useful tools.

2.1 Learning as an Approximation Problem

If we look at many of the problems that have been considered in the field of machine learning, we notice that in all the cases an instance of the problem is given by a set D of input-output pairs, $D \equiv \{(x_i, y_i) \in X \times Y\}_{i=1}^N$, belonging to some input and output space, X and Y (see fig. 1).

Figure 1 about here

We assume that there is some relationship between the input and the output, and consider the pairs (x_i, y_i) as *examples* of it. Therefore we assume that it exists a map

$$f : X \rightarrow Y$$

with the property:

$$f(x_i) = y_i \quad i = 1, \ldots, N \; .$$

Learning, or generalizing, means being able to estimate the function at points of its domain X where no data are available. From a mathematical point of view this means estimating the function f from the knowledge of a subset D ot its graph, that is from a set of sparse data. Therefore from this point of view the problem of learning is equivalent to the problem of surface reconstruction. Of course learning and generalization are possible only under the assumption that the world in which we live is – at the appropriate level of description – redundant. In terms of surface approximation it means that the surface to be reconstructed have to be *smooth*: small changes in some input determine a correspondingly small change in the output (it may be necessary in some cases to accept *piecewise smoothness*). Generalization is not possible if the mapping is completely random. For instance, any number of examples for the mapping represented by a telephone directory (people's names into telephone numbers) do not help in estimating the telephone number corresponding to a new name. Some examples are in order at this point.

1. *Learning to pronounce English words from their spelling*

   A well known neural network implementation has been claimed to solve the problem of converting strings of letters in strings of phonemes (Sejnowski and Rosenberg, 1987). The mapping to be learned was therefore $X \equiv$ {English words} $\rightarrow Y \equiv$ {phonemes}. The input string consisted of 7 letters, and the output was the phoneme corresponding to the letter in the middle of string.

Feeding English text in a window of 7 letters, a string of phonemes was produced and then processed by a digital speech synthesizer, therefore enabling the machine to read the text aloud. Because of the particular way of encoding letters and phonemes, the input consisted of a binary vector of 203 elements, and the output of a vector of 26 elements. Clearly in this case the smoothness assumption is often violated, since similar words do not always have similar pronunciations, and this is reflected in the fact that the percentage error on a test set of data was 78%, for continuous informal speech.

2. *Learning to recognize a 3D object from its 2D image*

   The input data set consists of $2D$ views of different objects in different poses. The output corresponding to a given input is 1 if the input is a $2D$ view of the object that has to be recognized, and 0 otherwise. A $2D$ view is a set of features of the $2D$ image of the object. In the simplest case a feature is the location, on the image plane, of some reference point, but may be any parameter associated to the image. For example, if we are interested in recognizing faces, common features are nose and mouse width, eyebrows thickness, pupil to eyebrows sepatation, pupil to nose verical distance, mouth height. In general the mapping to be learned is:

$$X \equiv \{\text{2D view}\} \quad \rightarrow Y \equiv \{0,\ 1\}\ .$$

   In (Poggio and Edelman, 1990) the problem of recognizing simple wire-frame objects was considered. The input was the 12-dimensional space of the locations on the image plane of 6 feature points. Good results were obtained applying least squares Radial Basis Functions (see section 4.1) with 80–100 examples. This indicates that the underlying mapping is smooth almost everywhere (in fact, under particular conditions, it may the characteristic function of a half-space).

   Good results have been also obtained, with real world images, on the similar problems of face and gender recognition, although in that case the map to be approximated may be much more complex (Brunelli and Poggio, 1991, 1991a).

3. *Learning Navigation tasks*

   An indoor robot is manually driven through a corridor, while its frontal camera records a sequence of frontal images. Can this sequence be used to train the robot to drive by itself without crashing into the walls by using the visual input and to generalize to slightly different corridors and different positions within them? The map to be approximated is

$$X \equiv \{\text{images }\} \quad \rightarrow Y \equiv \{\text{steering command }\}\ .$$

   D. Pomerlau (1989) has described a similar outdoor problem of driving the CMU Navlab and considered an image as described by all its pixel grey values. The corresponding approximation problem, solved with a neural network architecture, had therefore 900 variables, since images of 30×30 pixels have been used. A simpler possibility consists in coding an image by an appropriate set of features (as location or orientation of relevant edges of the image) (Aste and Caprile, 1991).

4. *Learning Motor Control*

   Consider a multiple-joint robot arm that has to be controlled. One needs to solve the inverse dynamics problem: compute from positions, velocities and accelerations of the joints the corresponding torques at the joints. The map to be learned is therefore the "inverse dynamics" map

   $$X \equiv \{\text{state space }\} \quad \rightarrow Y \equiv \{\text{torques }\} \ ,$$

   where the state space is the set of all admissible position, velocities and accelerations of the robot. For a two-joints arm the state space is six dimensional and there are two torques to be learned, one for each joint. Very good performances have been obtained in simulations run by Botros and Atkeson (1990) using the extensions of the Radial Basis Functions technique described in section (4.2).

5. *Learning to predict time series*

   Suppose we observe and collect data about the temporal evolution of a multidimensional dynamical system from some time in the past up to now. Given the state of the system at some time t in the future we would like to be able to predict its state at a successive time $t+T$. In practice we usually observe the temporal evolution of one of the variables of the system, say $x(t)$, and represent the state of the system as a vector of "delay variables" (Farmer and Sidorowich, 1987) [1]:

   $$\mathbf{x}(t) \equiv \{x(t), x(t-\tau), \ldots, x(t-(d+1)\tau)\}$$

   where τ is an appropriate delay time, and d is larger than the dimension of the attractor of the dynamical system. The map that has to be learned is therefore

   $$f_T \ : \ R^d \rightarrow R \ ,$$
   $$f_T(\mathbf{x}(t)) \rightarrow \mathbf{x}(t+T) \ .$$

   Some authors (Broomhead et al., 1990; Casdagli, 1989; Moody et al., 1989) have already successfully applied Radial Basis Functions techniques to time series prediction, with interesting results.

[1] It can shown (Takens, 1981) that this representation preserves the topological properties of the attractor. However this is not the only technique that can be used to represent the state of a dynamical system (Broomhead and King, 1987).

2.2 Learning from Examples and Radial Basis Functions

As we have seen, in many practical cases learning to perform some task is equivalent to recovering a function from a set of scattered data points. Many different techniques are currently available for surface approximation. However, the approximation problems arising from learning problems has some unusual features that make many of these techniques not very appropriate. Let us briefly see some of these features:

- *large number of dimensions.* This is the most striking characteristic of the problem of learning from examples. Usually a large set of number is needed in order to specify the input of the system. Consider for example all the problems related to vision, as object or character recognition: if all the pixels of the image are used the number of variables is enormous (of the order of 1000 for low resolution images, of 30×30 pixels). Clearly, not all the information carried in the pixels may be necessary, and several techniques have developed to extract, out of the thousands of pixels values, "small" sets of features, of the order of one hundred or less (16 features have been successfully used by Brunelli and Poggio (1991) for face and gender recognition). In speech recognition problems the number of dimensions is also easily of the order of hundreds (Lippmann, 1989). Time series prediction is a "simple" problem from this point of view. In fact the number of variables is of the order of the dimension of the attractor of the underlying dynamical system, that is typically smaller than 10;

- *relatively small number of data points.* In practical applications the number of available data points may vary, from few hundreds to several thousands, but usually never exceed 10^4. For such high dimensional problems, however, these number are small. The 30 dimensional hypercube, with 10^4 data points in it, is empty, cosidered that the number of its vertices is $2^{30} \approx 10^9$. The emptyness of these high dimensional space is a manifestation of the so called "curse of dimensionality", and sets the limits of any approximation technique;

- *noisy data.* In every practical application data are noisy. For this reason approximation, instead of pure interpolation, is of interest.

It is natural to ask if, besides computational problems, it is meaningful to approximate a function of 30 variables given 10^4 data points or less. The answer clearly depends on properties of the functions we want to approximate, as, for example, their degree of smoothness. However there is another factor to be taken in account, and it is the fact that often *only a low degree of accuracy is required.* In many cases, in fact, a relative error of $10^{-1} - 10^{-2}$ may be more than satisfactory, as long as we can be confident that is never excedeed. The extensive experimentation that has been carried over in the field of neural networks indicates that in practical cases these conditions are met, since good experimental results have been found in many cases.

Once established that approximation theory may be useful in some practical learning problems, a specific technique has to be chosen. Given the characteristic of the learning problems, Radial Basis Functions seems to be quite a natural choice. In fact, most of the other techniques, that may be very successful in 1 or 2 dimensions, run in problems when the number of dimensions increases. For example, in such large and empty spaces, techniques based on tensor products or triangulations do not seem appropriate. Moreover, since we can now take advantage of parallel machines, it is very convenient to deal with functions of the form

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i \phi_i(\mathbf{x}) \ ,$$

and Radial Basis Functions satisfy this requirement. As we will see in the next section Radial Basis Functions can be derived in a variational framework, in which noisy data are naturally taken in account.

However it will turn out that a straightforward application of Radial Basis Functions is not always possible because of the high computational complexity or because departure from radiality is needed. Nevertheless, approximations of the original Radial Basis Functions expansion can be devised ("least square Radial Basis Functions"), that still mantain the good features of Radial Basis Functions, and non radiality can be also taken in account, as it will be shown in section (4.2).

Of course, other techniques than Radial Basis Functions could be used. For example, Multilayer Perceptrons (MLP) are very common in the Artificial Intelligence community. In the simplest version a Multilayer Perceptron, that is a particular case of "neural network", is a nonlinear approximation scheme based on the following parametric representation:

$$f(\mathbf{x}) = \sum_{i=1}^{n} c_i \sigma(\mathbf{x} \cdot \mathbf{w}_i + \theta_i) \ , \tag{1}$$

where the parameters c_i, $\mathbf{w}_i$ and θ_i have to be found by minimizing the least square error on the data points. Not much is known on the properties of such an approximation scheme, except the fact that set of functions of the type (1) is dense in the space of continuous function on compact sets provided with the uniform norm (Cybenko, 1989; Funahashi, 1989). The huge literature in the field of neural networks shows that this approximation scheme performs well in many cases, but a solid analysis of its performances is still missing.

For these reasons we focused our attention on the Radial Basis Functions technique, being more well understood and susceptible of a theoretical analysis. The next section is therefore devoted to a review of the Radial Basis Functions method, and in particular to its variational formulation.

3 A Variational Approach to Surface Reconstruction

From the point of view of learning as approximation, the problem of learning a smooth mapping from examples is ill-posed (Hadamard, 1964; Tikhonov and Arsenin, 1977) in the sense that the information in the data is not sufficient to reconstruct uniquely the mapping in regions where data are not available. In addition, the data are usually noisy. Some *a priori* information about the mapping is needed in order to find a unique, physically meaningful, solution. Several techniques have been devised to embed a priori knowledge in the solution of ill-posed problems, and most of them have been unified in a very general theory, known as *regularization theory*, has been developed (Tikhonov, 1963; Tikhonov and Arsenin, 1977; Morozov, 1984; Bertero, 1986).

The a priori information considered by regularization theory can be of several kinds. The most common concern smoothness properties of the solution, but also localization properties and upper/lower bounds on the solution and/or its derivatives can be taken in account. We are mainly interested in smoothness properties. In fact, whenever we try to learn to perform some task from a set of examples, we are making the implicit assumption that we are dealing with a smooth map. If this is not true, that is if small changes in the input do no determine small changes in the output, there is no hope to generalize, and therefore to learn.

One of the idea underlying regularization theory is that the solution of an ill-posed problem can be obtained from a variational principle, that contains both data and a priori information. We now sketch the general form of this variational principle and of its solution, and show how it leads to the Radial Basis Functions method and to some possible generalizations of it.

3.1 A Variational Principle for Multivariate Approximation

Suppose that the set $D = \{(\mathbf{x}_i, y_i) \in R^d \times R\}_{i=1}^N$ of data has been obtained by random sampling a function f, defined on R^d, in presence of noise. We are interested in recovering the function f, or an estimate of it, from the set of data D. If the class of functions to which f belongs is large enough, this problem has clearly an infinite number of solutions, and some a priori knowledge is needed in order to find a unique solution. When data are not noisy we are looking for a strict interpolant, and therefore need to pick up one among all the possible interpolants. If we know a priori that the original function has to satisfy some constraint, and if we can define a functional $\phi(f)$ (usually called *stabilizer*) that measures the deviation from this constraint, we can choose as a solution the interpolant that minimizes $\phi(f)$. If data are noisy we do not want to interpolate the data, and a better choice consists in minimizing the following functional:

$$H[f] = \sum_{i=1}^{N}(y_i - f(\mathbf{x}_i))^2 + \lambda\phi(f) \ . \tag{2}$$

where λ is a positive parameter. The first term measures the distance between the data and the desired solution f, the second term measures the cost associated with the deviation from the constraint and λ, the *regularization parameter*, determines the trade-off between these two terms. This is, in essence, one of the ideas underlying regularization theory, applied to the approximation problem.

As previously said, we are interested in the case in which the stabilizer ϕ enforces some smoothness constraint, and therefore we face the problem to choose a suitable class of stabilizers.

Functionals that can be used to measure smoothness of a function have been studied in the past, and many of them have the property to be *semi-norms* in some Banach space. A common choice (Duchon, 1977; Meinguet, 1979, 1979a; Wahba, 1990 and references therein), is the following

$$\phi(f) = \sum_{|\alpha|=m} \|D^\alpha f\|^2 \tag{3}$$

where α is a multi-index, $|\alpha| = \alpha_1 + \alpha_2 + \ldots + \alpha_d$, D^α is the derivative of order α, and $\|\cdot\|$ is the standard L^2 norm.

The functional of eq. (3) is not the only sensible choice. Madych and Nelson (1990) proved that any conditionally positive definite function G can be used to define a semi-norm in a space $X_G \subset C[R^d]$ that generalizes the semi-norm (3) (see also (Dyn, 1987, 1991)) and therefore perfectly fits in the framework of regularization theory. In essence, to any conditionally positive definite function G of order m it is possible to associate the functional

$$\phi(f) = \int_{R^d} ds \ \frac{|\tilde{f}(\mathbf{s})|^2}{\tilde{G}(\mathbf{s})} \tag{4}$$

where $\tilde{f}$ and $\tilde{G}$ are the generalized Fourier transforms of f and G. It turns out that the functional (4) is a semi-norm, whose null space is the set of polynomials of degree at most m. If this functional is used as a stabilizer in eq. (2), the solution of the minimization problem is of the form

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i G(\mathbf{x} - \mathbf{x}_i) + \sum_{\alpha=1}^{k} d_\alpha \gamma_\alpha(\mathbf{x}) \tag{5}$$

where $\{\gamma_\alpha\}_{\alpha=1}^k$ is a basis in the space of polynomials of degree at most m, and c_i and d_α are coefficients that have to be determined.

If expression (5) is substituted in the functional (2), $H[f]$ becomes a function $H(\mathbf{c}, \mathbf{d})$ of the variables c_i and d_α. Therefore the coefficients c_i and d_α of equation (5) can be obtained by minimizing $H(\mathbf{c},\mathbf{d})$ with respect to them, obtaining the following linear system:

$$(G+\lambda I)\mathbf{c}+\Gamma^T\mathbf{d}=\mathbf{y} \tag{6}$$

$$\Gamma\mathbf{c}=0$$

where I is the identity matrix, and we have defined

$$(\mathbf{y})_i = y_i \ , \quad (\mathbf{c})_i = c_i \ , \quad (\mathbf{d})_i = d_i \ , \quad (G)_{ij} = G(\mathbf{x}_i - \mathbf{x}_j) \ , \quad (\Gamma)_{\alpha i} = \gamma_\alpha(\mathbf{x}_i)$$

Clearly, in the limit of $\lambda = 0$ these equations become the standard equation of the Radial Basis Functions technique, and the interpolation conditions $f(\mathbf{x}_i) = y_i$ are satisfied. If data are noisy, the value of λ should be proportional to the amount of noise in the data. The optimal value of λ can be found by means of techniques like Generalized Cross Validation (Wahba, 1990), but we do not discuss it here.

Of course the Radial Basis Functions approach to the problem of learning from examples has its drawbacks, and very often cannot be applied in a straightforward way to specific problems. Some approximations and extensions of the original method has to be developed, and the next section is devoted to this topic.

4 Extending Radial Basis Functions

The Radial Basis Functions technique seems to be an ideal tool to approximate functions in multidimensional spaces. However, at least for the kind of applications we are interested in, it suffers of two main drawbacks:

1. the coefficients of the Radial Basis Functions expansion are computed by solving a linear system of N equations, where N is the number of data points. In typical applications n is of the order of 10^3, and, moreover, the matrices associated to these linear systems are usually ill-conditioned. Preconditioning and iterative techniques can be used in order to deal with numerical instabilities (Dyn et al., 1986), but when the number of data points is several thousands the whole method is not very practical;

2. the Radial Basis Functions technique is based on the standard definition of Euclidean distance. However, in many situations the function to be approximated is defined on a space in which a natural notion of distance does not exist, for example beacuse different coordinates have different units of measurements. In this case it is crucial to define an alternative distance, whose particular form depends on the data, and has to be computed as well as the Radial Basis Functions coefficients.

We now proceed to show how to deal with these two drawbacks of the Radial Basis Functions technique.

4.1 Least Squares Radial Basis Functions

The solution given by regularization theory to the approximation problem is characterized by a very high number of parameters. In fact, even in moderately high dimensional cases ($6-10$ dimensions), the number of examples is the order of thousands, and so the number of parameters to be estimated. These considerations, along with the numerical problems encountered in solving such large linear systems, stimulated the developing of least squares techniques for Radial Basis Functions (Broomhead and Lowe, 1988; Poggio and Girosi, 1989, 1990; Sivakumar and Ward, 1991). We remind the reader that the original problem consists in minimizing the functional

$$H[f] = \sum_{i=1}^{N}(y_i - f(\mathbf{x}_i))^2 + \lambda\phi(f) \ . \tag{7}$$

where $\phi(f)$ is a seminorm associated to a conditionally definite positive function G (see eq. (4)). A least square Radial Basis Function is a solution to this problem of the form

$$f^*(\mathbf{x}) = \sum_{\alpha=1}^{n} c_\alpha G(\|\mathbf{x} - \mathbf{t}_\alpha\|) \tag{8}$$

where $\{\mathbf{t}_\alpha\}_{\alpha=1}^n$ is a fixed set of vectors, called "centers", whose locations may or may not coincide with some of the data points, and $n < N$. Substituting the least squares expansion (8) in the functional (7), and minimizing the resulting function $H[f^*]$ with respect to the coefficients c_α, the following linear system is obtained:

$$(G^T G + \lambda g)\mathbf{c} = G^T \mathbf{y} \ . \tag{9}$$

where we have defined the following vectors and matrices:

$$(\mathbf{y})_i = y_i \ , \quad (\mathbf{c})_\alpha = c_\alpha \ , \quad (G)_{i\alpha} = G(\|\mathbf{x}_i - \mathbf{t}_\alpha\|) \ , \quad (g)_{\alpha\beta} = G(\|\mathbf{t}_\alpha - \mathbf{t}_\beta\|) \ ,$$

and we used the fact that

$$\phi(f^*) = \int_{R^d} d\mathbf{s} \ \frac{|\tilde{f}^*(\mathbf{s})|^2}{\tilde{G}(\mathbf{s})} = \int_{R^d} d\mathbf{s} \ \frac{1}{\tilde{G}(\mathbf{s})} \sum_{\alpha,\beta=1}^{n} c_\alpha c_\beta e^{i\mathbf{s}\cdot(\mathbf{t}_\alpha - \mathbf{t}_\beta)}(\tilde{G}(\mathbf{s}))^2 =$$

$$= \sum_{\alpha,\beta=1}^{n} c_\alpha c_\beta \int_{R^d} d\mathbf{s} \ e^{i\mathbf{s}\cdot(\mathbf{t}_\alpha - \mathbf{t}_\beta)}\tilde{G}(\mathbf{s}) = \sum_{\alpha,\beta=1}^{n} c_\alpha c_\beta G(\|\mathbf{t}_\alpha - \mathbf{t}_\beta\|) \ .$$

Notice the similarity of the linear systems (9) and (6). However, in the least squares case the matrix that has to be inverted is $n \times n$, instead of $N \times N$. The case $\lambda = 0$ has been analyzed by Sivakumar and Ward (1991), that succeded in giving conditions under which the matrix $G^T G$ is invertible, and studied its condition number.

Notice that in this formulation there are two smoothing parameters, λ and the number n of basis functions. One could argue that one is more than enough and set

$\lambda = 0$. This is in fact what it is usually done (Broomhead and Lowe, 1988; Poggio and Edelman, 1990), even because it avoids the problem of choosing the right value for λ. However the smoothing efect of λ and n is different, and it has been shown that in some cases the choice $\lambda = 0$ is always suboptimal (von Golitschek and Schumaker, 1987; Wahba, 1990, pp. 99).

Least squares techniques greatly reduce the computational complexity of the approximation scheme, but have the obvious drawback of depending on the choice of the number and the locations of the centers $\mathbf{t}_\alpha$. If data have no structure, a common choice consists of setting the centers to a subset of the data points. If data are clustered, it is reasonable to rey to locate the centers of the clusters first and set the centers accordingly (Moody and Darken, 1989).

Another possibility consists in looking at the least squares expansion (8) from a different point of view, and regard the locations of the centers $\mathbf{t}_\alpha$ as unknowns, as well as the coefficients c_α. In order to find the optimal locations of the centers and of the coefficients we substitute expansion (8) in the functional (7) and look for the solution (if exists) of the following minimization problem:

$$\min_{c_\alpha, \mathbf{t}_\alpha} H[f^*] \; . \tag{10}$$

This is clearly a very difficult minimization problem, since $H[f^*]$ is certainly not convex, and many local minima may exist. Moreover there is not even a guarantee that a global minimum exists (although "good"local minima or "sufficiently low" values of $H[f^*]$ may be enough for our purposes).

In order to find solutions to this difficult minimization problem many techniques, as gradient descent, conjugate gradient or Levenberg-Marquardt (Press et al., 1987), can be used. All of them suffer the problem of getting stuck in local minima, and the easiest way to solve it is starting from several different points, and then select the best result. Nondeterministic techniques, as simulated annealing or genetic algorithms, (Kirkpatrick, Gelatt and Vecchi, 1983; Goldberg, 1989) seem to be more appropriate for dealing with local minima, but may be computationally very expensive. In Appendix (A) we will describe a simple stochastic algorithm (Caprile and Girosi, 1990), that is much simpler than simulated annealing, and that gave good results in a variety of cases.

Before introducing a further, and more important, extension, it is worth of having a closer look to the equations implied by eq. (10). We consider the case $\lambda = 0$, and look at the necessary conditions

$$\frac{\partial H[f^*]}{\partial c_\alpha} = 0 \; , \qquad \frac{\partial H[f^*]}{\partial \mathbf{t}_\alpha} = 0, \quad \alpha = 1, ..., n \; .$$

The equations for the coefficients clearly give us eq. (9). In order to write more easily the equations for the centers we define the function $h(x) = G(\sqrt{x})$. The equations for the centers can now be written in the following way:

$$\mathbf{t}_\alpha = \frac{\sum_{i=1}^n P_i^\alpha \mathbf{x}_i}{\sum_{i=1}^n P_i^\alpha} \tag{11}$$

where we have defined

$$P_i^\alpha = \Delta_i h'(\|\mathbf{x}_i - \mathbf{t}_\alpha\|^2) \ , \qquad \Delta_i = f(\mathbf{x}_i) - y_i \ ,$$

and h' is the first derivative of h.

Thus, the optimal centers, if exist, are weighted sums of the data points, where the weights depend on the center themselves. Notice that the sign of the weight of a data point may be positive or negative depending on the sign of the interpolation error at that point. If we consider basis functions that decrease at infinity, as the Gaussian or the inverse multiquadric, the magnitude of the weight increases with the interpolation error, but decreases with the distance between the center and the data point. Therefore the optimal locations of a center seems to be "locally" determined. However, since the sign of the weight may be positive or negative, an optimal center may be "attracted" or "repelled" by a data point, so that it does not need to lie in the convex hull of the neighboring data points. It is therefore not obvious that, in the case of clustered data points, optimal locations of the centers coincide with the centers of the clusters of the data points. An additional complication is that, in the case of decreasing basis functions, the gradient of the objective function $H[f^*]$ tends to zero at infinity, leading to an infinity of suboptimal solutions (see fig. 2). During the minimization procedure, therefore, it may happen that the centers move away from the data points, until they reach a sufficiently far region.

Figure 2 about here

The case in which the centers can be moved is therefore much more complicated than the standard least squares technique. This technique may lead to superior performances, at the expenses of a higher computational complexity, but a characterization of the cases in which it is useful to use it is still missing, as well as a study of the well-posedness of the problem.

In the next section we introduce another extension of the original Radial Basis Functions technique, that proved to be essential in many practical cases.

4.2 Weighted norm

In section (3) we have seen how Radial Basis Functions derive from a variational principle involving a seminorm $\phi(f)$. Radiality of the basis function is a consequence of the rotation invariance of $\phi(f)$. The seminorm $\phi(f)$ is a measure of smoothness of f, and its rotation invariance corresponds to the assumption that the derivatives of f must penalized in the same way. Since the derivative of a function is a measure of how much the function is sensitive to changes in that variable, the rotation invariance of

$\phi(f)$ corresponds to the assumption that all the variables have the same relevance. In practice this may be not always the case.

In fact, consider for example the problem of predicting a time series. The usual technique consists in assuming that the value of the variable at times $t+1$ depends on the values of that variable at times $t, t-1, t-2, \ldots, t-n$, with given n, and the map that has to be approximated is therefore:

$$x_{t+1} = f(x_t, x_{t-1}, \ldots, x_{t-n}) \ .$$

In most cases it is reasonable to assume that the relevance of the variable x_{t-n} is smaller than the one of the variable x_t, and therefore we expect the function f to be much smoother on the direction x_{t-n} than on the direction x_t. Therefore an appropriate stabilizing functional has to penalize more the derivatives with respect to the variable x_{t-n}, and allow larger values of the derivatives with respect to x_t. For example a stabilizing functional of the form

$$\phi(f) = \int dx_t dx_{t-1} \ldots dx_{t-n} \sum_{i=0}^{n} c_i^2 \left(\frac{\partial f}{\partial x_{t-i}} \right)^2$$

with $c_0 < c_1 < \ldots < c_n$ could be a reasonable choice. Notice that in the limit of $c_i \to \infty$ we force the function f not to depend on the variable x_{t-i}.

Another case in which a non radial stabilizer is appropriate is when the variables have different units of measurements. Consider, for example, the problem of controlling the dynamics of a two-joints robot arm, in which the input variables are $(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2, \ddot{\theta}_1, \ddot{\theta}_2)$, where θ_1 and θ_2 are the shoulder and elbow angles. If we express time in milliseconds, instead of seconds, θ_1 and θ_2 are unchanged, $\dot{\theta}_1$ and $\dot{\theta}_2$ are scaled by a factor 10^{-3} and $\ddot{\theta}_1$ and $\ddot{\theta}_2$ are scaled by a factor 10^{-6}. It is clear therefore that in general a non-radial, appropriate stabilizer must be selected, depending on the units of measurement that are currently used.

In order to write down a general class of non-radial stabilizer, we go back to the example about the time series. Let us define the function $f_W(\mathbf{x}) = f(W\mathbf{x})$, where W is the diagonal matrix $W = \mathrm{diag}(w_0, w_1, \ldots, w_n)$. A simple computation shows that:

$$\phi(f_W) = \frac{1}{\det W} \int dx_t dx_{t-1} \ldots dx_{t-n} \sum_{i=0}^{n} c_i^2 W_i^2 \left(\frac{\partial f}{\partial x_{t-i}} \right)^2 \ .$$

If we now choose $w_i = \frac{1}{c_i}$ we notice that the functional $\phi_W(f) = \phi(f_W)$ becomes rotation invariant. Therefore, an appropriate redefinition of the variables can transform stabilizers that are not rotation invariant in stabilizer with radial simmetry.

Reversing this argument, and not restricting ourselves to diagonal matrix W, we have now a simple way to model stabilizers which do not have radial simmetry. Starting from a radial stabilizer $\phi(f)$, we can define the stabilizer

$$\phi_{\mathbf{W}}(f) = \phi(f(\mathbf{W}^{-1}\mathbf{x})) \ ,$$

where $\mathbf{W}$ is now any invertible matrix. The regularization functional, whose minimum solution is the solution of the approximation problem, is now:

$$H_{\mathbf{W}}[f] = \sum_{i=1}^{N} (y_i - f(\mathbf{x}))^2 + \lambda \phi_{\mathbf{W}}(f) \ . \tag{12}$$

The interpretation of such a regularization functional is simple: the a priori knowledge consists in knowing that the function underlying the data satisfies an isotropic smoothness constraint in a space that is a linear transformation of the original space, where the linear transformation is represented by the matrix $\mathbf{W}$. In other words we are assuming that there exists a matrix $\mathbf{W}$ such that $\mathbf{W}^{-1}$ transforms an anisotropic class of functions in an isotropic one.

The solution of the variational problem is similar to the one we have shown in the previous section for the radial stabilizer ϕ. The only difference is that the Green's function is not radial anymore, and the minimum of the functional (12) becomes

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i G(\|\mathbf{x} - \mathbf{x}_i\|_{\mathbf{W}}) + \sum_{\alpha=1}^{k} d_\alpha \gamma_\alpha(\mathbf{x}) \tag{13}$$

where we have defined the *weighted norm*

$$\|\mathbf{x}\|_{\mathbf{W}}^2 = \mathbf{x}^T \mathbf{W}^T \mathbf{W} \mathbf{x} \ .$$

The solution is therefore given by a linear superposition of nonradial basis functions, some examples of which are shown in figure 3.

Figure 3 about here

However, as far as we rely on the a priori knowledge of the matrix $\mathbf{W}$, the use of this non radial basis functions is limited. In fact, in most cases we do not know exactly which is the linear transformation $\mathbf{W}^{-1}$ that transforms the anisotropic space in an isotropic one. We usually just know that such a matrix exists, and sometimes we can also have some guess on its more likely structure. It comes then natural to look at the functional (12) from a different perspective, and consider *both f and* $\mathbf{W}$ as unknown (Poggio and Girosi, 1990). We therefore look for the minimum of the following functional, that is formally the same of the one of eq. (12):

$$H[f, \mathbf{W}] = \sum_{i=1}^{N} (y_i - f(\mathbf{x}))^2 + \lambda \phi_{\mathbf{W}}(f) \ . \tag{14}$$

From a mathematical point of view, this functional is much more difficult to study than the standard regularization functional. We do not have at moment a proof of the existence of a minimum for this functional. All we can say is that the solution, if exists, has the form of eq. (13). Moreover, it is clear that if a solution exists is not unique. In fact, if the couple $(f, \mathbf{W})$ is a solution the couple $(f, \mathbf{SW})$, where $\mathbf{S}$ is any orthogonal matrix, is also a solution, due to the obvious identity $\|\mathbf{x}\|^2_{\mathbf{SW}} = \|\mathbf{x}\|^2_{\mathbf{W}}$.

Even if we do not know how to compute exactly the minimum of the functional (14), we can try to find approximated solutions. Following the motivations of the previous section, we are quite naturally led to look for solutions of the form

$$f^*(\mathbf{x}) = \sum_{\alpha=1}^{n} c_\alpha G(\|\mathbf{x} - \mathbf{t}_\alpha\|_{\mathbf{W}}) \; . \tag{15}$$

The parameters c_α, $\mathbf{t}_\alpha$ and $\mathbf{W}$ can now be found solving the minimization problem:

$$\min_{c_\alpha, \mathbf{t}_\alpha, \mathbf{W}} \; H[f^*, \mathbf{W}] \; .$$

Of course, this minimization problem is even more difficult than before, but much more information is obtained in this case. In fact, it is very useful to know the structure of the matrix $\mathbf{W}$, in practical cases. Moreover, the minimization problem can be greatly simplified if the centers are kept fixed, and it is usually easier to find good initial values for the center than for $\mathbf{W}$.

If there is no reason to think that the relevant variables are linear combinations of the current ones, it is sufficient to choose a diagonal $\mathbf{W}$ matrix, that takes in account the possibility that different variables have different scales. If the basis function is the Gaussian, the diagonal elements of the $\mathbf{W}$ matrix are simply the reciprocal of the variances along the coordinate directions.

The introduction of the $\mathbf{W}$ matrix gives us a richer and adaptive choice of basis functions, with possible great improvement of the results. The quest for a more flexible technique led Kansa (1990; see also the paper in this issue) to introduce radial basis functions with variable shape parameter. Allowing the parameter r of the multiquadric $f(x) = \sqrt{r^2 + x^2}$ to be different at different locations, and looking for its optimal values, much better performances were obtained, due to the increased flexibility of the method.

In the next section we give some other examples of why $\mathbf{W}$ it is so important, together with some numerical results.

5 Examples

As we have seen in section (2.2), in many practical problems the function that has to be approximated depends on a large number of variables. However, in some cases, the function can be described by a smaller number of variables than the one initially taken in account. We consider here the two following cases, in which the $\mathbf{W}$ matrix can be used to perform such "dimensionality reduction", and therefore reduce the complexity of the problem.

- Some of the variables are not independent. Suppose for example that $f = f(x_1, \ldots, x_d, x_{d+1}, \ldots, x_{d+k})$ and that $x_{d+n} = h_n(x_1, \ldots, x_d)$, $n = 1, \ldots, k$. In this case the data points lie on a d-dimensional manifold ambedded in a $k+d$-dimensional space[2]. Given the same number of data points, it is reasonable to think that a least square Radial Basis Functions approximation to f that uses only the d "relevant" variables can give a lower error than one that uses all the $k+d$ variables. Therefore, if we use the approximation scheme with the $\mathbf{W}$ matrix, and start considering all the $k+d$ variables, we expect $\mathbf{W}$ to converge to a diagonal matrix whose last k eigenvalues are zero. The usefulness of the $\mathbf{W}$ matrix is twofold:

1. If during the minimization we see that some of the eigenvalues of $\mathbf{W}$ are very small, we may restart the minimization procedure, disregarding the corresponding variables from the beginning. This can lead to an improvement of the results, and also to a speed-up of the convergence.
2. The structure of the eigenvalues of the $\mathbf{W}$ matrix gives us information on the nature of the problem, since can tell us which are the independent variables that are sufficient for the comprehension of the system we are looking at.

- The function depends only on some linear combinations of the variables. For example, consider the function $f(x,y,z) = h(x+y, y+z)$ or, as an extreme case $f(x,y,z) = h(x)$. In this case, if we could know in advance which are the relevant linear combination we could solve a problem in a smaller number of dimensions, with better chances of getting good results. The optimization over the $\mathbf{W}$ matrix is supposed to take in account all the possible linear combinations of the variables, and select the best one.

We now show to simple examples of applications of the $\mathbf{W}$ matrix.

Example 1

We consider the time series generated by the itration of the logistic map

$$x_{n+1} = 4x_n(1-x_n) \ . \tag{16}$$

We suppose not to know that the value of x at time $n+1$ depends only on the value of x at time n, and make the assumption that the data are generated by the iteration of a map:

$$x_{n+1} = f(x_{n-1}, x_n) \ . \tag{17}$$

Notice that this assumption is not wrong, since x_n can always be thought as a function of x_{n-1}. For example we can rewrite the map (16) in 3 other different ways:

[2]We do not consider here the more general case in which the dependence between the variables is given in the implicit form $g_n(x_1, \ldots, x_{d+k}) = 0, \ n = 1, \ldots, k$.

$$
\begin{aligned}
x_{n+1} &= 16x_{n-1}(1-x_{n-1})(1-x_n) \\
x_{n+1} &= 4x_n(1-4x_{n-1}(1-x_{n-1})) \\
x_{n+1} &= 16x_{n-1}(1-x_{n-1})(1-4x_{n-1}(1-x_{n-1}))
\end{aligned}
$$

All of them generates the same set of data for a 2 dimensional function $z = f(x, y)$:

$$
\begin{aligned}
(x_0, x_1) &\rightarrow x_2 \\
(x_1, x_2) &\rightarrow x_3 \\
&\vdots \\
(x_{n-1}, x_n) &\rightarrow x_{n+1}
\end{aligned}
$$

However this set of data has the two following uncommon features:

1. the data points (x, y) lie on the curve $y = 4x(1-x)$. This curve is the domain of definition of the function $z = f(x, y)$, that, therefore, describes a curve in R^3 rather than a surface;

2. the projection of the data on the plane $x = 0$ give a set of one dimensional data for a single valued function (the function $z = 4y(1-y)$).

Since it is not clear that a two dimensional Radial Basis Functions technique is a good way to approximate data from a curve, and since the one dimensional projection contains all the relevant information, it would be useful to be able to recognize that the x variable could be simply dropped from the data, leading to a much simpler one dimensional problem.

We therefore used the least squares Radial Basis Functions technique with the $\mathbf{W}$ matrix to approximate a two dimensional set of 50 data points from the logistic time series (16), with a Gaussian basis function. We used only 10 centers, whose locations were randomly chosen to coincide with some of the data points and kept fixed during the minimization. The result is shown in figure 4, where we report the approximated function $x_n = f(x_{n-2}, x_{n-1})$. We see that, since the $\mathbf{W}$ matrix converged to a diagonal matrix with a null first element, the value of x_n does not depend on x_{n-2}. Therefore it seems that the "simplest" interpretation of the data is also the one that minimizes the least squares error. Of course, also a solution in which $\mathbf{W}$ has a null second eigenvalue would be "simple", corresponding to a one dimensional map in which x_n is expressed as a function of x_{n-2} only. However, the projection of the data on the plane $x_{n-1} = 0$ is a 4th degree polynomial, and since we used only 10 centers, we expect to obtain a higher least squares error in this case. It is critical therefore, in order to get the "simplest" solution, that the number of centers is kept low: if we had as many centers as data points every choice of $\mathbf{W}$ would have led to a zero interpolation error, and the "simplest" solution would not have been found.

Figure 4 about here

Example 2

Here we consider the function $f(x,y) = \sin(7x+3y)$. This is apparently a function of two variables, but in an appropriate coordinate system we recognize it as function of one variable, since it is constant along the lines $7x+3y=c$, for all c. We therefore tested the least squares Radial Basis Functions scheme on this function, in order to see if the $\mathbf{W}$ matrix can help in understanding the structure of this function. We used 50 data points, 10 moving centers and the $\mathbf{W}$ matrix, in conjunction with a Gaussian basis function. In fig. (5) we show a section of one of the Gaussian basis functions at different stages of the approximation procedure. We start setting $\mathbf{W}$ to the identity matrix, so that the section is initially circular. After a certain number of iterations one of the eigenvectors of the $\mathbf{W}$ matrix is oriented along the direction $7x+3y=0$ and the corresponding eigenvalue is very small. Therefore the variance of the Gaussian along the direction $7x+3y=0$ is very large, and the reconstructed function almost does not vary along that direction. Therefore, looking at the structure of the eigenvalues and eigenvectors of $\mathbf{W}$ we extract information on the nature of the function to be approximated.

Figure 5 about here

6 Conclusions and Open Problems

The Radial Basis Functions technique has many interesting applications in Artificial Intelligence and Computer Science, as it can be seen by the large number of papers on Artificial Intelligence journals that report successful applications of Radial Basis Functions. The reason is that in many cases the problem of learning from examples can be formulated as the problem of reconstructing a surface from sparse data points, typically in a high dimensional space.

Radial Basis Functions have solid foundations in a variational framework, that naturally takes in account the possiblity of having noisy data, and also admits a probabilistic interpretation in the context of Bayes estimation (Wahba, 1990). Moreover, the technique has a simple interpretation in terms of feeforward neural networks with one layer of hidden units (Broomhead and Lowe, 1988; Poggio and Girosi, 1989) that, although not adding much to the theory, is significant from the practical point of view (in terms, for example, of parallelizability and VLSI implementation).

This method, however, cannot always be implemented as its original formulation prescribes, mainly for the two following reasons:

- computational complexity: the original technique is based on the solution of a linear system, whose dimension (and ill-conditioning) grows with the number of data points, that can be of several thousands;

- use of Euclidean distance: in many cases it is important to adopt a metric that differs from the Euclidean one. The metric is usually unknown, and it is important both because it greatly improves the results and because it contains relevant information about the role of the variables that are used.

We therefore proposed an approximation scheme based on the following expansion:

$$f(\mathbf{x}) = \sum_{\alpha=1}^{n} c_\alpha G(\|\mathbf{x} - \mathbf{t}_\alpha\|_{\mathbf{W}}) \ , \tag{18}$$

where the coefficients c_α, the centers $\mathbf{t}_\alpha$ and the matrix $\mathbf{W}$ are found according to a least square criterion.

The matrix $\mathbf{W}$ takes in account non radiality, and the different roles of the variables, and possibly singles out the relevant ones. The centers, especially if kept fixed, reduce the computational complexity of the problem, and smooth out the noise at the same time. Recent works based on this, or a similar, scheme (Botros and Atkeson, 1991; Saha et al., 1991; Kadirkamanathan et al., 1991; Aste and Caprile, 1991; Maruyama, Girosi and Poggio, 1991; Brunelli and Poggio, 1991, 1991a) indicates that the approximation scheme (18) may be very powerful in practical applications, and it is worth of studying it in more details. There are mainly two directions that would be interesting to pursue:

- theoretical study. The approximation properties of the scheme (18) are unknown. Best approximation is clearly important from a theoretical point of view, although in practice may not be a problem. Rates of convergence, in terms of number of centers and number of examples would be more important, since could help in establishing capabilities and limits of such technique in practical applications. Confidence intervals are important as well: we may not mind if the estimated distance of a mobile robot from the wall is only 90% correct, but we want to be sure that it is always in such an interval.

  The well-posedness of the problem of minimizing the functional $H[f, \mathbf{W}]$ of eq. (14) has not been studied yet. Moreover, it would be interesting to study functionals of the form

  $$H^*[f, \mathbf{W}] = H[f, \mathbf{W}] + I(\mathbf{W})$$

  where $I(\mathbf{W})$ is a term that reflects some a priori knowledge on the $\mathbf{W}$ matrix. For example, if we know that some of the variables may be not relevant, we may want to give lower cost to configurations of $\mathbf{W}$ with some small or zero eigenvalues;

- computational study. The computation of the parameters of the expansion (18) is a difficult task. It is greatly simplified when the centers are kept fixed, so that it is important to develop techniques for finding good locations of the centers in a pre-processing stage. It would also be important to have alternative techniques to compute the $\mathbf{W}$ matrix. For example, Botros and Atkeson (1991) developed an heuristic to estimate $\mathbf{W}$ from the data, or from an initial guess for the expansion (18). They also showed that if the initial estimate is sufficiently good, then the procedure can be iterated in order to get better and better approximations.

Another attractive possibility consists in computing expansion (18) by means of a iterative techniques similar to Projection Pursuit Regression (Huber, 1985). At each step a basis function can be added, and the minimization runs only over its parameters (something similar have been recently proposed by Platt (1991)). Therefore, instead of solving a big minimization problem once, many small minimization problems are sequentially solved. It has been shown that, in the case of Projection Pursuit Regression and neural networks, sequences of this type can achieve a rate of convergence $O(\frac{1}{\sqrt{n}})$ in any number of dimensions, provided the function being approximated belongs to a restricted class of functions (Jones, 1991; Barron, 1991). It also been shown that a similar result holds for some Radial Basis Functions scheme (Girosi and Anzellotti, 1991).

This area of research seems therefore to be rich of questions of many kinds, some more theoretical and some more practical. Hopefully theoretical results will help to use this technique in practical applications, and real world applications will suggest interesting theoretical questions.

Acknowledgements. The author is indebted with Tomaso Poggio, with whom most of this work has been done, and with Bruno Caprile, that developed the software for the simulations.

A A nondeterministic minimization algorithm

Traditional descent techniques, like as gradient descent and related methods (Press et al., 1987) – though popular – find harduous to cope with local minima, and require, at each step, the computation of the gradient – an operation that may happen to be very time consuming. In (Caprile and Girosi, 1990) we propose a nondeterministic algorithm that, although it is not guaranteed to attain the global minimum, usually find "good" local minima. The method is simple, straightforward to implement, and proved to be robust with respect to variations in the choice of parameters and initial conditions.

The algorithm is iterative, and consists, in essence, of a local random search, that is certainly not new in the theory of optimization. Starting from an initial point $\mathbf{x}_0$, a new point is randomly generated in a neighbour of $\mathbf{x}_0$ of radius a_0. If the objective function assumes a lower value at this new point than at the starting point $\mathbf{x}_0$, the new point is selected as $\mathbf{x}_1$, and will be used as starting point at the next step. Otherwise we simply set $\mathbf{x}_1 = \mathbf{x}_0$. Before iterating this procedure the value of the radius a_0 is also updated: if the new point was "successful" we double a_0, setting $a_1 = 2a_0$, and half it otherwise. Now the procedure can be iterated, starting from the new values $\mathbf{x}_1$ and a_1, until some form of convergence is reached. The algorithm can be therefore be seen as the temporal evolution of the following discrete stochastic dynamical system:

$$\begin{aligned} \mathbf{x}_{n+1} = &\quad \mathbf{x}_n + a_n \mathbf{s}_n \theta[g(\mathbf{x}_n) - g(\mathbf{x}_n + a_n \mathbf{s}_n)] \\ a_{n+1} = &\quad \tfrac{1}{2} a_n \{1 + 3\theta[g(\mathbf{x}_n) - g(\mathbf{x}_n + a_n \mathbf{s}_n)]\} \end{aligned}$$

where $\mathbf{s}_n$ is a random vector of norm smaller than one, that is generated at each step according to a uniform distribution. We notice that the dynamics of the radius a_n is the mechanism that allows the algorithm to escape the local minima. In fact, during a sequence of"successful" steps, the radius increases exponentially, and eventually allows to jump from a basin of attraction to another one. Instead to moving uphill, however, as in simulated annealing (Kirkpatrick, Gelatt and Vecchi, 1983), the search point "tunnels" through the hill that separates two local minima. This effect is shown in fig. (6) in the case of the minimization of a two dimensional objective function.

Figure 6 about here

Some improvements and modifications of this technique have been proposed (Tecchiolli and Brunelli, 1991), together with a new version (Caprile, Girosi and Poggio, 1991) that borrowed the idea of *cross-over* from genetic algorithms (Goldberg, 1989) to enhance the performances of the technique presented here.

Among the advantages of this technique are its simplicity, and the fact that it does not require any knowledge about the function that has to be minimized. Being based only on evaluation of the objective function, and not on its derivatives, and taking advantage of parallel machines, large numbers of iterations of the algorithm can be run, with better chances to find the global minimum.

Unfortunately a theoretical analysis of the performances of this algorithm is still missing, so that its use is justificated only by a quite large amount of experimental results.

References

[1] A. Saha et al. Oriented Non-Radial Basis Functions for Image Coding and Analysis. In R. Lippmann, J. Moody, and D. Touretzky, editors, *Advances in Neural information processings systems 3*, pages 728–734, San Mateo, CA, 1991. Morgan Kaufmann Publishers.

[2] M. Aste and B. Caprile. Learning to navigate using radial basis functions. Technical report, I.R.S.T., Povo (IT), 1991. (in preparation).

[3] A.R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. Technical Report 58, Department of Statistics, University of Illinois at Urbana-Champaign, Champaign, IL, March 1991.

[4] M. Bertero. Regularization methods for linear inverse problems. In C. G. Talenti, editor, *Inverse Problems*. Springer-Verlag, Berlin, 1986.

[5] S. Botros and C. Atkeson. Generalization properties of Radial Basis Function. In R. Lippmann, J. Moody, and D. Touretzky, editors, *Advances in Neural information processings systems 3*, San Mateo, CA, 1991. Morgan Kaufmann Publishers.

[6] D. Broomhead and G. King. Extracting qualitative dynamics from experimental data. *Physica*, 20(D), 1987.

[7] D.S. Broomhead and D. Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2:321–355, 1988.

[8] R. Brunelli and T. Poggio. Face recognition: features vs. templates. Technical Report 9110-04, I.R.S.T., Povo (IT), 1991.

[9] R. Brunelli and T. Poggio. HyperBF networks for gender recognition. Technical report, I.R.S.T., Povo (IT), 1991a. (in preparation).

[10] B. Caprile and F. Girosi. A nondeterministic minimization algorithm. A.I. Memo 1254, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, September 1990.

[11] B. Caprile, F. Girosi, and T. Poggio. A nondeterministic method for multivariate functions minimization. In *Architetture Parallele e Reti Neurali*, Vietri sul Mare, Salerno, (IT), May 8-10 1991. (in press).

[12] M. Casdagli. Nonlinear prediction of chaotic time-series. *Physica D*, 35:335–356,

1989.

[13] G. Cybenko. Approximation by superposition of a sigmoidal function. *Math. Control Systems Signals*, 2(4):303–314, 1989.

[14] J. Duchon. Spline minimizing rotation-invariant semi-norms in Sobolev spaces. In W. Schempp and K. Zeller, editors, *Constructive theory of functions os several variables, Lecture Notes in Mathematics, 571.* Springer-Verlag, Berlin, 1977.

[15] N. Dyn. Interpolation of scattered data by radial functions. In C.K. Chui, L.L. Schumaker, and F.I. Utreras, editors, *Topics in multivariate approximation.* Academic Press, New York, 1987.

[16] N. Dyn. Interpolation and approximation by radial and related functions. In C.K. Chui, L.L. Schumaker, and D.J. Ward, editors, *Approximation Theory, VI*, pages 211–234. Academic Press, New York, 1991.

[17] N. Dyn, D. Levin, and S. Rippa. Numerical procedures for surface fitting of scattered data by radial functions. *SIAM J. Sci. Stat. Comput.*, 7(2):639–659, April 1986.

[18] J.D. Farmer and J.J. Sidorowich. Predicting chaotic time series. *Physical Review Letters*, 59:845, 1987.

[19] R. Franke. Scattered data interpolation: tests of some method. *Math. Comp.*, 38(5):181–200, 1982.

[20] K. Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2:183–192, 1989.

[21] F. Girosi and G. Anzellotti. Rates of convergence of approximation by translates. A.I. Memo 1288, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1991.

[22] D.E. Goldberg. *Genetic algorithms in search, optimization, and machine learning.* Addison-Wesley, Reading, MA, 1989.

[23] J. Hadamard. *La theorie des equations aux derivees partielles.* Editions Scientifiques, Pekin, 1964.

[24] J.A. Hertz, A. Krogh, and R. Palmer. *Introduction to the theory of neural computation.* Addison-Wesley, Redwood City, CA, 1991.

[25] P.J. Huber. Projection pursuit. *The Annals of Statistics*, 13(2):435–475, 1985.

[26] L.K. Jones. A simple lemma on greedy approximation in Hilbert space and convergence rates for Projection Pursuit Regression and neural network training. *Annals of Statistics*, 1991. (to appear).

[27] V. Kadirkamanathan, M. Niranjan, and F. Fallside. Sequential adaptation of Radial Basis Functions neural networks and its applications to time-series prediction. In R. Lippmann, J. Moody, and D. Touretzky, editors, *Advances in Neural information processings systems 3*, pages 721–727, San Mateo, CA, 1991. Morgan Kaufmann Publishers.

[28] E.J. Kansa. Multiquadrics – a scattered data approximation scheme with applications to computational fluid dynamics – I. *Computers Math. Applic.*, 19(8/9):127–145, 1990a.

[29] E.J. Kansa. Multiquadrics – a scattered data approximation scheme with applications to computational fluid dynamics – II. *Computers Math. Applic.*, 19(8/9):147–161, 1990b.

[30] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220:219–227, 1983.

[31] R. P. Lippmann. Review of neural networks for speech recognition. *Neural Computation*, 1:1–38, 1989.

[32] W.R. Madych and S.A. Nelson. Multivariate interpolation and conditionally positive definite functions. II. *Mathematics of Computation*, 54(189):211–230, January 1990.

[33] M. Maruyama, F. Girosi, and T. Poggio. Techniques for learning from examples: Numerical comparisons and approximation power. A.I. Memo 1290, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1991.

[34] J. Meinguet. Multivariate interpolation at arbitrary points made simple. *J. Appl. Math. Phys.*, 30:292–304, 1979.

[35] C. A. Micchelli. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constr. Approx.*, 2:11–22, 1986.

[36] J. Moody and C. Darken. Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1(2):281–294, 1989.

[37] V.A. Morozov. *Methods for solving incorrectly posed problems.* Springer-Verlag, Berlin, 1984.

[38] J. Platt. A resource-allocating network for function interpolation. *Neural Computation*, 3:213–225, 1991.

[39] T. Poggio and S. Edelman. A network that learns to recognize 3D objects. *Nature*, 343:263–266, 1990.

[40] T. Poggio and F. Girosi. A theory of networks for approximation and learning. A.I. Memo No. 1140, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1989.

[41] T. Poggio and F. Girosi. Extension of a theory of networks for approximation and learning: dimensionality reduction and clustering. A.I. Memo 1167, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1990.

[42] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9), September 1990.

[43] D. Pomerlau. ALVINN: an autonomous land vehicle in a neural network. In D. Touretzky, editor, *Advances in Neural Information Processing Systems I*, pages 305–313, San Mateo, CA, 1989. Morgan Kaufmann.

[44] M. J. D. Powell. Radial basis functions for multivariable interpolation: a review. In J. C. Mason and M. G. Cox, editors, *Algorithms for Approximation.* Clarendon Press, Oxford, 1987.

[45] W.H. Press et al. *Numerical Recipes: the Art of Scientific Computing.* Cambridge University Press, Cambridge, 1987.

[46] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Parallel Distributed Processing.* MIT Press, Cambridge, MA, 1986.

[47] T. J. Sejnowski and C. R. Rosenberg. Parallel networks that learn to pronounce english text. *Complex Systems*, 1:145–168, 1987.

[48] N. Sivakumar and J.D. Ward. On the best least square fit by radial functions to multidimensional scattered data. Technical Report 251, Center for Approximation Theory, Texas A & M University, June 1991.

[49] F. Takens. Detecting strange attractors in fluid turbulence. In D. Rand and L.S. Young, editors, *Dynamical Systems and Turbulence.* Springer-Verlag, Berlin, 1981.

[50] G.P. Tecchiolli and R. Brunelli. On random minimization of functions. Technical Report 9109-09, I.R.S.T., Povo (IT), 1991. (to appear on *Biological Cybernetics*).

[51] A. N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Math. Dokl.*, 4:1035–1038, 1963.

[52] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-posed Problems*. W. H. Winston, Washington, D.C., 1977.

[53] M. von Golitschek and L. Schumaker. Data fitting by penalized least squares. In M. Cox, J. Mason, and A. Watson, editors, *Algorithms for Approximation II*. 1987.

[54] G. Wahba. *Splines Models for Observational Data*. Series in Applied Mathematics, Vol. 59, SIAM, Philadelphia, 1990.

Figure Captions

Figure 1: The examples (x_i, y_i) are points belonging to the graph of a function $f : X \to Y$.

Figure 2: a) A section of the objective function $H[f^*]$, in a two dimensional case, along the axis corresponding to the two coordinates of one center. Five Gaussian basis functions were used in the least squares Radial Basis Functions expansion f^*. The data set consisted of 50 points in the squares $[-1.1] \times [-1, 1]$, randomly drawn from the function f_7 of (Kansa, 1990a). Notice how the objective functions becomes constant outside the squares $[-1.1] \times [-1, 1]$. Any descent algorithm starting from the wrong side of the hill would tend to move this center away. b) The same as in (a), but with multiquadrics basis functions.

Figure 3: Nonradial basis function: a) multiquadric b) Gaussian.

Figure 4: The bars corresponds to 50 data points from the logistic time series, and the surface is the result of the least squares Radial Basis Functions technique with the $\mathbf{W}$ matrix and fixed centres. In (b) the projection of the surface on the $x(n) - x(n-1)$ plane is shown.

Figure 5: See text for explanation.

Figure 6: The figure shows the "tunnel effect" that allows the algorithm to escape local minima. The solid line is the trajectory of $\mathbf{x}_n$ during the minimization procedure.