Regularization Theory and Neural Networks Architectures[1]

Federico Girosi, Michael Jones and Tomaso Poggio

Center for Biological and Computational Learning
and
Artificial Intelligence Laboratory
Massachusetts Institute of Technology, Cambridge, MA 02139 USA

Abstract

We had previously shown that regularization principles lead to approximation schemes which are equivalent to networks with one layer of hidden units, called *Regularization Networks*. In particular, standard smoothness functionals lead to a subclass of regularization networks, the well known Radial Basis Functions approximation schemes. This paper shows that regularization networks encompass a much broader range of approximation schemes, including many of the popular general additive models and some of the neural networks. In particular, we introduce new classes of smoothness functionals that lead to different classes of basis functions. Additive splines as well as some tensor product splines can be obtained from appropriate classes of smoothness functionals. Furthermore, the same generalization that extends Radial Basis Functions (RBF) to Hyper Basis Functions (HBF) also leads from additive models to ridge approximation models, containing as special cases Breiman's hinge functions, some forms of Projection Pursuit Regression and several types of neural networks. We propose to use the term *Generalized Regularization Networks* for this broad class of approximation schemes that follow from an extension of regularization. In the probabilistic interpretation of regularization, the different classes of basis functions correspond to different classes of prior probabilities on the approximating function spaces, and therefore to different types of smoothness assumptions.

In summary, different multilayer networks with one hidden layer, which we collectively call Generalized Regularization Networks, correspond to different classes of priors and associated smoothness functionals in a classical regularization principle. Three broad classes are a) Radial Basis Functions that can be generalized to Hyper Basis Functions, b) some tensor product splines, and c) additive splines that can be generalized to schemes of the type of ridge approximation, hinge functions and several perceptron-like neural networks with one-hidden layer.

[1]This paper will appear on *Neural Computation*, vol. 7, pages 219–269, 1995. An earlier version of this paper appeared as MIT AI Memo 1430, 1993.

1 Introduction

In recent years we and others have argued that the task of learning from examples can be considered in many cases to be equivalent to multivariate function approximation, that is, to the problem of approximating a smooth function from sparse data, the examples. The interpretation of an approximation scheme in terms of networks and vice versa has also been extensively discussed (Barron and Barron, 1988; Poggio and Girosi, 1989, 1990, 1990b; Girosi, 1992; Broomhead and Lowe, 1988; Moody and Darken, 1988, 1989; White, 1989, 1990; Ripley, 1994; Omohundro, 1987; Kohonen, 1990; Lapedes and Farber, 1988; Rumelhart, Hinton and Williams, 1986; Hertz, Krogh and Palmer, 1991; Kung, 1993; Sejnowski and Rosenberg, 1987; Hurlbert and Poggio, 1988; Poggio, 1975).

In a series of papers we have explored a quite general approach to the problem of function approximation. The approach regularizes the ill-posed problem of function approximation from sparse data by assuming an appropriate prior on the class of approximating functions. Regularization techniques (Tikhonov, 1963; Tikhonov and Arsenin, 1977; Morozov, 1984; Bertero, 1986; Wahba, 1975, 1979, 1990) typically impose smoothness constraints on the approximating set of functions. It can be argued that some form of smoothness is necessary to allow meaningful generalization in approximation type problems (Poggio and Girosi, 1989, 1990). A similar argument can also be used (see section 9.1) in the case of classification where smoothness is a condition on the classification boundaries rather than on the input-output mapping itself. Our use of regularization, which follows the classical technique introduced by Tikhonov, identifies the approximating function as the minimizer of a cost functional that includes an *error term* and a smoothness functional, usually called a *stabilizer*. In the Bayesian interpretation of regularization (see Kimeldorf and Wahba, 1971; Wahba, 1990; Bertero, Poggio and Torre, 1988; Marroquin, Mitter and Poggio, 1987; Poggio, Torre and Koch, 1985) the stabilizer corresponds to a smoothness prior, and the error term to a model of the noise in the data (usually Gaussian and additive).

In Poggio and Girosi (1989, 1990, 1990b) and Girosi (1992) we showed that regularization principles lead to approximation schemes which are equivalent to networks with one "hidden" layer, which we call *Regularization Networks* (RN). In particular, we described how a certain class of radial stabilizers – and the associated priors in the equivalent Bayesian formulation – lead to a subclass of regularization networks, the already-known Radial Basis Functions (Powell, 1987, 1990; Franke, 1982, 1987; Micchelli, 1986; Kansa, 1990a,b; Madych and Nelson, 1990; Dyn, 1987, 1991; Hardy, 1971, 1990; Buhmann, 1990; Lancaster and Salkauskas, 1986; Broomhead and Lowe, 1988; Moody and Darken, 1988, 1989; Poggio and Girosi, 1990, 1990b; Girosi, 1992). The regularization networks with radial stabilizers we studied include many classical one-dimensional (Schumaker, 1981; de Boor, 1978) as well as multidimensional splines and approximation techniques, such as radial and non-radial Gaussian, thin-plate splines (Duchon, 1977; Meinguet,

1979; Grimson, 1982; Cox, 1984; Eubank, 1988) and multiquadric functions (Hardy, 1971, 1990). In Poggio and Girosi (1990, 1990a) we extended this class of networks to Hyper Basis Functions (HBF). In this paper we show that an extension of Regularization Networks, which we propose to call *Generalized Regularization Networks* (GRN), encompasses an even broader range of approximation schemes including, in addition to HBF, tensor product splines, many of the general additive models, and some of the neural networks. As expected, GRN have approximation properties of the same type as already shown for some of the neural networks (Girosi and Poggio, 1990; Cybenko, 1989; Hornik, Stinchcombe and White, 1989; White, 1990; Irie and Miyake, 1988; Funahashi, 1989, Barron, 1991, 1994; Jones, 1992; Mhaskar and Micchelli, 1992, 1993; Mhaskar, 1993, 1993a).

The plan of the paper is as follows. We first discuss the solution of the variational problem of regularization. We then introduce three different classes of stabilizers – and the corresponding priors in the equivalent Bayesian interpretation – that lead to different classes of basis functions: the well-known radial stabilizers, tensor-product stabilizers, and the new additive stabilizers that underlie additive splines of different types. It is then possible to show that the same argument that extends Radial Basis Functions to Hyper Basis Functions also leads from additive models to some ridge approximation schemes, defined as

$$f(\mathbf{x}) = \sum_{\mu=1}^{K} h_\mu(\mathbf{w}_\mu \cdot \mathbf{x})$$

where h_μ are appropriate one-dimensional functions.

Special cases of ridge approximation are Breiman's hinge functions (1993), Projection Pursuit Regression (PPR) (Friedman and Stuezle, 1981; Huber, 1985; Diaconis and Freedman, 1984; Donoho and Johnstone, 1989; Moody and Yarvin, 1991) and Multilayer Perceptrons (Lapedes and Farber, 1988; Rumelhart, Hinton and Williams, 1986; Hertz, Krogh and Palmer, 1991; Kung, 1993; Sejnowski and Rosenberg, 1987). Simple numerical experiments are then described to illustrate the theoretical arguments.

In summary, the chain of our arguments shows that some ridge approximation schemes are approximations of Regularization Networks with appropriate additive stabilizers. The form of h_μ depends on the stabilizer, and includes in particular cubic splines (used in typical implementations of PPR) and one-dimensional Gaussians. Perceptron-like neural networks with one-hidden layer and with a Gaussian activation function are included. It seems impossible, however, to directly derive from regularization principles the sigmoidal activation functions typically used in feedforward neural networks. We discuss, however, in a simple example, the close relationship between basis functions of the hinge, the sigmoid and the Gaussian type.

The appendices deal with observations related to the main results of the paper and

more technical details.

2 The regularization approach to the approximation problem

Suppose that the set $g = \{(\mathbf{x}_i, y_i) \in R^d \times R\}_{i=1}^N$ of data has been obtained by random sampling a function f, belonging to some space of functions X defined on R^d, in the presence of noise, and suppose we are interested in recovering the function f, or an estimate of it, from the set of data g. This problem is clearly ill-posed, since it has an infinite number of solutions. In order to choose one particular solution we need to have some *a priori* knowledge of the function that has to be reconstructed. The most common form of *a priori* knowledge consists in assuming that the function is *smooth*, in the sense that two similar inputs correspond to two similar outputs. The main idea underlying regularization theory is that the solution of an ill-posed problem can be obtained from a variational principle, which contains both the data and prior smoothness information. Smoothness is taken into account by defining a *smoothness functional* $\phi[f]$ in such a way that lower values of the functional correspond to smoother functions. Since we look for a function that is simultaneously close to the data and also smooth, it is natural to choose as a solution of the approximation problem the function that minimizes the following functional:

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

where λ is a positive number that is usually called the *regularization parameter*. The first term is enforcing closeness to the data, and the second smoothness, while the regularization parameter controls the tradeoff between these two term, and can be chosen according to cross-validation techniques (Allen, 1974; Wahba and Wold, 1975; Golub, Heath and Wahba, 1979; Craven and Wahba, 1979; Utreras, 1979; Wahba, 1985) or to some other principle, such as Structural Risk Minimization (Vapnik, 1988).

It can be shown that, for a wide class of functionals ϕ, the solutions of the minimization of the functional (1) all have the same form. Although a detailed and rigorous derivation of the solution of this problem is out of the scope of this paper, a simple derivation of this general result is presented in appendix (A). In this section we just present a family of smoothness functionals and the corresponding solutions of the variational problem. We refer the reader to the current literature for the mathematical details (Wahba, 1990; Madych and Nelson, 1990; Dyn, 1987).

We first need to give a more precise definition of what we mean by smoothness and define a class of suitable smoothness functionals. We refer to smoothness as a measure

of the "oscillatory" behavior of a function. Therefore, within a class of differentiable functions, one function will be said to be smoother than another one if it oscillates less. If we look at the functions in the frequency domain, we may say that a function is smoother than another one if it has less energy at high frequency (smaller bandwidth). The high frequency content of a function can be measured by first high-pass filtering the function, and then measuring the power, that is the L_2 norm, of the result. In formulas, this suggests defining smoothness functionals of the form:

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

where $\tilde{}$ indicates the Fourier transform, $\tilde{G}$ is some positive function that tends to zero as $\|\mathbf{s}\| \to \infty$ (so that $\frac{1}{\tilde{G}}$ is an high-pass filter) and for which the class of functions such that this expression is well defined is not empty. For a well defined class of functions G (Madych and Nelson, 1990; Dyn, 1991; Dyn et al., 1989) this functional is a semi-norm, with a finite dimensional null space $\mathcal{N}$. The next section will be devoted to giving examples of the possible choices for the stabilizer ϕ. For the moment we just assume that it can be written as in equation (2), and make the additional assumption that $\tilde{G}$ is symmetric, so that its Fourier transform G is real and symmetric. In this case it is possible to show (see appendix (A) for a sketch of the proof) that the function that minimizes the functional (1) has the form:

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i G(\mathbf{x} - \mathbf{x}_i) + \sum_{\alpha=1}^{k} d_\alpha \psi_\alpha(\mathbf{x}) \tag{3}$$

where $\{\psi_\alpha\}_{\alpha=1}^{k}$ is a basis in the k dimensional null space $\mathcal{N}$ of the functional ϕ, that in most cases is a set of polynomials, and therefore will be referred to as the "polynomial term" in equation (3). The coefficients d_α and c_i depend on the data, and satisfy the following linear system:

$$(G + \lambda I)\mathbf{c} + \Psi^T \mathbf{d} = \mathbf{y} \tag{4}$$

$$\Psi \mathbf{c} = 0 \tag{5}$$

where I is the identity matrix, and we have defined

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

Notice that if the data term in equation (1) is replaced by $\sum_{i=1}^{N} V(f(\mathbf{x}_i) - y_i)$ where V is any differentiable function, the solution of the variational principle has still the form

3, but the coefficients cannot be found anymore by solving a linear system of equations (Girosi, 1991; Girosi, Poggio and Caprile, 1991).

The existence of a solution to the linear system shown above is guaranteed by the existence of the solution of the variational problem. The case of $\lambda = 0$ corresponds to pure interpolation. In this case the existence of an exact solution of the linear system of equations depends on the properties of the basis function G (Micchelli, 1986).

The approximation scheme of equation (3) has a simple interpretation in terms of a network with one layer of hidden units, which we call a *Regularization Network* (RN). Appendix B describes the extension to the vector output scheme.

In summary, the argument of this section shows that using a regularization network of the form (3), for a certain class of basis functions G, is equivalent to minimizing the functional (1). In particular, the choice of G is equivalent to the corresponding choice of the smoothness functional (2).

2.1 Dual representation of Regularization Networks

Consider an approximating function of the form (3), neglecting the "polynomial term" for simplicity. A compact notation for this expression is:

$$f(\mathbf{x}) = \mathbf{c} \cdot \mathbf{g}(\mathbf{x}) \tag{6}$$

where $\mathbf{g}(\mathbf{x})$ is the vector of functions such that $(\mathbf{g}(\mathbf{x}))_i = G(\mathbf{x} - \mathbf{x}_i)$. Since the coefficients $\mathbf{c}$ satisfy the linear system (4), solution (6) becomes

$$f(\mathbf{x}) = (G + \lambda I)^{-1}\mathbf{y} \cdot \mathbf{g}(\mathbf{x}) \ .$$

We can rewrite this expression as

$$f(\mathbf{x}) = \sum_{i=1}^{N} y_i b_i(\mathbf{x}) = \mathbf{y} \cdot \mathbf{b}(\mathbf{x}) \tag{7}$$

in which the vector $\mathbf{b}(\mathbf{x})$ of basis functions is defined

$$\mathbf{b}(\mathbf{x}) = (G + \lambda I)^{-1}\mathbf{g}(\mathbf{x}) \tag{8}$$

and now depends on all the data points and on the regularization parameter λ. The representation (7) of the solution of the approximation problem is known as the *dual* of equation (6), and the basis functions $b_i(\mathbf{x})$ are called the *equivalent kernels*, because of the similarity between equation (7) and the kernel smoothing technique that we will define in section 2.2 (Silverman, 1984; Härdle, 1990; Hastie and Tibshirani, 1990). While in equation (6) the "difficult" part is the computation of the vector of coefficients c_i, the set of basis functions $\mathbf{g}(\mathbf{x})$ being easily built, in equation (7) the "difficult" part is the

computation of the basis functions $\mathbf{b}(\mathbf{x})$, the coefficients of the expansion being explicitly given by the y_i. Notice that $\mathbf{b}(\mathbf{x})$ depends on the distribution of the data in the input space and that the kernels $b_i(\mathbf{x})$, unlike the kernels $G(\mathbf{x}-\mathbf{x}_i)$, are not translated replicas of the same kernel. Notice also that, as shown in appendix B, a dual representation of the form (7) exists for all the approximation schemes that consists of linear superpositions of arbitrary numbers of basis functions, as long as the error criterion that is used to determine the parameters of the approximation is quadratic.

The dual representation provides an intuitive way of looking at the approximation scheme (3): the value of the approximating function at an evaluation point $\mathbf{x}$ is explicitly expressed as a weighted sum of the values y_i of the function at the examples $\mathbf{x}_i$. This concept is not new in approximation theory, and has been used, for example, in the theory of quasi-interpolation. The case in which the data points $\{\mathbf{x}_i\}$ coincide with the multi-integers $\mathcal{Z}^d$, where $\mathcal{Z}$ is the set of integers number, has been extensively studied in the literature, and it is also known as *Schoenberg's approximation* (Schoenberg, 1946a, 1946b, 1969; Rabut, 1991, 1992; Madych and Nelson, 1990a; Jackson, 1988; de Boor, 1990; Buhmann, 1990, 1991; Dyn et al., 1989). In this case, an approximation f^* to a function f is sought of the form:

$$f^*(\mathbf{x}) = \sum_{\mathbf{j}\in\mathcal{Z}^d} f(\mathbf{j})\psi(\mathbf{x}-\mathbf{j}) \ . \tag{9}$$

where ψ is some fast-decaying function that is a linear combination of Radial Basis Functions. The approximation scheme (9) is therefore a linear superposition of Radial Basis Functions in which the functions $\psi(\mathbf{x}-\mathbf{j})$ play the role of equivalent kernels. Quasi-interpolation is interesting because it could provide good approximation without the need of solving complex minimization problems or solving large linear systems. For a discussion of such non iterative training algorithms see Mhaskar (1993a) and references therein.

Although difficult to prove rigorously, we can expect the kernels $b_i(\mathbf{x})$ to decrease with the distance of the data points $\mathbf{x}_i$ from the evaluation point, so that only the neighboring points affect the estimate of the function at $\mathbf{x}$, providing therefore a "local" approximation scheme. Even if the original basis function G is not "local", like the multiquadric $G(\mathbf{x}) = \sqrt{1+\|\mathbf{x}\|^2}$, the basis functions $b_i(\mathbf{x})$ are bell shaped, local functions, whose locality will depend on the choice of the basis function G, on the density of data points, and on the regularization parameter λ. This shows that apparently "global" approximation schemes can be regarded as local, *memory-based* techniques (see equation 7) (Mhaskar, 1993a). It should be noted however, that these techniques do not have the highest possible degree of locality, since the parameter that controls the locality is the regularization parameter λ, that is the same for all the kernels. It is possible to devise even more local techniques, in which each kernel has a parameter that controls its locality (Bottou and Vapnik, 1992; Vapnik, personal communication).

When the data are equally spaced on an infinite grid, we expect the basis functions $b_i(\mathbf{x})$ to become translation invariant, and therefore the dual representation (7) becomes a convolution filter. For a study of the properties of these filters in the case of one dimensional cubic splines see the work of Silverman (1984), who gives explicit results for the shape of the equivalent kernel.

Let us consider some simple experiments that show the shape of the equivalent kernels in specific situations. We first considered a data set composed of 36 equally spaced points on the domain $[0,1] \times [0,1]$, at the nodes of a regular grid with spacing equal to 0.2. We use the multiquadric basis functions $G(\mathbf{x}) = \sqrt{\sigma^2 + \|\mathbf{x}\|^2}$, where σ has been set to 0.2. Figure 1a shows the original multiquadric function, and figure 1b the equivalent kernel b_{16}, in the case of $\lambda = 0$, where, according to definition (8)

$$b_i(\mathbf{x}) = \sum_{j=1}^{36} (G^{-1})_{ij} G(\mathbf{x} - \mathbf{x}_j) \ .$$

All the other kernels, except those close to the border, are very similar, since the data are equally spaced, and translation invariance holds approximately.

Consider now a one dimensional example with a multiquadric basis function:

$$G(x) = \sqrt{\sigma^2 + x^2} \ .$$

The data set was chosen to be a non uniform sampling of the interval $[0,1]$, that is the set

$$\{0.0, 0.1, 0.2, 0.3, 0.4, 0.7, 1.0\} \ .$$

In figure 1c, 1d and 1e we have drawn, respectively, the equivalent kernels b_3, b_5 and b_6, under the same definitions. Notice that all of them are bell-shaped, although the original basis function is an increasing, cup-shaped function. Notice, moreover, that the shape of the equivalent kernels changes from b_3 to b_6, becoming broader in moving from a high to low sample density region. This phenomenon has been shown by Silverman (1986) for cubic splines, but we expect it to appear in much more general cases.

The connection between regularization theory and the dual representation (7) becomes clear in the special case of "continuous" data, for which the regularization functional has the form:

$$H[f] = \int d\mathbf{x} \ (f(\mathbf{x}) - y(\mathbf{x}))^2 + \lambda \phi[f] \ . \tag{10}$$

where $y(\mathbf{x})$ is the function to be approximated. This functional can be intuitively seen as the limit of the functional (1) when the number of data points goes to infinity and their spacing is uniform. It is easily seen that, when the stabilizer $\phi[f]$ is of the form (2), the solution of the regularization functional (10) is

$$f(\mathbf{x}) = y(\mathbf{x}) * B(\mathbf{x}) \tag{11}$$

where $B(\mathbf{x})$ is the Fourier transform of

$$\tilde{B}(\mathbf{s}) = \frac{\tilde{G}(\mathbf{s})}{\lambda + \tilde{G}(\mathbf{s})} \ ,$$

(see Poggio, Voorhes and Yuille, (1988) for some examples of $B(\mathbf{x})$). The solution (11) is therefore a filtered version of the original function $y(\mathbf{x})$ and, consistently with the results of Silverman (1984), has the form (7), where the equivalent kernels are translates of the function $B(\mathbf{x})$ defined above. Notice the effect of the regularization parameter: for $\lambda = 0$ the equivalent kernel $B(\mathbf{x})$ is a Dirac delta function, and $f(\mathbf{x}) = y(\mathbf{x})$ (no noise) , while for $\lambda \to \infty$ we have $B(\mathbf{x}) = G(\mathbf{x})$ and $f = G * y$ (a low pass filter).

The dual representation is illuminating and especially interesting for the case of a multi-output network – approximating a vector field – that is discussed in Appendix B.

2.2 Normalized kernels

An approximation technique very similar to Radial Basis Functions is the so-called *Normalized Radial Basis Functions* (Moody and Darken, 1988, 1989). A Normalized Radial Basis Functions expansion is a function of the form:

$$f(\mathbf{x}) = \frac{\sum_{\alpha=1}^{n} c_\alpha G(\mathbf{x} - \mathbf{t}_\alpha)}{\sum_{\alpha=1}^{n} G(\mathbf{x} - \mathbf{t}_\alpha)} \ . \tag{12}$$

The only difference between equation (12) and Radial Basis Functions is the normalization factor in the denominator, which is an estimate of the probability distribution of the data. A discussion about the relation between normalized Gaussian basis function networks, Gaussian mixtures and Gaussian mixture classifiers can be found in the work of Tresp, Hollatz and Ahmad (1993). In the rest of this section we show that a particular version of this approximation scheme has again a tight connection to regularization theory.

Let $P(\mathbf{x}, y)$ be the joint probability of inputs and outputs of the network, and let us assume that we have a sample of N pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ randomly drawn according to P. Our goal is to build an estimator (a network) f that minimizes the *expected risk*:

$$I[f] = \int \ d\mathbf{x}dy P(\mathbf{x}, y)(y - f(\mathbf{x}))^2 \ . \tag{13}$$

This cannot be done, since the probability P is unknown, and usually the *empirical risk*:

$$I_{\text{emp}}[f] = \frac{1}{N} \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i))^2 \tag{14}$$

is minimized instead. An alternative consists in obtaining an approximation of the probability $P(\mathbf{x}, y)$ first, and then in minimizing the expected risk. If this option is chosen, one could use the regularization approach to probability estimation (Vapnik and Stefanyuk, 1978; Aidu and Vapnik, 1989; Vapnik, 1982), that leads to the well known technique of *Parzen windows*. A Parzen window estimator P^* for the probability distribution of a set of data $\{\mathbf{z}_i\}_{i=1}^N$ has the form:

$$P^*(\mathbf{z}) = \frac{1}{Nh} \sum_{i=1}^{N} \Phi\left(\frac{\mathbf{z} - \mathbf{z}_i}{h}\right) \tag{15}$$

where Φ is an appropriate kernel, for example a Gaussian, whose L_1 norm is 1, and where h is a positive parameter, that, for simplicity, we set to 1 from now on. If the joint probability $P(\mathbf{x}, y)$ in the expected risk (13) is approximated with a Parzen window estimator P^*, we obtain an approximated expression for the expected risk, $I^*[f]$, that can be explicitly minimized. In order to show how this can be done, we notice that we need to approximate the probability distribution $P(\mathbf{x}, y)$, and therefore the random variable $\mathbf{z}$ of equation (15) is $\mathbf{z} = (\mathbf{x}, y)$. Hence, we choose a kernel of the following form[2]:

$$\Phi(\mathbf{z}) = K(\|\mathbf{x}\|)K(y)$$

where K is a standard one-dimensional, symmetric kernel, like the Gaussian. The Parzen window estimator to $P(\mathbf{x}, y)$ is therefore:

$$P^*(\mathbf{x}, y) = \frac{1}{N} \sum_{i=1}^{N} K\left(\|\mathbf{x} - \mathbf{x}_i\|\right) K(y - y_i) \tag{16}$$

An approximation to the expected risk is therefore obtained as:

$$I^*[f] = \frac{1}{N} \sum_{i=1}^{N} \int d\mathbf{x} dy K\left(\|\mathbf{x} - \mathbf{x}_i\|\right) K(y - y_i)(y - f(\mathbf{x}))^2 \ .$$

In order to find an analytical expression for the minimum of $I^*[f]$ we impose the stationarity constraint:

$$\frac{\delta \tilde{I}[f]}{\delta f(\mathbf{s})} = 0 \ ,$$

that leads to the following equation:

[2] Any kernel of the form $\Phi(\mathbf{z}) = K(\mathbf{x}, y)$ in which the function K is even in each of the variables $\mathbf{x}$ and y would lead to the same conclusions that we obtain for this choice.

$$\sum_{i=1}^{N} \int d\mathbf{x} dy K\left(\|\mathbf{x}-\mathbf{x}_i\|\right) K(y-y_i)(y-f(\mathbf{x}))\delta(\mathbf{x}-\mathbf{s}) = 0 \ .$$

Performing the integral over $\mathbf{x}$, and using the fact that $\|K\|_{L_1} = 1$ we obtain:

$$f(\mathbf{x}) = \frac{\sum_{i=1}^{N} K\left(\|\mathbf{x}-\mathbf{x}_i\|\right) \int dy K(y-y_i) y}{\sum_{i=1}^{N} K\left(\|\mathbf{x}-\mathbf{x}_i\|\right)} \ .$$

Performing a change of variable in the integral of the previous expression and using the fact that the kernel K is symmetric, we finally conclude that the function that minimizes the approximated expected risk is:

$$f(\mathbf{x}) = \frac{\sum_{i=1}^{N} y_i K(\|\mathbf{x}-\mathbf{x}_i\|)}{\sum_{i=1}^{N} K(\|\mathbf{x}-\mathbf{x}_i)\|} \ . \tag{17}$$

The right hand side of the equation converges to f when the number of examples goes to infinity, provided that the scale factor h tends to zero at an appropriate rate. This form of approximation is known as *kernel regression*, or *Nadaraya-Watson estimator*, and it has been the subject of extensive study in the statistics community (Nadaraya, 1964; Watson, 1964; Rosenblatt, 1971; Priestley and Chao, 1972; Gasser and Müller, 1985; Devroye and Wagner, 1980). A similar derivation of equation (17) has been given by Specht (1991), but we should remark that this equation is usually derived in a different way, within the framework of locally weighted regression, assuming a locally constant model (Hardle, 1990) with a local weight function K.

Notice that this equation has the form of equation (12), in which the centers coincide with the examples, and the coefficients c_i are simply the values y_i of the function at the data points $\mathbf{x}_i$. On the other hand, the equation is an estimate of f which is linear in the observations y_i and has therefore also the general form of equation (7).

The Parzen window estimator, and therefore expression (17), can be derived in the framework of regularization theory (Vapnik and Stefanyuk, 1978; Aidu and Vapnik, 1989; Vapnik, 1982) under a smoothness assumption on the probability distribution that has to be estimated. This means that, in order to derive equation (17), a smoothness assumption has to be made on the joint probability distribution $P(\mathbf{x}, y)$, rather than on the regression function as in (2).

3 Classes of stabilizers

In the previous section we considered the class of stabilizers of the form:

$$\phi[f] = \int_{R^d} d\mathbf{s} \ \frac{|\tilde{f}(\mathbf{s})|^2}{\tilde{G}(\mathbf{s})} \tag{18}$$

and we have seen that the solution of the minimization problem always has the same form. In this section we discuss three different types of stabilizers belonging to the class (18), corresponding to different properties of the basis functions G. Each of them corresponds to different *a priori* assumptions on the smoothness of the function that must be approximated.

3.1 Radial stabilizers

Most of the commonly used stabilizers have radial symmetry, that is, they satisfy the following equation:

$$\phi[f(\mathbf{x})] = \phi[f(R\mathbf{x})]$$

for any rotation matrix R. This choice reflects the *a priori* assumption that all the variables have the same relevance, and that there are no privileged directions. Rotation invariant stabilizers correspond to radial basis function $G(\|\mathbf{x}\|)$. Much attention has been dedicated to this case, and the corresponding approximation technique is known as Radial Basis Functions (Powell, 1987, 1990; Franke, 1982, 1987; Micchelli, 1986; Kansa, 1990a,b; Madych and Nelson, 1990; Dyn, 1987, 1991; Hardy, 1971, 1990; Buhmann, 1990; Lancaster and Salkauskas, 1986; Broomhead and Lowe, 1988; Moody and Darken, 1988, 1989; Poggio and Girosi, 1990, 1990b; Girosi, 1992). The class of admissible Radial Basis Functions is the class of conditionally positive definite functions (Micchelli, 1986) of any order, since it has been shown (Madych and Nelson, 1990; Dyn, 1991) that in this case the functional of equation (18) is a semi-norm, and the associated variational problem is well defined. All the Radial Basis Functions can therefore be derived in this framework. We explicitly give two important examples.

Duchon multidimensional splines

Duchon (1977) considered measures of smoothness of the form

$$\phi[f] = \int_{R^d} d\mathbf{s}\ \|\mathbf{s}\|^{2m} |\tilde{f}(\mathbf{s})|^2\ .$$

In this case $\tilde{G}(\mathbf{s}) = \frac{1}{\|\mathbf{s}\|^{2m}}$ and the corresponding basis function is therefore

$$G(\mathbf{x}) = \begin{cases} \|\mathbf{x}\|^{2m-d} \ln \|\mathbf{x}\| & \text{if } 2m > d \text{ and } d \text{ is even} \\ \|\mathbf{x}\|^{2m-d} & \text{otherwise.} \end{cases} \tag{19}$$

In this case the null space of $\phi[f]$ is the vector space of polynomials of degree at most m in d variables, whose dimension is

$$k = \left(\begin{array}{c} d+m-1 \\ d \end{array} \right) .$$

These basis functions are radial and conditionally positive definite, so that they represent just particular instances of the well known Radial Basis Functions technique (Micchelli, 1986; Wahba, 1990). In two dimensions, for $m = 2$, equation (19) yields the so called "thin plate" basis function $G(\mathbf{x}) = \|\mathbf{x}\|^2 \ln \|\mathbf{x}\|$ (Harder and Desmarais, 1972; Grimson, 1982).

The Gaussian

A stabilizer of the form

$$\phi[f] = \int_{R^d} d\mathbf{s} \; e^{\frac{\|\mathbf{s}\|^2}{\beta}} |\tilde{f}(\mathbf{s})|^2 \; ,$$

where β is a fixed positive parameter, has $\tilde{G}(\mathbf{s}) = e^{-\frac{\|\mathbf{s}\|^2}{\beta}}$ and as basis function the Gaussian function (Poggio and Girosi, 1989; Yuille and Grzywacz, 1988). The Gaussian function is positive definite, and it is well known from the theory of reproducing kernels (Aronszajn, 1950) that positive definite functions (Stewart, 1976) can be used to define *norms* of the type (18). Since $\phi[f]$ is a norm, its null space contains only the zero element, and the additional null space terms of equation (3) are not needed, unlike in Duchon splines. A disadvantage of the Gaussian is the appearance of the scaling parameter β, while Duchon splines, being homogeneous functions, do not depend on any scaling parameter. However, it is possible to devise good heuristics that furnish sub-optimal, but still good, values of β, or good starting points for cross-validation procedures.

Other Basis Functions

Here we give a list of other functions that can be used as basis functions in the Radial Basis Functions technique, and that are therefore associated with the minimization of some functional. In the following table we indicate as "p.d." the positive definite functions, which do not need any polynomial term in the solution, and as "c.p.d. k" the conditionally positive definite functions of order k, which need a polynomial of degree k in the solution. It is a well known fact that positive definite functions tend to zero at infinity whereas conditionally positive functions tend to infinity.

$G(r) = e^{-\beta r^2}$ Gaussian, p.d.

$G(r) = \sqrt{r^2 + c^2}$ multiquadric, c.p.d. 1

$G(r) = \frac{1}{\sqrt{c^2+r^2}}$ inverse multiquadric, p.d.

$G(r) = r^{2n+1}$ thin plate splines, c.p.d. n

$G(r) = r^{2n} \ln r$ thin plate splines, c.p.d. n

3.2 Tensor product stabilizers

An alternative to choosing a radial function $\tilde{G}(\mathbf{s})$ in the stabilizer (18) is a *tensor product* type of basis function, that is a function of the form

$$\tilde{G}(\mathbf{s}) = \Pi_{j=1}^{d} \tilde{g}(s_j) \tag{20}$$

where s_j is the j-th coordinate of the vector $\mathbf{s}$, and $\tilde{g}$ is an appropriate one-dimensional function. When g is positive definite the functional $\phi[f]$ is clearly a norm and its null space is empty. In the case of a conditionally positive definite function the structure of the null space can be more complicated and we do not consider it here. Stabilizers with $\tilde{G}(\mathbf{s})$ as in equation (20) have the form

$$\phi[f] = \int_{R^d} d\mathbf{s} \; \frac{|\tilde{f}(\mathbf{s})|^2}{\Pi_{j=1}^{d} \tilde{g}(s_j)}$$

which leads to a *tensor product* basis function

$$G(\mathbf{x}) = \Pi_{j=1}^{d} g(x_j)$$

where x_j is the j-th coordinate of the vector $\mathbf{x}$ and $g(x)$ is the Fourier transform of $\tilde{g}(s)$. An interesting example is the one corresponding to the choice:

$$\tilde{g}(s) = \frac{1}{1 + s^2} \; ,$$

which leads to the basis function:

$$G(\mathbf{x}) = \Pi_{j=1}^{d} e^{-|x_j|} = e^{-\sum_{j=1}^{d} |x_j|} = e^{-\|\mathbf{x}\|_{L_1}} \; .$$

This basis function is interesting from the point of view of VLSI implementations, because it requires the computation of the L_1 norm of the input vector $\mathbf{x}$, which is usually easier to compute than the Euclidean norm L_2. However, this basis function is not very

smooth, and its performance in practical cases should first be tested experimentally. Notice that if the approximation is needed for computing derivatives smoothness of an appropriate degree is clearly a necessary requirement (see Poggio, Vorhees and Yuille, 1988).

We notice that the choice

$$\tilde{g}(s) = e^{-s^2}$$

leads again to the Gaussian basis function $G(\mathbf{x}) = e^{-\|\mathbf{x}\|^2}$.

3.3 Additive stabilizers

We have seen in the previous section how some tensor product approximation schemes can be derived in the framework of regularization theory. We now will see that it is also possible to derive the class of *additive approximation* schemes in the same framework, where by additive approximation we mean an approximation of the form

$$f(\mathbf{x}) = \sum_{\mu=1}^{d} f_\mu(x^\mu) \tag{21}$$

where x^μ is the μ-th component of the input vector $\mathbf{x}$ and the f_μ are one-dimensional functions that will be defined as the *additive components* of f (from now on Greek letter indices will be used in association with components of the input vectors). Additive models are well known in statistics (Hastie and Tibshirani, 1986, 1987, 1990; Stone, 1985; Wahba, 1990; Buja, Hastie and Tibshirani 1989) and can be considered as a generalization of linear models. They are appealing because, being essentially a superposition of one-dimensional functions, they have a low complexity, and they share with linear models the feature that the effects of the different variables can be examined separately.

The simplest way to obtain such an approximation scheme is to choose, if possible, a stabilizer that corresponds to an additive basis function:

$$G(\mathbf{x}) = \sum_{\mu=1}^{n} \theta_\mu g(x^\mu) \tag{22}$$

where θ_μ are certain fixed parameters. Such a choice would lead to an approximation scheme of the form (21) in which the additive components f_μ have the form:

$$f_\mu(x) = \theta_\mu \sum_{i=1}^{N} c_i G(x^\mu - x_i^\mu) \tag{23}$$

Notice that the additive components are not independent at this stage, since there is only one set of coefficients c_i. We postpone the discussion of this point to section (4.2).

We would like then to write stabilizers corresponding to the basis function (22) in the form (18), where $\tilde{G}(\mathbf{s})$ is the Fourier transform of $G(\mathbf{x})$. We notice that the Fourier transform of an additive function like the one in equation (22) exists only in the generalized sense (Gelfand and Shilov, 1964), involving the δ distribution. For example, in two dimensions we obtain

$$\tilde{G}(\mathbf{s}) = \theta_x \tilde{g}(s_x)\delta(s_y) + \theta_y \tilde{g}(s_y)\delta(s_x) \tag{24}$$

and the interpretation of the reciprocal of this expression is delicate. However, *almost* additive basis functions can be obtained if we approximate the delta functions in equation (24) with Gaussians of very small variance. Consider, for example in two dimensions, the stabilizer:

$$\phi[f] = \int_{R^d} d\mathbf{s}\ \epsilon \frac{|\tilde{f}(\mathbf{s})|^2}{\theta_x \tilde{g}(s_x) e^{-(\frac{s_y}{\epsilon})^2} + \theta_y \tilde{g}(s_y) e^{-(\frac{s_x}{\epsilon})^2}} \tag{25}$$

This corresponds to a basis function of the form:

$$G(x,y) = \theta_x g(x) e^{-\epsilon^2 y^2} + \theta_y g(y) e^{-\epsilon^2 x^2}\ . \tag{26}$$

In the limit of ϵ going to zero the denominator in expression (25) approaches equation (24), and the basis function (26) approaches a basis function that is the sum of one-dimensional basis functions. In this paper we do not discuss this limit process in a rigorous way. Instead we outline another way to obtain additive approximations in the framework of regularization theory.

Let us assume that we know *a priori* that the function f that we want to approximate is additive, that is:

$$f(\mathbf{x}) = \sum_{\mu=1}^{d} f_\mu(x^\mu)$$

We then apply the regularization approach and impose a smoothness constraint, not on the function f as a whole, but on each single additive component, through a regularization functional of the form (Wahba, 1990; Hastie and Tibshirani, 1990):

$$H[f] = \sum_{i=1}^{N} (y_i - \sum_{\mu=1}^{d} f_\mu(x_i^\mu))^2 + \lambda \sum_{\mu=1}^{d} \frac{1}{\theta_\mu} \int_R ds\ \frac{|\tilde{f}_\mu(s)|^2}{\tilde{g}(s)}$$

where θ_μ are given positive parameters which allow us to impose different degrees of smoothness on the different additive components. The minimizer of this functional is found with the same technique described in appendix (A), and, skipping null space terms, it has the usual form

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i G(\mathbf{x} - \mathbf{x}_i) \tag{27}$$

where

$$G(\mathbf{x} - \mathbf{x}_i) = \sum_{\mu=1}^{d} \theta_\mu g(x^\mu - x_i^\mu) \ ,$$

as in equation (22).

We notice that the additive component of equation (27) can be written as

$$f_\mu(x^\mu) = \sum_{i=1}^{N} c_i^\mu g(x^\mu - x_i^\mu)$$

where we have defined

$$c_i^\mu = \frac{c_i}{\theta_\mu} \ .$$

The additive components are therefore not independent because the parameters θ_μ are fixed. If the θ_μ were free parameters, the coefficients c_i^μ would be independent, as well as the additive components.

Notice that the two ways we have outlined for deriving additive approximation from regularization theory are equivalent. They both start from *a priori* assumptions of additivity and smoothness of the class of functions to be approximated. In the first technique the two assumptions are woven together in the choice of the stabilizer (equation 25); in the second they are made explicit and exploited sequentially.

4 Extensions: from Regularization Networks to Generalized Regularization Networks

In this section we will first review some extensions of regularization networks, and then will apply them to Radial Basis Functions and to additive splines.

A fundamental problem in almost all practical applications in learning and pattern recognition is the choice of the relevant input variables. It may happen that some of the variables are more relevant than others, that some variables are just totally irrelevant, or that the relevant variables are linear combinations of the original ones. It can therefore be useful to work not with the original set of variables $\mathbf{x}$, but with a linear transformation of them, $\mathbf{Wx}$, where $\mathbf{W}$ is a possibily rectangular matrix. In the framework of regularization theory, this can be taken into account by making the assumption that

the approximating function f has the form $f(\mathbf{x}) = F(\mathbf{W}\mathbf{x})$ for some smooth function F. The smoothness assumption is now made directly on F, through a smoothness functional $\phi[F]$ of the form (18). The regularization functional is expressed in terms of F as

$$H[F] = \sum_{i=1}^{N}(y_i - F(\mathbf{z}_i))^2 + \lambda\phi[F]$$

where $\mathbf{z}_i = \mathbf{W}\mathbf{x}_i$. The function that minimizes this functional is clearly, accordingly to the results of section (2), of the form:

$$F(\mathbf{z}) = \sum_{i=1}^{N} c_i G(\mathbf{z} - \mathbf{z}_i) \ .$$

(plus eventually a polynomial in $\mathbf{z}$). Therefore the solution for f is:

$$f(\mathbf{x}) = F(\mathbf{W}\mathbf{x}) = \sum_{i=1}^{N} c_i G(\mathbf{W}\mathbf{x} - \mathbf{W}\mathbf{x}_i) \tag{28}$$

This argument is rigorous for given and known $\mathbf{W}$, as in the case of classical Radial Basis Functions. Usually the matrix $\mathbf{W}$ is unknown, and it must be estimated from the examples. Estimating both the coefficients c_i and the matrix $\mathbf{W}$ by least squares is usually not a good idea, since we would end up trying to estimate a number of parameters that is larger than the number of data points (though one may use regularized least squares). Therefore, it has been proposed (Moody and Darken, 1988, 1989; Broomhead and Lowe, 1988; Poggio and Girosi, 1989, 1990) to replace the approximation scheme of equation (28) with a similar one, in which the basic shape of the approximation scheme is retained, but the number of basis functions is decreased. The resulting approximating function that we call the *Generalized Regularization Network* (GRN) is:

$$f(\mathbf{x}) = \sum_{\alpha=1}^{n} c_\alpha G(\mathbf{W}\mathbf{x} - \mathbf{W}\mathbf{t}_\alpha) \ . \tag{29}$$

where $n < N$ and the *centers* $\mathbf{t}_\alpha$ are chosen according to some heuristic, or are considered as free parameters (Moody and Darken, 1988, 1989; Poggio and Girosi, 1989, 1990). The coefficients c_α, the elements of the matrix $\mathbf{W}$, and eventually the centers $\mathbf{t}_\alpha$, are estimated according to a least squares criterion. The elements of the matrix $\mathbf{W}$ could also be estimated through cross-validation (Allen, 1974; Wahba and Wold, 1975; Golub, Heath and Wahba, 1979; Craven and Wahba, 1979; Utreras, 1979; Wahba, 1985), which may be a formally more appropriate technique.

In the special case in which the matrix $\mathbf{W}$ and the centers are kept fixed, the resulting technique is one originally proposed by Broomhead and Lowe (1988), and the coefficients satisfy the following linear equation:

$$G^T G\mathbf{c} = G^T\mathbf{y} \ ,$$

where we have defined the following vectors and matrices:

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

This technique, which has become quite common in the neural network community, has the advantage of retaining the form of the regularization solution, while being less complex to compute. A complete theoretical analysis has not yet been given, but some results, in the case in which the matrix $\mathbf{W}$ is set to identity, are already available (Sivakumar and Ward, 1991; Poggio and Girosi, 1989).

The next sections discuss approximation schemes of the form (29) in the cases of radial and additive basis functions.

4.1 Extensions of Radial Basis Functions

In the case in which the basis function is radial, the approximation scheme of equation (29) becomes:

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

where we have defined the weighted norm:

$$\|\mathbf{x}\|_\mathbf{W} \equiv \mathbf{x}\mathbf{W}^T\mathbf{W}\mathbf{x} \ . \tag{30}$$

The basis functions of equation (29) are not radial anymore, or, more precisely, they are radial in the metric defined by equation (30). This means that the level curves of the basis functions are not circles, but ellipses, whose axis do not need to be aligned with the coordinate axis. Notice that in this case what is important is not the matrix $\mathbf{W}$ itself, but rather the symmetric matrix $\mathbf{W}^T\mathbf{W}$. Therefore, by the Cholesky decomposition, it is sufficient to consider $\mathbf{W}$ to be upper triangular. The optimal center locations $\mathbf{t}_\alpha$ satisfy the following set of nonlinear equations (Poggio and Girosi, 1990, 1990a):

$$\mathbf{t}_\alpha = \frac{\sum_i P_i^\alpha \mathbf{x}_i}{\sum_i P_i^\alpha} \quad \alpha = 1, \ldots, n \tag{31}$$

where P_i^α are coefficients that depend on all the parameters of the network and are not necessarily positive. The optimal centers are then a weighted sum of the example points. Thus in some cases it may be more efficient to "move" the coefficients P_i^α rather than the components of $\mathbf{t}_\alpha$ (for instance when the dimensionality of the inputs is high relative to the number of data points).

The approximation scheme defined by equation (29) has been discussed in detail in Poggio and Girosi (1990) and Girosi(1992), so we will not discuss it further. In the next section we will consider its analogue in the case of additive basis functions.

4.2 Extensions of additive splines

In the previous sections we have seen an extension of the classical regularization technique. In this section we derive the form that this extension takes when applied to additive splines. The resulting scheme is very similar to Projection Pursuit Regression (Friedman and Stuezle, 1981; Huber, 1985; Diaconis and Freedman, 1984; Donoho and Johnstone, 1989; Moody and Yarvin, 1991).

We start from the "classical" additive spline, derived from regularization in section (3.3):

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i \sum_{\mu=1}^{d} \theta_\mu G(x^\mu - x_i^\mu) \tag{32}$$

In this scheme the smoothing parameters θ_μ should be known, or can be estimated by cross-validation. An alternative to cross-validation is to consider the parameters θ_μ as *free parameters*, and estimate them with a least square technique together with the coefficients c_i. If the parameters θ_μ are free, the approximation scheme of equation (32) becomes the following:

$$f(\mathbf{x}) = \sum_{i=1}^{N} \sum_{\mu=1}^{d} c_i^\mu G(x^\mu - x_i^\mu)$$

where the coefficients c_i^μ are now independent. Of course, now we must estimate $N \times d$ coefficients instead of just N, and we are likely to encounter an overfitting problem. We then adopt the same idea presented in section (4), and consider an approximation scheme of the form

$$f(\mathbf{x}) = \sum_{\alpha=1}^{n} \sum_{\mu=1}^{d} c_\alpha^\mu G(x^\mu - t_\alpha^\mu) \ , \tag{33}$$

in which the number of centers is smaller than the number of examples, reducing the number of coefficients that must be estimated. We notice that equation (33) can be written as

$$f(\mathbf{x}) = \sum_{\mu=1}^{d} f_\mu(x^\mu)$$

where each additive component has the form:

$$f_\mu(x^\mu) = \sum_{\alpha=1}^{n} c_\alpha^\mu G(x^\mu - t_\alpha^\mu) \ .$$

Therefore another advantage of this technique is that the *additive components are now independent*, each of them being a one-dimensional Radial Basis Functions.

We can now use the same argument from section (4) to introduce a linear transformation of the inputs $\mathbf{x} \to \mathbf{W}\mathbf{x}$, where $\mathbf{W}$ is a $d' \times d$ matrix. Calling $\mathbf{w}_\mu$ the μ-th row of W, and performing the substitution $\mathbf{x} \to \mathbf{W}\mathbf{x}$ in equation (33), we obtain

$$f(\mathbf{x}) = \sum_{\alpha=1}^{n} \sum_{\mu=1}^{d'} c_\alpha^\mu G(\mathbf{w}_\mu \cdot \mathbf{x} - t_\alpha^\mu) \ . \tag{34}$$

We now define the following one-dimensional function:

$$h_\mu(y) = \sum_{\alpha=1}^{n} c_\alpha^\mu G(y - t_\alpha^\mu)$$

and rewrite the approximation scheme of equation (34) as

$$f(\mathbf{x}) = \sum_{\mu=1}^{d'} h_\mu(\mathbf{w}_\mu \cdot \mathbf{x}) \ . \tag{35}$$

Notice the similarity between equation (35) and the Projection Pursuit Regression technique: in both schemes the unknown function is approximated by a linear superposition of one-dimensional variables, which are projections of the original variables on certain vectors that have been estimated. In Projection Pursuit Regression the choice of the functions $h_\mu(y)$ is left to the user. In our case the h_μ are one-dimensional Radial Basis Functions, for example cubic splines, or Gaussians. The choice depends, strictly speaking, on the specific prior, that is, on the specific smoothness assumptions made by the user. Interestingly, in many applications of Projection Pursuit Regression the functions h_μ have been indeed chosen to be cubic splines but other choices are Flexible Fourier Series, rational approximations and orthogonal polynomials (see Moody and Yarvin, 1991).

Let us briefly review the steps that bring us from the classical additive approximation scheme of equation (23) to a Projection Pursuit Regression-like type of approximation:

1. the regularization parameters θ_μ of the classical approximation scheme (23) are considered as free parameters;

2. the number of centers is chosen to be smaller than the number of data points;

3. the true relevant variables are assumed to be some unknown linear combination of the original variables;

We notice that in the extreme case in which each additive component has just one center ($n = 1$), the approximation scheme of equation (34) becomes:

$$f(\mathbf{x}) = \sum_{\mu=1}^{d'} c^{\mu} G(\mathbf{w}_{\mu} \cdot \mathbf{x} - t^{\mu}) \ . \tag{36}$$

When the basis function G is a Gaussian we call – somewhat improperly – a network of this type a *Gaussian Multilayer Perceptron (MLP) Network*, because if G were a threshold function sigmoidal function this would be a Multilayer Perceptron with one layer of hidden units. The sigmoidal functions, typically used instead of the threshold, cannot be derived directly from regularization theory because it is not symmetric, but we will see in section (6) the relationship between a sigmoidal function and the absolute value function, which is a basis function that can be derived from regularization.

There are a number of computational issues related to how to find the parameters of an approximation scheme like the one of equation (34), but we do not discuss them here. We present instead, in section (7), some experimental results, and will describe the algorithm used to obtain them.

5 The Bayesian interpretation of Generalized Regularization Networks

It is well known that a variational principle such as equation (1) can be derived not only in the context of functional analysis (Tikhonov and Arsenin, 1977), but also in a probabilistic framework (Kimeldorf and Wahba, 1971; Wahba, 1980, 1990; Poggio, Torre and Koch, 1985; Marroquin, Mitter and Poggio, 1987; Bertero, Poggio and Torre, 1988). In this section we illustrate this connection informally, without addressing the related mathematical issues.

Suppose that the set $g = \{(\mathbf{x}_i, y_i) \in R^n \times R\}_{i=1}^N$ of data has been obtained by random sampling a function f, defined on R^n, in the presence of noise, that is

$$f(\mathbf{x}_i) = y_i + \epsilon_i, \quad i = 1, \ldots, N \tag{37}$$

where ϵ_i are random independent variables with a given distribution. We are interested in recovering the function f, or an estimate of it, from the set of data g. We take a probabilistic approach, and regard the function f as the realization of a random field with a known prior probability distribution. Let us define:

– $\mathcal{P}[f|g]$ as the conditional probability of the function f given the examples g.

– $\mathcal{P}[g|f]$ as the conditional probability of g given f. If the function underlying the data is f, this is the probability that by random sampling the function f at the sites $\{\mathbf{x}_i\}_{i=1}^N$ the set of measurement $\{y_i\}_{i=1}^N$ is obtained. This is therefore a model of the noise.

– $\mathcal{P}[f]$: is the *a priori* probability of the random field f. This embodies our *a priori* knowledge of the function, and can be used to impose constraints on the model, assigning significant probability only to those functions that satisfy those constraints.

Assuming that the probability distributions $\mathcal{P}[g|f]$ and $\mathcal{P}[f]$ are known, the posterior distribution $\mathcal{P}[f|g]$ can now be computed by applying the Bayes rule:

$$\mathcal{P}[f|g] \propto \mathcal{P}[g|f]\ \mathcal{P}[f]. \tag{38}$$

We now make the assumption that the noise variables in equation (37) are normally distributed, with variance σ. Therefore the probability $\mathcal{P}[g|f]$ can be written as:

$$\mathcal{P}[g|f] \propto e^{-\frac{1}{2\sigma^2}\sum_{i=1}^N (y_i - f(\mathbf{x}_i))^2}$$

where σ is the variance of the noise.

The model for the prior probability distribution $\mathcal{P}[f]$ is chosen in analogy with the discrete case (when the function f is defined on a finite subset of a n-dimensional lattice) for which the problem can be formalized (see for instance Marroquin, Mitter and Poggio, 1987). The prior probability $\mathcal{P}[f]$ is written as

$$\mathcal{P}[f] \propto e^{-\alpha\phi[f]} \tag{39}$$

where $\phi[f]$ is a smoothness functional of the type described in section (3) and α a positive real number. This form of probability distribution gives high probability only to those functions for which the term $\phi[f]$ is small, and embodies the *a priori* knowledge that one has about the system.

Following the Bayes rule (38) the *a posteriori* probability of f is written as

$$\mathcal{P}[f|g] \propto e^{-\frac{1}{2\sigma^2}[\sum_{i=1}^N (y_i - f(\mathbf{x}_i))^2 + 2\alpha\sigma^2\phi[f]]}. \tag{40}$$

One simple estimate of the function f from the probability distribution (40) is the so called MAP (*Maximum A Posteriori*) estimate, that considers the function that maximizes the *a posteriori* probability $\mathcal{P}[f|g]$, and therefore minimizes the exponent in equation (40). The MAP estimate of f is therefore the minimizer of the following functional:

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

where $\lambda = 2\sigma^2\alpha$. This functional is the same as that of equation (1), and from here it is clear that the parameter λ, that is usually called the "regularization parameter" determines the trade-off between the level of the noise and the strength of the *a priori* assumptions about the solution, therefore controlling the compromise between the degree of smoothness of the solution and its closeness to the data. Notice that functionals of the type (39) are common in statistical physics (Parisi, 1988), where $\phi[f]$ plays the role of an energy functional. It is interesting to notice that, in that case, the correlation function of the physical system described by $\phi[f]$ is the basis function $G(\mathbf{x})$.

As we have pointed out (Poggio and Girosi, 1989; Rivest, pers. comm.), prior probabilities can also be seen as a measure of complexity, assigning high complexity to the functions with small probability. It has been proposed by Rissanen (1978) to measure the complexity of a hypothesis in terms of the bit length needed to encode it. It turns out that the MAP estimate mentioned above is closely related to the Minimum Description Length Principle: the hypothesis f which for given g can be described in the most compact way is chosen as the "best" hypothesis. Similar ideas have been explored by others (for instance Solomonoff,1978). They connect data compression and coding with Bayesian inference, regularization, function approximation and learning.

6 Additive splines, hinge functions, sigmoidal neural nets

In the previous sections we have shown how to extend RN to schemes that we have called GRN, which include ridge approximation schemes of the PPR type, that is

$$f(\mathbf{x}) = \sum_{\mu=1}^{d'} h_\mu(\mathbf{w}_\mu \cdot \mathbf{x}) \ ,$$

where

$$h_\mu(y) = \sum_{\alpha=1}^{n} c_\alpha^\mu G(y - t_\alpha^\mu).$$

The form of the basis function G depends on the stabilizer, and a list of "admissible" G has been given in section (3). These include the absolute value $G(x) = |x|$ – corresponding to piecewise linear splines, and the function $G(x) = |x|^3$ – corresponding to cubic splines (used in typical implementations of PPR), as well as Gaussian functions. Though it may seem natural to think that sigmoidal multilayer perceptrons may be included in this framework, it is actually impossible to derive *directly* from regularization principles the sigmoidal activation functions typically used in Multilayer Perceptrons.

In the following section we show, however, that there is a close relationship between basis functions of the hinge, the sigmoid and the Gaussian type.

6.1 From additive splines to ramp and hinge functions

We will consider here the one-dimensional case, since multidimensional additive approximations consist of one-dimensional terms. We consider the approximation with the lowest possible degree of smoothness: piecewise linear. The associated basis function $G(x) = |x|$ is shown in figure 2a, and the associated stabilizer is given by

$$\phi[f] = \int_{-\infty}^{\infty} ds \; s^2 |\tilde{f}(s)|^2$$

This assumption thus leads to approximating a one-dimensional function as the linear combination with appropriate coefficients of translates of $|x|$. It is easy to see that a linear combination of two translates of $|x|$ with appropriate coefficients (positive and negative and equal in absolute value) yields the piecewise linear threshold function $\sigma_L(x)$ also shown in figure 2b. Linear combinations of translates of such functions can be used to approximate one-dimensional functions. A similar derivative-like, linear combination of two translates of $\sigma_L(x)$ functions with appropriate coefficients yields the Gaussian-like function $g_L(x)$ also shown in figure 2c. Linear combinations of translates of this function can also be used for approximation of a function. Thus any given approximation in terms of $g_L(x)$ can be rewritten in terms of $\sigma_L(x)$ and the latter can be in turn expressed in terms of the basis function $|x|$.

Notice that the basis functions $|x|$ underlie the "hinge" technique proposed by Breiman (1993), whereas the basis functions $\sigma_L(x)$ are sigmoidal-like and the $g_L(x)$ are Gaussian-like. The arguments above show the close relations between all of them, despite the fact that only $|x|$ is strictly a "legal" basis function from the point of view of regularization ($g_L(x)$ is not, though the very similar but smoother Gaussian is). Notice also that $|x|$ can be expressed in terms of "ramp" functions, that is $|x| = x_+ + x_-$. Thus a one-hidden layer perceptron using the activation function $\sigma_L(x)$ can be rewritten in terms of a generalized regularization network with basis function $|x|$. The equivalent kernel is effectively local only if there exist a sufficient number of centers for each dimension $(\mathbf{w}_\mu \cdot \mathbf{x})$. This is the case for Projection Pursuit Regression but not for usual one-hidden layer perceptrons.

These relationships imply that it may be interesting to compare how well each of these basis functions is able to approximate some simple function. To do this we used the model $f(x) = \sum_\alpha^n c_\alpha G(w_\alpha x - t_\alpha)$ to approximate the function $h(x) = \sin(2\pi x)$ on $[0, 1]$, where $G(x)$ is one of the basis functions of figure 2. Fifty training points and 10,000 test points were chosen uniformly on $[0, 1]$. The parameters were learned using the iterative backfitting algorithm (Friedman and Stuezle, 1981; Hastie and Tibshirani, 1990;

Breiman, 1993) that will be described in section 7. We looked at the function learned after fitting 1, 2, 4, 8 and 16 basis functions. Some of the resulting approximations are plotted in figure 3.

The results show that the performance of all three basis functions is fairly close as the number of basis functions increases. All models did a good job of approximating $\sin(2\pi x)$. The absolute value function did slightly worse and the "Gaussian" function did slightly better. It is interesting that the approximation using two absolute value functions is almost identical to the approximation using one "sigmoidal" function which again shows that two absolute value basis functions can sum to equal one "sigmoidal" piecewise linear function.

7 Numerical illustrations

7.1 Comparing additive and non-additive models

In order to illustrate some of the ideas presented in this paper and to provide some practical intuition about the various models, we present numerical experiments comparing the performance of additive and non-additive networks on two-dimensional problems. In a model consisting of a sum of two-dimensional Gaussians, the model can be changed from a non-additive Radial Basis Function network to an additive network by "elongating" the Gaussians along the two coordinate axis x and y. This allows us to measure the performance of a network as it changes from a non-additive scheme to an additive one.

Five different models were tested. The first three differ only in the variances of the Gaussian along the two coordinate axis. The ratio of the x variance to the y variance determines the elongation of the Gaussian. These models all have the same form and can be written as:

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i [G_1(\mathbf{x} - \mathbf{x}_i) + G_2(\mathbf{x} - \mathbf{x}_i)]$$

where

$$G_1 = e^{-(\frac{x^2}{\sigma_1} + \frac{y^2}{\sigma_2})} \quad ; \qquad G_2 = e^{-(\frac{x^2}{\sigma_2} + \frac{y^2}{\sigma_1})} \ .$$

The models differ only in the values of σ_1 and σ_2. For the first model, $\sigma_1 = .5$ and $\sigma_2 = .5$ (RBF), for the second model $\sigma_1 = 10$ and $\sigma_2 = .5$ (elliptical Gaussian), and for the third model, $\sigma_1 = \infty$ and $\sigma_2 = .5$ (additive). These models correspond to placing two Gaussians at each data point $\mathbf{x}_i$, with one Gaussian elongated in the x direction and one elongated in the y direction. In the first case (RBF) there is no elongation, in

Model 1	$f(x,y) = \sum_{i=1}^{20} c_i [e^{-\left(\frac{(x-x_i)^2}{\sigma_1} + \frac{(y-y_i)^2}{\sigma_2}\right)} + e^{-\left(\frac{(x-x_i)^2}{\sigma_2} + \frac{(y-y_i)^2}{\sigma_1}\right)}]$	$\sigma_1 = \sigma_2 = 0.5$
Model 2	$f(x,y) = \sum_{i=1}^{20} c_i [e^{-\left(\frac{(x-x_i)^2}{\sigma_1} + \frac{(y-y_i)^2}{\sigma_2}\right)} + e^{-\left(\frac{(x-x_i)^2}{\sigma_2} + \frac{(y-y_i)^2}{\sigma_1}\right)}]$	$\sigma_1 = 10,\ \sigma_2 = 0.5$
Model 3	$f(x,y) = \sum_{i=1}^{20} c_i [e^{-\frac{(x-x_i)^2}{\sigma}} + e^{-\frac{(y-y_i)^2}{\sigma}}]$	$\sigma = 0.5$
Model 4	$f(x,y) = \sum_{\alpha=1}^{n} c_\alpha e^{-(\mathbf{w}_\alpha \cdot \mathbf{x} - t_\alpha)^2}$	–
Model 5	$f(x,y) = \sum_{\alpha=1}^{n} c_\alpha \sigma(\mathbf{w}_\alpha \cdot \mathbf{x} - t_\alpha)$	–

Table 1: *The five models we tested in our numerical experiments.*

the second case (elliptical Gaussian) there is moderate elongation, and in the last case (additive) there is infinite elongation.

The fourth model is a Generalized Regularization Network model, of the form (36), that uses a Gaussian basis function:

$$f(\mathbf{x}) = \sum_{\alpha=1}^{n} c_\alpha e^{-(\mathbf{w}_\alpha \cdot \mathbf{x} - t_\alpha)^2}.$$

In this model, to which we referred earlier as a Gaussian MLP network (equation 36), the weight vectors, centers, and coefficients are all learned.

In order to see how sensitive were the performances to the choice of basis function, we also repeated the experiments for model (4) with a sigmoid (that is *not* a basis function that can be derived from regularization theory) replacing the Gaussian basis function. In our experiments we used the standard sigmoid function:

$$\sigma(x) = \frac{1}{1 + e^{-x}}\ .$$

Models (1) to (5) are summarized in table 1: notice that only model (5) is a Multilayer Perceptron in the standard sense.

In the first three models, the centers were fixed in the learning algorithm and equal to the training examples. The only parameters that were learned were the coefficients c_i, that were computed by solving the linear system of equations (4). The fourth and the fifth model were trained by fitting one basis function at a time according to the following recursive algorithm with backfitting (Friedman and Stuezle, 1981; Hastie and Tibshirani, 1990; Breiman, 1993)

- Add a new basis function;
- Optimize the parameters $\mathbf{w}_\alpha$, t_α and c_α using the "random step" algorithm (Caprile and Girosi, 1990) described below;
- Backfitting: for each basis function α added so far:
  - hold the parameters of all other functions fixed;
  - re-optimize the parameters of function α;
- Repeat the backfitting stage until there is no significant decrease in L_2 error.

The "random step" (Caprile and Girosi, 1990) is a stochastic optimization algorithm that is very simple to implement and that usually finds good local minima. The algorithm works as follows: pick random changes to each parameter such that each random change lies within some interval $[a, b]$. Add the random changes to each parameter and then calculate the new error between the output of the network and the target values. If the error decreases, then keep the changes and double the length of the interval for picking random changes. If the error increases, then throw out the changes and halve the size of the interval. If the length of the interval becomes less than some threshold, then reset the length of the interval to some larger value.

The five models were each tested on two different functions: a two-dimensional additive function:

$$h_{\rm add}(x, y) = \sin(2\pi x) + 4(y - 0.5)^2$$

and the two-dimensional Gabor function:

$$g_{\rm Gabor}(x, y) = e^{-\|\mathbf{x}\|^2} \cos(.75\pi(x + y)).$$

The training data for the functions $h_{\rm add}$ and $g_{\rm Gabor}$ consisted of 20 points picked from a uniform distribution on $[0, 1] \times [0, 1]$ and $[-1, 1] \times [-1, 1]$ respectively. Another 10,000 points were randomly chosen to serve as test data. The results are summarized in table 2 (see Girosi, Jones and Poggio, 1993 for a more extensive description of the results). As expected, the results show that the additive model (3) was able to approximate the additive function, $h_{\rm add}(x, y)$ better than both the RBF model (1) and the elliptical Gaussian model (2), and that there seems to be a smooth degradation of performance as the model changes from the additive to the Radial Basis Function. Just the opposite results are seen in approximating the non-additive Gabor function, $g_{\rm Gabor}(x, y)$, shown in figure 4a. The RBF model (1) did very well, while the additive model (3) did a very poor job, as shown in figure 4b. However, figure 4c shows that the GRN scheme

	Model 1	Model 2	Model 3	Model 4	Model 5
$h_{\mathrm{add}}(x,y)$	train: 0.000036	0.000067	0.000001	0.000170	0.000743
	test: 0.011717	0.001598	0.000007	0.001422	0.026699
$g_{\mathrm{Gabor}}(x,y)$	train: 0.000000	0.000000	0.000000	0.000001	0.000044
	test: 0.003818	0.344881	67.95237	0.033964	0.191055

Table 2: *A summary of the results of our numerical experiments. Each table entry contains the L_2 errors for both the training set and the test set.*

(model 4) gives a fairly good approximation, because the learning algorithm finds better directions for projecting the data than the x and y axis as in the pure additive model.

Notice that the first three models we considered had a number of parameters equal to the number of data points, and were supposed to exactly interpolate the data, so that one may wonder why the training errors are not exactly zero. The reason is the ill-conditioning of the associated linear system, which is a typical problem of Radial Basis Functions (Dyn, Levin and Rippa, 1986).

8 Hardware and biological implementation of network architectures

We have seen that different network architectures can be derived from regularization by making somewhat different assumptions on the classes of functions used for approximation. Given the basic common roots, one is tempted to argue – and numerical experiments support the claim – that there will be small differences in average performance of the various architectures (see also Lippmann 1989, 1991). It becomes therefore interesting to ask which architectures are easier to implement in hardware.

All the schemes that use the same number of centers as examples – such as RBF and additive splines – are expensive in terms of memory requirements (if there are many examples) but have a simple learning stage. More interesting are the schemes that use fewer centers than examples (and use the linear transformation $\mathbf{W}$). There are at least two perspectives for our discussion: we can consider implementation of radial vs. additive schemes and we can consider different activation functions. Let us first discuss radial vs. non-radial functions such as a Gaussian RBF vs. a Gaussian MLP network. For a VLSI implementations, the main difference is in computing a scalar product rather than a L_2 distance, which is usually more expensive both for digital and analog VLSI. The L_2 distance, however, might be replaced with the L_1 distance, that is a sum of absolute values, which can be computed efficiently. Notice that a Radial

Basis Functions scheme that uses the L_1 norm has been derived in section (3.2) from a tensor-product stabilizer.

Let us consider now different activation functions. Activation functions such as Gaussian, sigmoid or absolute values are equally easy to compute, especially if look-up table approaches are used. In analog hardware it is somewhat simpler to generate a sigmoid than a Gaussian, although Gaussian-like shapes can be synthesized with less than 10 transistors (J. Harris, personal communication).

In practical implementations other issues, such as trade-offs between memory and computation and on-chip learning, are likely to be much more relevant than the specific chosen architecture. In other words, a general conclusion about ease of implementation is not possible: none of the architectures we have considered holds a clear edge.

From the point of view of *biological implementations* the situation is somewhat different. The hidden unit in MLP networks with sigmoidal-like activation functions is a plausible, albeit much over-simplified, model of real neurons. The sigmoidal transformation of a scalar product seems much easier to implement in terms of known biophysical mechanisms than the Gaussian of a multidimensional euclidean distance. On the other hand, it is intriguing to observe that HBF centers and tuned cortical neurons behave alike (Poggio and Hurlbert, 1993). In particular, a Gaussian HBF unit is maximally excited when each component of the input exactly matches each component of the center. Thus the unit is optimally tuned to the stimulus value specified by its center. Units with multidimensional centers are tuned to complex features, made of the conjunction of simpler features. This description is very like the customary description of cortical cells optimally tuned to some more or less complex stimulus. So-called place coding is the simplest and most universal example of tuning: cells with roughly bell shaped receptive fields have peak sensitivities for given locations in the input space, and by overlapping, cover all of that space. Thus tuned cortical neurons seem to behave more like Gaussian HBF units than like the sigmoidal units of MLP networks: the tuned response function of cortical neurons mostly resembles $\exp(-\|\mathbf{x}-\mathbf{t}\|^2)$ more than it does $\sigma(\mathbf{x}\cdot\mathbf{w})$. When the stimulus to a cortical neuron is changed from its optimal value in any direction, the neuron's response typically decreases. The activity of a Gaussian HBF unit would also decline with any change in the stimulus away from its optimal value $\mathbf{t}$. For the sigmoid unit, though, certain changes away from the optimal stimulus will not decrease its activity, for example when the input $\mathbf{x}$ is multiplied by a constant $\alpha > 1$.

How might, then, multidimensional Gaussian receptive fields be synthesized from known receptive fields and biophysical mechanisms?

The simplest answer is that cells tuned to complex features may be constructed from a hierarchy of simpler cells tuned to incrementally larger conjunctions of elementary features. This idea – popular among physiologists – can immediately be formalized in terms of Gaussian radial basis functions, since a multidimensional Gaussian function can be decomposed into the product of lower dimensional Gaussians (Ballard, 1986;

Mel, 1988,1990,1992; Poggio and Girosi, 1990). There are several biophysically plausible ways to implement Gaussian RBF-like units (see Poggio and Girosi, 1989 and Poggio, 1990), but none is particularly simple. Ironically one of the plausible implementations of a RBF unit may exploit circuits based on sigmoidal nonlinearities (see Poggio and Hurlbert, 1993). In general, the circuits required for the various schemes described in this paper are reasonable from a biological point of view (Poggio and Girosi, 1989; Poggio, 1990). For example, the normalized basis function scheme of section (2.2) could be implemented as outlined in figure 5 where a "pool" cell summates the activities of all hidden units and shunts the output unit with a shunting inhibition approximating the required division operation.

9 Summary and remarks

A large number of approximation techniques can be written as multilayer networks with one hidden layer. In past papers (Poggio and Girosi, 1989; Poggio and Girosi, 1990, 1990b; Girosi, 1992) we showed how to derive Radial Basis Functions, Hyper Basis Functions and several types of multidimensional splines from regularization principles. We had not used regularization to yield approximation schemes of the additive type (Wahba, 1990; Hastie and Tibshirani, 1990), such as additive splines, ridge approximation of the Projection Pursuit Regression type and hinge functions. In this paper, we show that appropriate stabilizers can be defined to justify such additive schemes, and that the same extensions that leads from RBF to HBF leads from additive splines to ridge function approximation schemes of the Projection Pursuit Regression type. Our Generalized Regularization Networks include, depending on the stabilizer (that is on the prior knowledge on the functions we want to approximate), HBF networks, ridge approximation, tensor products splines and Perceptron-like networks with one hidden layer and appropriate activation functions (such as the Gaussian). Figure 6 shows a diagram of the relationships. Notice that HBF networks and ridge approximation networks are directly related in the special case of normalized inputs (Maruyama, Girosi and Poggio, 1992).

We now feel that a common theoretical framework justifies a large spectrum of approximation schemes in terms of different smoothness constraints imposed within the same regularization functional to solve the ill-posed problem of function approximation from sparse data. The claim is that many different networks and corresponding approximation schemes can be derived from the variational principle

$$H[f] = \sum_{i=1}^{N} (f(\mathbf{x}_i) - y_i)^2 + \lambda \phi[f] \ . \tag{41}$$

They differ because of different choices of stabilizers ϕ, which correspond to different

assumptions of smoothness. In this context, we believe that the Bayesian interpretation is one of the main advantages of regularization: it makes clear that different network architectures correspond to different prior assumptions of smoothness of the functions to be approximated.

The common framework we have derived suggests that differences between the various network architectures are relatively minor, corresponding to different smoothness assumptions. One would expect that each architecture will work best for the class of function defined by the associated prior (that is stabilizer), an expectation which is consistent with numerical results in this paper (see also Donoho and Johnstone, 1989).

9.1 Classification and smoothness

From the point of view of regularization, the task of classification – instead of regression – may seem to represent a problem since the role of smoothness is less obvious. Consider for simplicity binary classification, in which the output y is either 0 or 1 and let $P(\mathbf{x}, y) = P(\mathbf{x})P(y|\mathbf{x})$ be the joint probability of the input-output pairs $(\mathbf{x}, y)$. The average cost associated to an estimator $f(\mathbf{x})$ is the *expected risk* (see section 2.2)

$$I[f] = \int d\mathbf{x}dy \; P(\mathbf{x}, y)(y - f(\mathbf{x}))^2 \; .$$

The problem of learning is now equivalent to minimizing the expected risk based on N samples of the joint probability distribution $P(\mathbf{x}, y)$, and it is usually solved by minimizing the empirical risk (14). Here we discuss two possible approaches to the problem of finding the best estimator:

- If we look for an estimator in the class of real valued functions, it is well known that the minimizer f_0 of $Q[f]$ is the so called *regression function*, that is

$$f_0(x) = \int dy \; yP(y|\mathbf{x}) = P(1|\mathbf{x}) \; . \tag{42}$$

  Therefore, a real valued network f trained on the empirical risk (14) will approximate, under certain conditions of consistency (Vapnik, 1982; Vapnik and Chervonenkis, 1991), the conditional probability distribution of class 1, $P(1|\mathbf{x})$. In this case our final estimator f is real valued, and in order to obtain a binary estimator we have to apply a threshold function to it, so that our final solution turns out to be:

$$f^*(\mathbf{x}) = \theta(f(\mathbf{x}))$$

  where θ is the Heaviside function.

- We could look for an estimator with range $\{0,1\}$, for example of the form $f(\mathbf{x}) = \theta[g(\mathbf{x})]$. In this case the expected risk becomes the average number of misclassified vectors. The function that minimizes the expected risk is not the regression function anymore, but a binary approximation to it.

We argue that in both cases it makes sense to assume that f (and g) is a smooth real-valued function and therefore to use regularization networks to approximate it. The argument is that a natural prior constraint for classification is smoothness of classification boundaries, since otherwise it would be impossible to effectively generalize the correct classification from a set of examples. Furthermore, a condition that usually provides smooth classification boundaries is smoothness of the underlying regressor: a smooth function usually has "smooth" level crossings. Thus both approaches described above suggest to impose smoothness of f or g, that is to approximate f or g with a regularization network.

9.2 Complexity of the approximation problem

So far we have discussed several approximation techniques only from the point of view of the representation and architecture, and we did not discuss how well they perform in approximating functions of different functions spaces. Since these techniques are derived under different *a priori* smoothness assumptions, we clearly expect them to perform optimally when those *a priori* assumptions are satisfied. This makes it difficult to compare their performances, since we expect each technique to work best on a different class of functions. However, if we measure performances by how quickly the approximation error goes to zero when the number of parameters of the approximation scheme goes to infinity, very general results from the theory of linear and nonlinear widths (Timan, 1963; Pinkus, 1986; Lorentz, 1962; 1986; DeVore, Howard and Micchelli, 1989; DeVore, 1991; DeVore and Yu, 1991) suggest that all techniques share the same limitations. For example, when approximating an s times continuously differentiable function in d variables with some function parametrized by n parameters, one can prove that even the "best" nonlinear parametrization cannot achieve an accuracy that is better than the *Jackson type* bound, that is $O(n^{-\frac{s}{d}})$. Here the adjective "best" is used in the sense defined by DeVore, Howard and Micchelli (1989) in their work on nonlinear n-widths, which restricts the sets of nonlinear parametrization to those for which the optimal parameters depend continuously on the function that has to be approximated. Notice that, although this is a desirable property, not all the approximation techniques may have it, and therefore these results may not always be applicable. However, the basic intuition is that a class of functions has an intrinsic complexity that increases exponentially in the the ratio $\frac{d}{s}$, where s is a smoothness index, that is a measure of the amount of constraints imposed on the functions of the class. Therefore, if the smoothness index is kept constant, we expect

that the number of parameters needed in order to achieve a certain accuracy increases exponentially with the number of dimensions, irrespectively of the approximation technique, showing the phenomenon known as "the curse of dimensionality" (Bellman, 1961). Clearly, if we consider classes of functions with a smoothness index that increases when the number of variables increase, then a rate of convergence independent of the dimensionality can be obtained, because the increase in complexity due to the larger number of variables is compensated by the decrease due to the stronger smoothness constraint. In order to make this concept clear, we summarized in table (3) a number of different approximation techniques, and the constraints that can be imposed on them in order to make the approximation error to be $O(\frac{1}{\sqrt{n}})$, that is "indepedent of the dimension", and therefore immune to the curse of dimensionality. Notice that, since these techniques are derived under different *a priori* assumptions, the explicit form of the constraints are different. For example in entries 5 and 6 of the table (Girosi and Anzellotti, 1992, 1993; Girosi, 1993) the result holds in $H^{2m,1}(R^d)$, that is the Sobolev space of functions whose derivatives up to order $2m$ are integrable (Ziemer, 1989). Notice that the number of derivatives that are integrable has to increase with the dimension d in order to keep the rate of convergence constant. A similar phenomenon appears in entries 2 and 3 (Barron, 1991, 1993; Breiman, 1993), but in a less obvious way. In fact, it can be shown (Girosi and Anzellotti, 1992, 1993) that, for example, the space of functions considered by Barron (entry 2) and Breiman (entry 3) are the set of functions that can be written respectively as $f(\mathbf{x}) = \|\mathbf{x}\|^{1-d} * \lambda$ and $f(\mathbf{x}) = \|\mathbf{x}\|^{2-d} * \lambda$, where λ is any function whose Fourier Transform is integrable, and $*$ stands for the convolution operator. Notice that, in this way, it becomes more apparent that these space of functions become more and more constrained as the dimensions increases, due to the more and more rapid fall-off of the terms $\|\mathbf{x}\|^{1-d}$ and $\|\mathbf{x}\|^{2-d}$. The same phenomenon is also very clear in the results of H.N. Mhaskar (1993), who proved that the rate of convergence of approximation of functions with s continuous derivatives by multilayered feedforward neural networks is $O(n^{-\frac{s}{d}})$: if the number of continuous derivatives s increases linearly with the dimension d, the curse of dimensionality disappears, leading to a rate of convergence independent of the dimension.

It is important to emphasize that in practice the parameters of the approximation scheme have to be estimated using a finite amount of data (Vapnik and Chervonenkis, 1971, 1981, 1991; Vapnik, 1982; Pollard, 1984; Geman, Bienenstock and Doursat, 1992; Haussler, 1989; Baum and Haussler, 1989, Baum, 1988; Moody, 1991a, 1991b). In fact, what one does in practice is to minimize the *empirical risk* (see equation 14), while what one would really like to do is to minimize the *expected risk* (see equation 13). This introduces an additional source of error, sometimes called "estimation error", that usually depends on the dimension d in a much milder way than the approximation error, and can be estimated using the theory of uniform convergence of relative frequences to probabilities (Vapnik and Chervonenkis, 1971, 1981, 1991; Vapnik, 1982; Pollard, 1984).

Specific results on the generalization error, that combine both approximation and estimation error, have been obtained by A. Barron (1991, 1994) for sigmoidal neural networks, and by Niyogi and Girosi (1994) for Gaussian Radial Basis Functions. Although these bounds are different, they all have the same qualitative behaviour: for a fixed number of data points the generalization error first decreases when the number of parameters increases, then reaches a minimum and start increasing again, revealing the well known phenomenon of *overfitting*. For a general description of how the approximation and estimation error combine together to bound the generalization error see Niyogi and Girosi (1994).

9.3 Additive structure and the sensory world

In this last section we address the surprising relative success of additive schemes of the ridge approximation type in real world applications. As we have seen, ridge approximation schemes depend on priors that combine additivity of one-dimensional functions with the usual assumption of smoothness. Do such priors capture some fundamental property of the physical world? Consider for example the problem of object recognition, or the problem of motor control. We can recognize almost any object from any of many small subsets of its features, visual and non-visual. We can perform many motor actions in several different ways. In most situations, our sensory and motor worlds are *redundant*. In terms of GRN this means that instead of high-dimensional centers, any of several lower-dimensional centers, that is components, are often sufficient to perform a given task. This means that the "and" of a high-dimensional conjunction can be replaced by the "or" of its components (low-dimensional conjunctions) – a face may be recognized by its eyebrows alone, or a mug by its color. To recognize an object, we may use not only templates comprising all its features, but also subtemplates, comprising subsets of features and in some situations the latter, by themselves, may be fully sufficient. Additive, small centers – in the limit with dimensionality one – with the appropriate $\mathbf{W}$ are of course associated with stabilizers of the additive type.

Splitting the recognizable world into its additive parts may well be preferable to reconstructing it in its full multidimensionality, because a system composed of several independent, additive parts is inherently more robust than a whole simultaneously dependent on each of its parts. The small loss in uniqueness of recognition is easily offset by the gain against noise and occlusion. There is also a possible meta-argument that we mention here only for the sake of curiosity. It may be argued that humans would not be able to understand the world if it were not additive because of the too-large number of necessary examples (because of high dimensionality of any sensory input such as an image). Thus one may be tempted to conjecture that our sensory world is biased towards an "additive structure".

Function space	Norm	Approximation scheme
$\int_{R^d} d\mathbf{s}\ \|\tilde{f}(\mathbf{s})\| < +\infty$ (Jones, 1992)	$L_2(\Omega)$	$f(\mathbf{x}) = \sum_{i=1}^{n} c_i \sin(\mathbf{x} \cdot \mathbf{w}_i + \theta_i)$
$\int_{R^d} d\mathbf{s}\ \|\mathbf{s}\|\|\tilde{f}(\mathbf{s})\| < +\infty$ (Barron, 1991)	$L_2(\Omega)$	$f(\mathbf{x}) = \sum_{i=1}^{n} c_i \sigma(\mathbf{x} \cdot \mathbf{w}_i + \theta_i)$
$\int_{R^d} d\mathbf{s}\ \|\mathbf{s}\|^2\|\tilde{f}(\mathbf{s})\| < +\infty$ (Breiman, 1993)	$L_2(\Omega)$	$f(\mathbf{x}) = \sum_{i=1}^{n} c_i \|\mathbf{x} \cdot \mathbf{w}_i + \theta_i\|_+ + \mathbf{x} \cdot \mathbf{a} + b$
$e^{-\|\mathbf{x}\|^2} * \lambda \quad , \lambda \in L_1(R^d)$ (Girosi and Anzellotti, 1992)	$L_\infty(R^2)$	$f(\mathbf{x}) = \sum_{\alpha=1}^{n} c_\alpha e^{-\|\mathbf{x}-\mathbf{t}_\alpha\|^2}$
$H^{2m,1}(R^d),\ 2m > d$ (Girosi and Anzellotti, 1992)	$L_\infty(R^2)$	$f(\mathbf{x}) = \sum_{\alpha=1}^{n} c_\alpha G_m(\|\mathbf{x} - \mathbf{t}_\alpha\|^2)$
$H^{2m,1}(R^d),\ 2m > d$ (Girosi, 1993)	$L_2(R^2)$	$f(\mathbf{x}) = \sum_{\alpha=1}^{n} c_\alpha e^{-\frac{\|\mathbf{x}-\mathbf{t}_\alpha\|^2}{\sigma_\alpha^2}}$

Table 3: *Approximation schemes and corresponding functions spaces with the same rate of convergence* $O(n^{\frac{1}{2}})$. *The function* σ *in the standard sigmoidal function, the function* $|x|_+$ *in the third entry is the ramp function, and the function* G_m *in the fifth entry is a Bessel potential, that is the Fourier Transform of* $(1+\|\mathbf{s}\|^2)^{-\frac{m}{2}}$ *(Stein, 1970).* $H^{2m,1}(R^d)$ *is the Sobolev space of functions whose derivatives up to order* $2m$ *are integrable (Ziemer, 1989).*

Acknowledgements We are grateful to P. Niyogi, H. Mhaskar, J. Friedman, J. Moody, V. Tresp and one of the (anonymous) referees for useful discussions and suggestions. This paper describes research done within the Center for Biological and Computational Learning in the Department of Brain and Cognitive Sciences and at the Artificial Intelligence Laboratory at MIT. This research is sponsored by grants from the Office of Naval Research under contracts N00014-91-J-0385 and N00014-92-J-1879; by a grant from the National Science Foundation under contract ASC-9217041 (which includes funds from DARPA provided under the HPCC program). Support for the A.I. Laboratory's artificial intelligence research is provided by ONR contract N00014-91-J-4038. Tomaso Poggio is supported by the Uncas and Helen Whitaker Chair at the Whitaker College, Massachusetts Institute of Technology.

APPENDICES

A Derivation of the general form of solution of the regularization problem

We have seen in section (2) that the regularized solution of the approximation problem is the function that minimizes a cost functional of the following form:

$$H[f] = \sum_{i=1}^{N}(y_i - f(\mathbf{x}_i))^2 + \lambda\phi[f] \tag{43}$$

where the smoothness functional $\phi[f]$ is given by

$$\phi[f] = \int_{R^d} d\mathbf{s}\ \frac{|\tilde{f}(\mathbf{s})|^2}{\tilde{G}(\mathbf{s})}\ .$$

The first term measures the distance between the data and the desired solution f, and the second term measures the cost associated with the deviation from smoothness. For a wide class of functionals ϕ the solutions of the minimization problem (43) all have the same form. A detailed and rigorous derivation of the solution of the variational principle associated with equation (43) is outside the scope of this paper. We present here a simple derivation and refer the reader to the current literature for the mathematical details (Wahba, 1990; Madych and Nelson, 1990; Dyn, 1987).

We first notice that, depending on the choice of G, the functional $\phi[f]$ can have a non-empty null space, and therefore there is a certain class of functions that are "invisible" to it. To cope with this problem we first define an equivalence relation among all the functions that differ for an element of the null space of $\phi[f]$. Then we express the first term of $H[f]$ in terms of the Fourier transform of f[3]:

$$f(\mathbf{x}) = \int_{R^d} d\mathbf{s}\ \tilde{f}(\mathbf{s})e^{i\mathbf{x}\cdot\mathbf{s}}$$

obtaining the functional

$$H[\tilde{f}] = \sum_{i=1}^{N}\left(y_i - \int_{R^d} d\mathbf{s}\ \tilde{f}(\mathbf{s})e^{i\mathbf{x}_i\cdot\mathbf{s}}\right)^2 + \lambda\int_{R^d} d\mathbf{s}\ \frac{|\tilde{f}(\mathbf{s})|^2}{\tilde{G}(\mathbf{s})}\ .$$

Then we notice that since f is real, its Fourier transform satisfies the constraint:

$$\tilde{f}^*(\mathbf{s}) = \tilde{f}(-\mathbf{s})$$

[3]For simplicity of notation we take all the constants that appear in the definition of the Fourier transform to be equal to 1.

so that the functional can be rewritten as:

$$H[\tilde{f}] = \sum_{i=1}^{N} \left(y_i - \int_{R^d} d\mathbf{s}\ \tilde{f}(\mathbf{s}) e^{i\mathbf{x}_i \cdot \mathbf{s}} \right)^2 + \lambda \int_{R^d} d\mathbf{s}\ \frac{\tilde{f}(-\mathbf{s})\tilde{f}(\mathbf{s})}{\tilde{G}(\mathbf{s})} \ .$$

In order to find the minimum of this functional we take its functional derivatives with respect to $\tilde{f}$ and set it to zero:

$$\frac{\delta H[\tilde{f}]}{\delta \tilde{f}(\mathbf{t})} = 0 \quad \forall \mathbf{t} \in R^d \ . \tag{44}$$

We now proceed to compute the functional derivatives of the first and second term of $H[\tilde{f}]$. For the first term we have:

$$\begin{aligned}
&\frac{\delta}{\delta \tilde{f}(\mathbf{t})} \sum_{i=1}^{N} \left(y_i - \int_{R^d} d\mathbf{s}\ \tilde{f}(\mathbf{s}) e^{i\mathbf{x}_i \cdot \mathbf{s}} \right)^2 \\
&= 2 \sum_{i=1}^{N} \left(y_i - f(\mathbf{x}_i) \right) \int_{R^d} d\mathbf{s}\ \frac{\delta \tilde{f}(\mathbf{s})}{\delta \tilde{f}(\mathbf{t})} e^{i\mathbf{x}_i \cdot \mathbf{s}} \\
&= 2 \sum_{i=1}^{N} \left(y_i - f(\mathbf{x}_i) \right) \int_{R^d} d\mathbf{s}\ \delta(\mathbf{s} - \mathbf{t}) e^{i\mathbf{x}_i \cdot \mathbf{s}} \\
&= 2 \sum_{i=1}^{N} \left(y_i - f(\mathbf{x}_i) \right) e^{i\mathbf{x}_i \cdot \mathbf{t}}
\end{aligned}$$

For the smoothness functional we have:

$$\begin{aligned}
\frac{\delta}{\delta \tilde{f}(\mathbf{t})} \int_{R^d} d\mathbf{s}\ \frac{\tilde{f}(-\mathbf{s})\tilde{f}(\mathbf{s})}{\tilde{G}(\mathbf{s})} &= 2 \int_{R^d} d\mathbf{s}\ \frac{\tilde{f}(-\mathbf{s})}{\tilde{G}(\mathbf{s})} \frac{\delta \tilde{f}(\mathbf{s})}{\delta \tilde{f}(\mathbf{t})} \\
= 2 \int_{R^d} d\mathbf{s}\ \frac{\tilde{f}(-\mathbf{s})}{\tilde{G}(\mathbf{s})} \delta(\mathbf{s} - \mathbf{t}) &= 2 \frac{\tilde{f}(-\mathbf{t})}{\tilde{G}(\mathbf{t})} \ .
\end{aligned}$$

Using these results we can now write equation (44) as:

$$\sum_{i=1}^{N} (y_i - f(\mathbf{x}_i)) e^{i\mathbf{x}_i \cdot \mathbf{t}} + \lambda \frac{\tilde{f}(-\mathbf{t})}{\tilde{G}(\mathbf{t})} = 0 \ .$$

Changing $\mathbf{t}$ in $-\mathbf{t}$ and multiplying by $\tilde{G}(\mathbf{t})$ on both sides of this equation we get:

$$\tilde{f}(\mathbf{t}) = \tilde{G}(-\mathbf{t}) \sum_{i=1}^{N} \frac{(y_i - f(\mathbf{x}_i))}{\lambda} e^{i\mathbf{x}_i \cdot \mathbf{t}} \ .$$

We now define the coefficients

$$c_i = \frac{(y_i - f(\mathbf{x}_i))}{\lambda} \quad i = 1, \ldots, N \ ,$$

assume that $\tilde{G}$ is symmetric (so that its Fourier transform is real), and take the Fourier transform of the last equation, obtaining:

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i \delta(\mathbf{x}_i - \mathbf{x}) * G(\mathbf{x}) = \sum_{i=1}^{N} c_i G(\mathbf{x} - \mathbf{x}_i) \ .$$

We now recall that we had defined as equivalent all the functions differing by a term that lies in the null space of $\phi[f]$, and therefore the most general solution of the minimization problem is

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i G(\mathbf{x} - \mathbf{x}_i) + p(\mathbf{x})$$

where $p(\mathbf{x})$ is a term that lies in the null space of $\phi[f]$, that is a set of polynomials for most common choices of stabilizer $\phi[f]$.

B Approximation of vector fields with regularization networks

Consider the problem of approximating a q-dimensional vector field $\mathbf{y}(\mathbf{x})$ from a set of sparse data, the examples, which are pairs $(\mathbf{x}_i, \mathbf{y}_i)$ for $i = 1, \ldots, N$. Choose a *Generalized Regularization Network* as the approximation scheme, that is, a network with one "hidden" layer and linear output units. Consider the case of N examples, $n \leq N$ centers, input dimensionality d and output dimensionality q (see figure 7). Then the approximation is

$$\mathbf{y}(\mathbf{x}) = \sum_{\alpha=1}^{n} \mathbf{c}_\alpha G(\mathbf{x} - \mathbf{t}_\alpha) \tag{45}$$

where G is the chosen basis function and the coefficients $\mathbf{c}_\alpha$ are now q-dimensional vectors[4]: $\mathbf{c}_\alpha = (c_\alpha^1, \ldots, c_\alpha^\mu, \ldots, c_\alpha^q)$.
Here we assume, for simplicity, that G is positive definite in order to avoid the need of additional polynomial terms in the previous equation. Equation (45) can be rewritten in matrix notation as

[4]The components of an output vector will be always denoted by superscript, greek indices.

$$\mathbf{y}(\mathbf{x}) = C\mathbf{g}(\mathbf{x}) \tag{46}$$

where the matrix C is defined by $(C)_{\mu,\alpha} = c_\alpha^\mu$ and $\mathbf{g}$ is the vector with elements $(\mathbf{g}(\mathbf{x}))_\alpha = G(\mathbf{x} - \mathbf{t}_\alpha)$. Assuming, for simplicity, that there is no noise in the data (that is equivalent to choosing $\lambda = 0$ in the regularization functional (1)), the equations for the coefficients $\mathbf{c}_\alpha$ can be found imposing the interpolation conditions:

$$\mathbf{y}_i = C\mathbf{g}(\mathbf{x}_i)$$

Introducing the following notation:

$$(Y)_{i,\mu} = y^\mu(\mathbf{x}_i) \quad , \quad (C)_{\mu,\alpha} = c_\alpha^\mu \quad , \quad (G)_{\alpha,i} = G(\mathbf{x}_i - \mathbf{t}_\alpha)$$

the matrix of coefficients C is given by:

$$C = YG^+.$$

where G^+ is the pseudoinverse of G (Penrose, 1955; Albert, 1972). Substituting this expression in equation (46), the following expression is obtained:

$$\mathbf{y}(\mathbf{x}) = YG^+\mathbf{g}(\mathbf{x}) \ .$$

After some algebraic manipulations, this expression can be rewritten as

$$\mathbf{y}(\mathbf{x}) = \sum_{i=1}^{N} b_i(\mathbf{x})\mathbf{y}_i$$

where the functions $b_i(\mathbf{x})$,that are the elements of the vector $\mathbf{b}(\mathbf{x})$, depend on the chosen G, according to

$$\mathbf{b}(\mathbf{x}) = G^+\mathbf{g}(\mathbf{x}).$$

Therefore, it follows (though it is not so well known) that the vector field $\mathbf{y}(\mathbf{x})$ is approximated by the network as the linear combination of the example fields $\mathbf{y}_i$.

Thus *for any choice of the regularization network and any choice of the (positive definite) basis function the estimated output vector is always a linear combination of the output example vectors with coefficients* $\mathbf{b}$ *that depend on the input value.* The result is valid for all networks with one hidden layer and linear outputs, provided that the mean square error criterion is used for training.

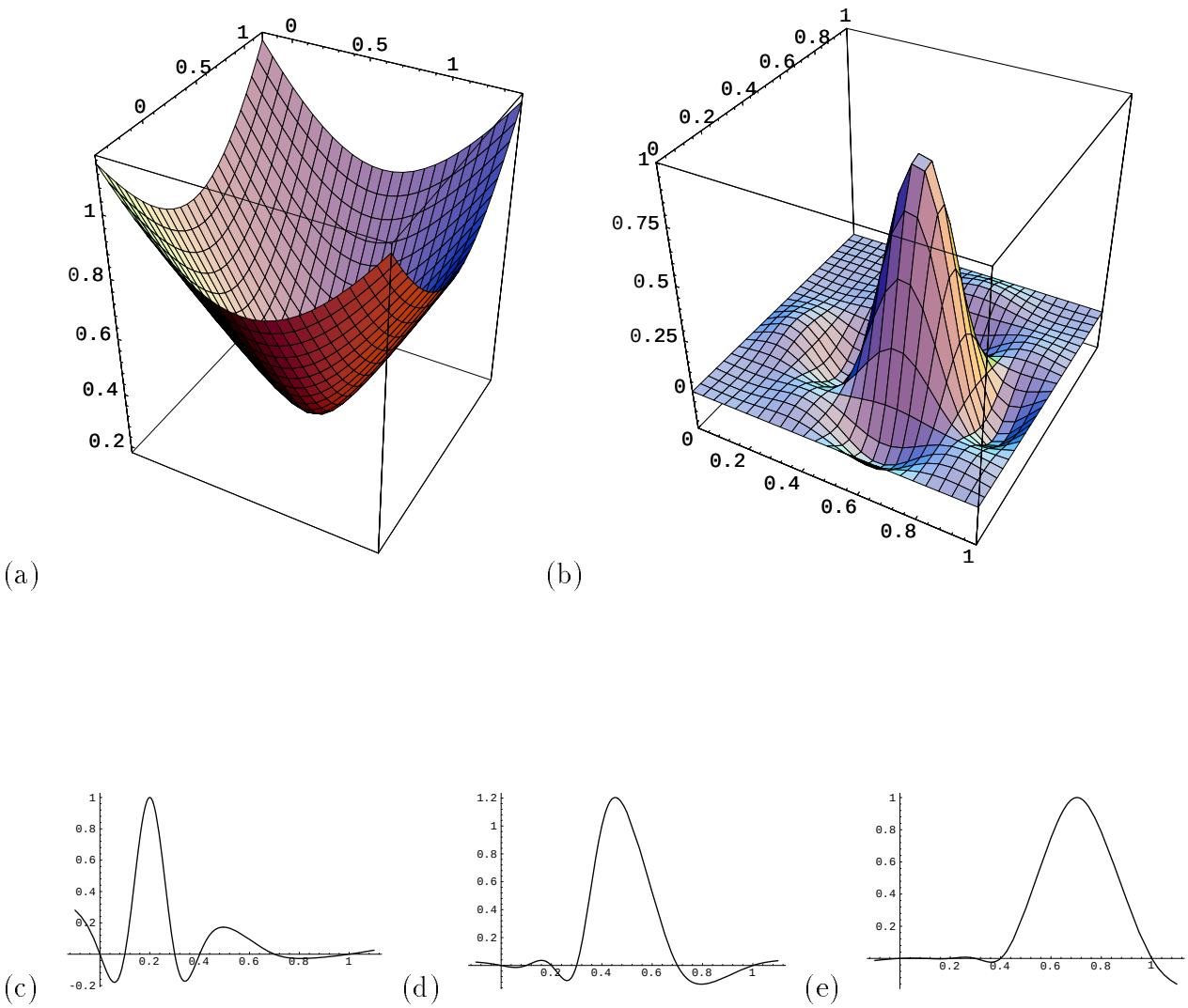

Figure 1: (a) The multiquadric function (b) An equivalent kernel for the multiquadric basis function in the cases of two-dimensional equally spaced data. (c, d, e) The equivalent kernels b_3, b_5 and b_6, for nonuniform one-dimensional multiquadric interpolation (see text for explanation).

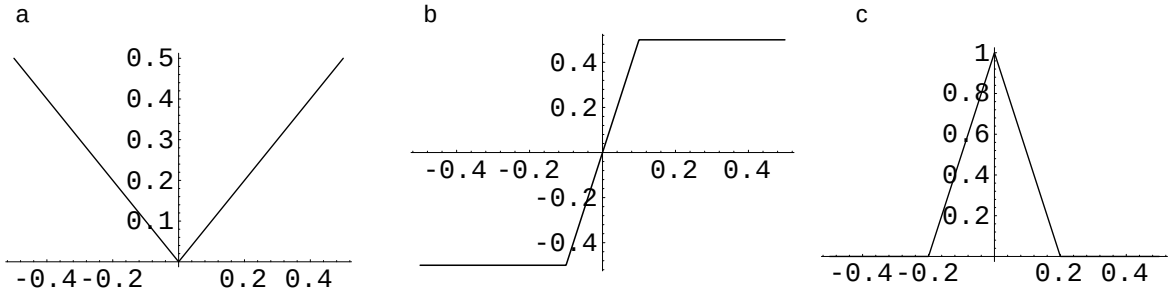

Figure 2: a) Absolute value basis function, $|x|$, b) Sigmoidal-like basis function $\sigma_L(x)$ c) Gaussian-like basis function $g_L(x)$

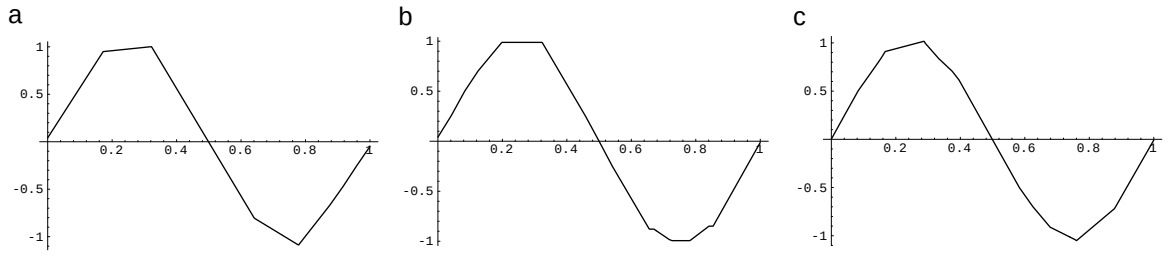

Figure 3: Approximation of $\sin(2\pi x)$ using 8 basis functions of the a) absolute value type, b) Sigmoidal-like type, and c) Gaussian-like type.

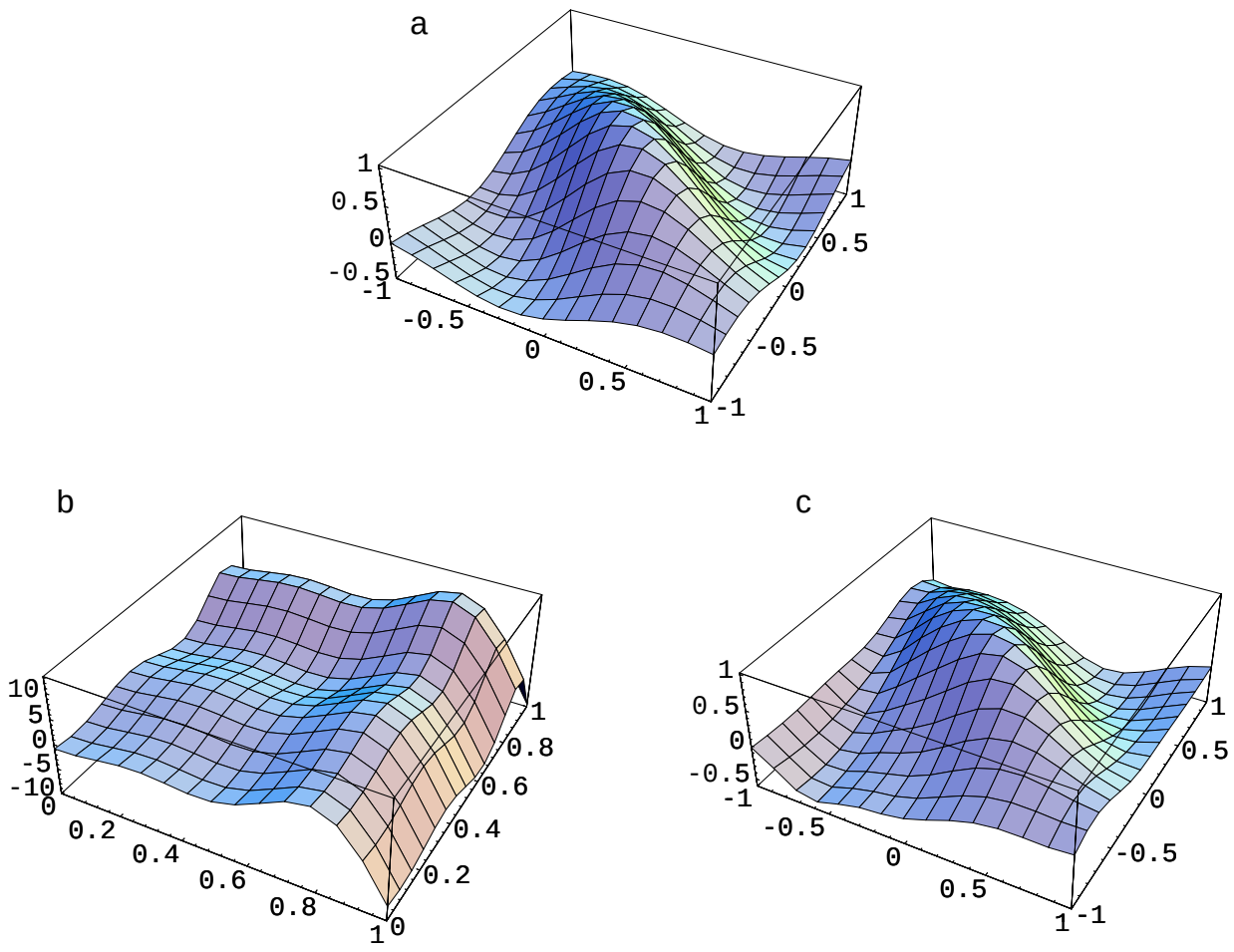

Figure 4: a) The function to be approximated $g(x, y)$. b) Additive Gaussian model approximation of $g(x, y)$ (model 3). c) GRN Approximation of $g(x, y)$ (model 4).

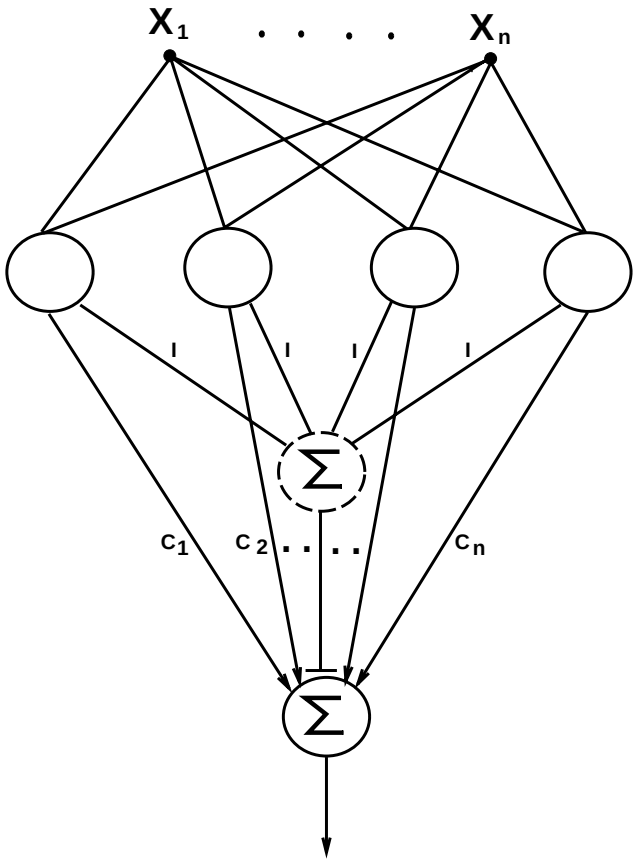

Figure 5: An implementation of the normalized radial basis function scheme. A "pool" cell (dotted circle) summates the activities of the hidden units and then divides the output of the network. The division may be approximated in a physiological implementation by shunting inhibition.

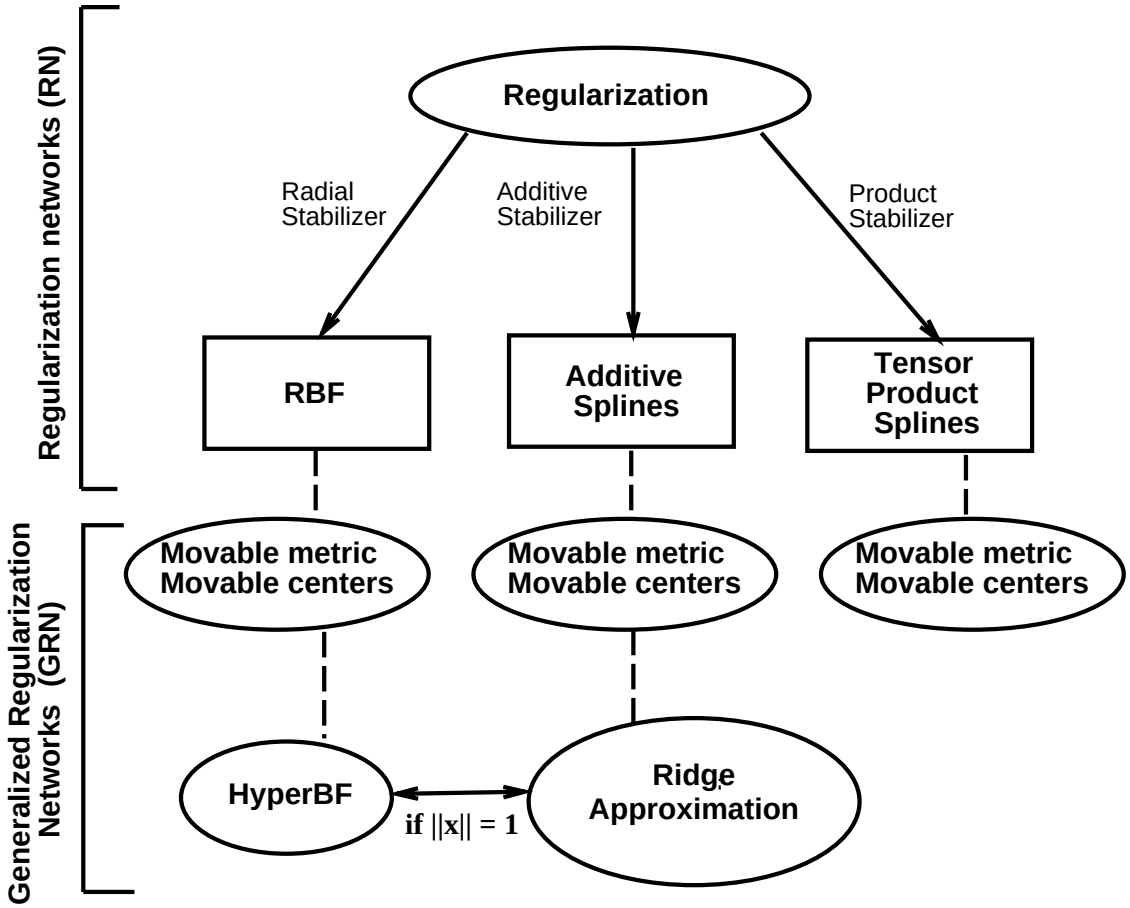

Figure 6: Several classes of approximation schemes and corresponding network architectures can be derived from regularization with the appropriate choice of smoothness priors and associated stabilizers and basis functions, showing the common Bayesian roots.

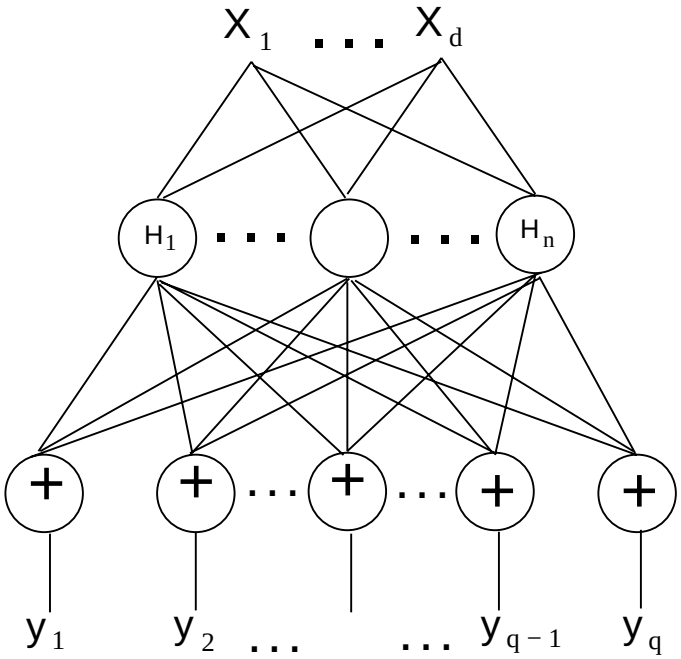

Figure 7: The most general network with one hidden layer and vector output. Notice that this approximation of a q-dimensional vector field has in general fewer parameters than the alternative representation consisting of q networks with one-dimensional outputs. If the only free parameters are the weights from the hidden layer to the output (as for simple RBF with $n = N$, where N is the number of examples) the two representations are equivalent.

References

[1] F.A. Aidu and V.N. Vapnik. Estimation of probability density on the basis of the method of stochastic regularization. *Avtomatika i Telemekhanika*, (4):84–97, April 1989.

[2] A. Albert. *Regression and the Moore-Penrose Pseudoinverse*. Academic Press, New York, 1972.

[3] D. Allen. The relationship between variable selection and data augmentation and a method for prediction. *Technometrics*, 16:125–127, 1974.

[4] N. Aronszajn. Theory of reproducing kernels. *Trans. Amer. Math. Soc.*, 686:337–404, 1950.

[5] D. H. Ballard. Cortical connections and parallel processing: structure and function. *Behavioral and Brain Sciences*, 9:67–120, 1986.

[6] A. R. Barron and Barron R. L. Statistical learning networks: a unifying view. In *Symposium on the Interface: Statistics and Computing Science*, Reston, Virginia, April 1988.

[7] A.R. Barron. Approximation and estimation bounds for artificial neural networks. Technical Report 59, Department of Statistics, University of Illinois at Urbana-Champaign, Champaign, IL, March 1991.

[8] A.R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transaction on Information Theory*, 39(3):930–945, May 1993.

[9] A.R. Barron. Approximation and estimation bounds for artificial neural networks. *Machine Learning*, 14:115–133, 1994.

[10] E. B. Baum. On the capabilities of multilayer perceptrons. *J. Complexity*, 4:193–215, 1988.

[11] E.B. Baum and D. Haussler. What size net gives valid generalization? *Neural Computation*, 1:151–160, 1989.

[12] R.E. Bellman. *Adaptive Control Processes*. Princeton University Press, Princeton, NJ, 1961.

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

[14] M. Bertero, T. Poggio, and V. Torre. Ill-posed problems in early vision. *Proceedings of the IEEE*, 76:869–889, 1988.

[15] L. Bottou and V. Vapnik. Local learning algorithms. *Neural Computation*, 4(6):888–900, November 1992.

[16] L. Breiman. Hinging hyperplanes for regression, classification, and function approximation. *IEEE Transaction on Information Theory*, 39(3):999–1013, May 1993.

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

[18] M.D. Buhmann. Multivariate cardinal interpolation with radial basis functions. *Constructive Approximation*, 6:225–255, 1990.

[19] M.D. Buhmann. On quasi-interpolation with Radial Basis Functions. Numerical Analysis Reports DAMPT 1991/NA3, Department of Applied Mathematics and Theoretical Physics, Cambridge, England, March 1991.

[20] A. Buja, T. Hastie, and R. Tibshirani. Linear smoothers and additive models. *The Annals of Statistics*, 17:453–555, 1989.

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

[22] D.D. Cox. Multivariate smoothing spline functions. *SIAM J. Numer. Anal.*, 21:789–813, 1984.

[23] P. Craven and G. Wahba. Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross validation. *Numer. Math*, 31:377–403, 1979.

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

[25] C. de Boor. *A Practical Guide to Splines*. Springer-Verlag, New York, 1978.

[26] C. de Boor. Quasi-interpolants and approximation power of multivariate splines. In M. Gasca and C.A. Micchelli, editors, *Computation of Curves and Surfaces*, pages 313–345. Kluwer Academic Publishers, Dordrecht, Netherlands, 1990.

[27] R. DeVore, R. Howard, and C. Micchelli. Optimal nonlinear approximation. *Manuskripta Mathematika*, 1989.

[28] R.A. DeVore. Degree of nonlinear approximation. In C.K. Chui, L.L. Schumaker, and D.J. Ward, editors, *Approximation Theory, VI*, pages 175–201. Academic Press, New York, 1991.

[29] R.A. DeVore and X.M. Yu. Nonlinear n-widths in Besov spaces. In C.K. Chui, L.L. Schumaker, and D.J. Ward, editors, *Approximation Theory, VI*, pages 203–206. Academic Press, New York, 1991.

[30] L.P. Devroye and T.J. Wagner. Distribution-free consistency results in nonparametric discrimination and regression function estimation. *The Annals of Statistics*, 8:231–239, 1980.

[31] P. Diaconis and D. Freedman. Asymptotics of graphical projection pursuit. *The Annals of Statistics*, 12(3):793–815, 1984.

[32] D.L. Donoho and I.M. Johnstone. Projection-based approximation and a duality with kernel methods. *The Annals of Statistics*, 17(1):58–106, 1989.

[33] 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.

[34] 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.

[35] 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.

[36] N. Dyn, I.R.H. Jackson, D. Levin, and A. Ron. On multivariate approximation by integer translates of a basis function. Computer Sciences Technical Report 886, University of Wisconsin–Madison, November 1989.

[37] 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.

[38] R.L. Eubank. *Spline Smoothing and Nonparametric Regression*, volume 90 of *Statistics, textbooks and monographs*. Marcel Dekker, Basel, 1988.

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

[40] R. Franke. Recent advances in the approximation of surfaces from scattered data. In C.K. Chui, L.L. Schumaker, and F.I. Utreras, editors, *Topics in multivariate approximation*. Academic Press, New York, 1987.

[41] J.H. Friedman and W. Stuetzle. Projection pursuit regression. *Journal of the American Statistical Association*, 76(376):817–823, 1981.

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

[43] Th. Gasser and H.G. Müller. Estimating regression functions and their derivatives by the kernel method. *Scand. Journ. Statist.*, 11:171–185, 1985.

[44] I.M. Gelfand and G.E. Shilov. *Generalized functions. Vol. 1: Properties and Operations*. Academic Press, New York, 1964.

[45] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4:1–58, 1992.

[46] F. Girosi. Models of noise and robust estimates. A.I. Memo 1287, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1991.

[47] F. Girosi. On some extensions of radial basis functions and their applications in artificial intelligence. *Computers Math. Applic.*, 24(12):61–80, 1992.

[48] F. Girosi. Regularization theory, radial basis functions and networks. In V. Cherkassky, J.H. Friedman, and H. Wechsler, editors, *From Statistics to Neural Networks. Theory and Pattern Recognition Applications*. Springer-Verlag, Subseries F, Computer and Systems Sciences, 1993.

[49] F. Girosi and G. Anzellotti. Rates of convergence of approximation by translates. A.I. Memo 1288, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1992.

[50] F. Girosi and G. Anzellotti. Rates of convergence for radial basis functions and neural networks. In R.J. Mammone, editor, *Artificial Neural Networks for Speech and Vision*, pages 97–113, London, 1993. Chapman & Hall.

[51] F. Girosi, M. Jones, and T. Poggio. Priors, stabilizers and basis functions: From regularization to radial, tensor and additive splines. A.I. Memo No. 1430, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1993.

[52] F. Girosi and T. Poggio. Networks and the best approximation property. *Biological Cybernetics*, 63:169–176, 1990.

[53] F. Girosi, T. Poggio, and B. Caprile. Extensions of a theory of networks for approximation and learning: outliers and negative examples. In R. Lippmann, J. Moody, and D. Touretzky, editors, *Advances in Neural information processings systems 3*, San Mateo, CA, 1991. Morgan Kaufmann Publishers.

[54] G. Golub, M. Heath, and G. Wahba. Generalized cross validation as a method for choosing a good ridge parameter. *Technometrics*, 21:215–224, 1979.

[55] W. E. L. Grimson. A computational theory of visual surface interpolation. *Proceedings of the Royal Society of London B*, 298:395–427, 1982.

[56] R.L. Harder and R.M. Desmarais. Interpolation using surface splines. *J. Aircraft*, 9:189–191, 1972.

[57] W. Härdle. *Applied nonparametric regression*, volume 19 of *Econometric Society Monographs*. Cambridge University Press, 1990.

[58] R.L. Hardy. Multiquadric equations of topography and other irregular surfaces. *J. Geophys. Res*, 76:1905–1915, 1971.

[59] R.L. Hardy. Theory and applications of the multiquadric-biharmonic method. *Computers Math. Applic.*, 19(8/9):163–208, 1990.

[60] T. Hastie and R. Tibshirani. Generalized additive models. *Statistical Science*, 1:297–318, 1986.

[61] T. Hastie and R. Tibshirani. Generalized additive models: some applications. *J. Amer. Statistical Assoc.*, 82:371–386, 1987.

[62] T. Hastie and R. Tibshirani. *Generalized Additive Models*, volume 43 of *Monographs on Statistics and Applied Probability*. Chapman and Hall, London, 1990.

[63] D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. Technical Report UCSC-CRL-91-02, University of California, Santa Cruz, 1989.

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

[65] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

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

[67] A. Hurlbert and T. Poggio. Synthetizing a color algorithm from examples. *Science*, 239:482–485, 1988.

[68] B. Irie and S. Miyake. Capabilities of three-layered Perceptrons. *IEEE International Conference on Neural Networks*, 1:641–648, 1988.

[69] I.R.H. Jackson. *Radial Basis Functions methods for multivariate approximation.* Ph.d. thesis, University of Cambridge, U.K., 1988.

[70] L.K. Jones. A simple lemma on greedy approximation in Hilbert space and convergence rates for Projection Pursuit Regression and neural network training. *The Annals of Statistics*, 20(1):608–613, March 1992.

[71] 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.

[72] 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.

[73] G.S. Kimeldorf and G. Wahba. A correspondence between Bayesan estimation on stochastic processes and smoothing by splines. *Ann. Math. Statist.*, 2:495–502, 1971.

[74] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[75] S.Y. Kung. *Digital Neural Networks.* Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[76] P. Lancaster and K. Salkauskas. *Curve and Surface Fitting.* Academic Press, London, 1986.

[77] A. Lapedes and R. Farber. How neural nets work. In Dana Z. Anderson, editor, *Neural Information Processing Systems*, pages 442–456. Am. Inst. Physics, NY, 1988. Proceedings of the Denver, 1987 Conference.

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

[79] R. P. Lippmann and Y. Lee. A critical overview of neural network pattern classifiers. *Presented at Neural Networks for Computing Conference, Snowbird, UT*, 1991.

[80] G. G. Lorentz. Metric entropy, widths, and superposition of functions. *Amer. Math. Monthly*, 69:469–485, 1962.

[81] G. G. Lorentz. *Approximation of Functions.* Chelsea Publishing Co., New York, 1986.

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

[83] W.R. Madych and S.A. Nelson. Polyharmonic cardinal splines: a minimization property. *Journal of Approximation Theory*, 63:303–320, 1990a.

[84] J. L. Marroquin, S. Mitter, and T. Poggio. Probabilistic solution of ill-posed problems in computational vision. *J. Amer. Stat. Assoc.*, 82:76–89, 1987.

[85] M. Maruyama, F. Girosi, and T. Poggio. A connection between HBF and MLP. A.I. Memo No. 1291, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1992.

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

[87] B. W. Mel. MURPHY: a robot that learns by doing. In D. Z. Anderson, editor, *Neural Information Processing Systems.* American Institute of Physics, University of Colorado, Denver, 1988.

[88] B.W. Mel. The Sigma-Pi column: A model of associative learning in cerebral neocortex. Technical Report 6, California Institute of Technology, 1990.

[89] B.W. Mel. NMDA-based pattern-discrimination in a modeled cortical neuron. *Neural Computation*, 4:502–517, 1992.

[90] H.N. Mhaskar. Approximation properties of a multilayered feedforward artificial neural network. *Advances in Computational Mathematics*, 1:61–80, 1993.

[91] H.N. Mhaskar. Neural networks for localized approximation of real functions. In C.A. Kamm et al., editor, *Neural networks for signal processing III, Proceedings of the 1993 IEEE-SP Workshop*, pages 190–196, New York, 1993a. IEEE Signal Processing Society.

[92] H.N. Mhaskar and C.A. Micchelli. Approximation by superposition of a sigmoidal function. *Advances in Applied Mathematics*, 13:350–373, 1992.

[93] H.N. Mhaskar and C.A. Micchelli. How to choose an activation function. In S. J. Hanson, J. D. Cowan, and C. L. Giles, editors, *Advances in Neural Information Processing Systems 5*. San Mateo, CA: Morgan Kaufmann Publishers, 1993.

[94] C. A. Micchelli. Interpolation of scattered data: distance matrices and conditionally positive definite functions. *Constructive Approximation*, 2:11–22, 1986.

[95] J. Moody. Note on generalization, regularization, and architecture selection in nonlinear learning systems. In *Proceedings of the First IEEE-SP Workshop on Neural Networks for Signal Processing*, pages 1–10, Los Alamitos, CA, 1991a. IEEE Computer Society Press.

[96] J. Moody. The *effective* number of parameters: an analysis of generalization and regularization in nonlinear learning systems. In J. Moody, S. Hanson, and R. Lippmann, editors, *Advances in Neural information processings systems 4*, pages 847–854, Palo Alto, CA, 1991b. Morgan Kaufmann Publishers.

[97] J. Moody and C. Darken. Learning with localized receptive fields. In G. Hinton, T. Sejnowski, and D. Touretzsky, editors, *Proceedings of the 1988 Connectionist Models Summer School*, pages 133–143, Palo Alto, 1988.

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

[99] J. Moody and N. Yarvin. Networks with learned unit response functions. In J. Moody, S. Hanson, and R. Lippmann, editors, *Advances in Neural information processings systems 4*, pages 1048–1055, Palo Alto, CA, 1991. Morgan Kaufmann Publishers.

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

[101] E.A. Nadaraya. On estimating regression. *Theor. Prob. Appl.*, 9:141–142, 1964.

[102] P. Niyogi and F. Girosi. On the relationship between generalization error, hypothesis complexity, and sample complexity for radial basis functions. A.I. Memo 1467, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1994.

[103] S. Omohundro. Efficient algorithms with neural network behaviour. *Complex Systems*, 1:273, 1987.

[104] G. Parisi. *Statistical Field Theory.* Addison-Wesley, Reading, Massachusets, 1988.

[105] E. Parzen. On estimation of a probability density function and mode. *Ann. Math. Statis.*, 33:1065–1076, 1962.

[106] R. Penrose. A generalized inverse for matrices. *Proc. Cambridge Philos. Soc.*, 51:406–413, 1955.

[107] A. Pinkus. *N-widths in Approximation Theory*. Springer-Verlag, New York, 1986.

[108] T. Poggio. On optimal nonlinear associative recall. *Biological Cybernetics*, 19:201–209, 1975.

[109] T. Poggio. A theory of how the brain might work. In *Cold Spring Harbor Symposia on Quantitative Biology*, pages 899–910. Cold Spring Harbor Laboratory Press, 1990.

[110] 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.

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

[112] T. Poggio and F. Girosi. Extension of a theory of networks for approximation and learning: dimensionality reduction and clustering. In *Proceedings Image Understanding Workshop*, pages 597–603, Pittsburgh, Pennsylvania, September 11–13 1990a. Morgan Kaufmann.

[113] T. Poggio and F. Girosi. Regularization algorithms for learning that are equivalent to multilayer networks. *Science*, 247:978–982, 1990b.

[114] T. Poggio and A. Hurlbert. Observation on cortical mechanisms for object recognition and learning. In C. Koch and J. Davis, editors, *Large-scale Neuronal Theories of the Brain*. In press.

[115] T. Poggio, V. Torre, and C. Koch. Computational vision and regularization theory. *Nature*, 317:314–319, 1985.

[116] T. Poggio, H. Voorhees, and A. Yuille. A regularized solution to edge detection. *Journal of Complexity*, 4:106–123, 1988.

[117] D. Pollard. *Convergence of stochastic processes*. Springer-Verlag, Berlin, 1984.

[118] 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.

[119] M.J.D. Powell. The theory of radial basis functions approximation in 1990. In W.A. Light, editor, *Advances in Numerical Analysis Volume II: Wavelets, Subdivision Algorithms and Radial Basis Functions*, pages 105–210. Oxford University Press, 1992.

[120] M.B. Priestley and M.T. Chao. Non-parametric function fitting. *Journal of the Royal Statistical Society B*, 34:385–392, 1972.

[121] C. Rabut. How to build quasi-interpolants. applications to polyharmonic B-splines. In P.-J. Laurent, A. Le Mehautè, and L.L. Schumaker, editors, *Curves and Surfaces*, pages 391–402. Academic Press, New York, 1991.

[122] C. Rabut. An introduction to Schoenberg's approximation. *Computers Math. Applic.*, 24(12):149–175, 1992.

[123] B.D. Ripley. Neural networks and related methods for classification. *Proc. Royal Soc. London*, in press, 1994.

[124] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[125] M. Rosenblatt. Curve estimates. *Ann. Math. Statist.*, 64:1815–1842, 1971.

[126] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(9):533–536, October 1986.

[127] I.J. Schoenberg. Contributions to the problem of approximation of equidistant data by analytic functions, part a: On the problem of smoothing of graduation, a first class of analytic approximation formulae. *Quart. Appl. Math.*, 4:45–99, 1946a.

[128] I.J. Schoenberg. Cardinal interpolation and spline functions. *Journal of Approximation theory*, 2:167–206, 1969.

[129] L.L. Schumaker. *Spline functions: basic theory.* John Wiley and Sons, New York, 1981.

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

[131] B.W. Silverman. Spline smoothing: the equivalent variable kernel method. *The Annals of Statistics*, 12:898–916, 1984.

[132] 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.

[133] R.J. Solomonoff. Complexity-based induction systems: comparison and convergence theorems. *IEEE Transactions on Information Theory*, 24, 1978.

[134] D.F. Specht. A general regression neural network. *IEEE Transactions on Neural Networks*, 2(6):568–576, 1991.

[135] E.M. Stein. *Singular integrals and differentiability properties of functions.* Princeton, N.J., Princeton University Press, 1970.

[136] J. Stewart. Positive definite functions and generalizations, an historical survey. *Rocky Mountain J. Math.*, 6:409–434, 1976.

[137] C. J. Stone. Additive regression and other nonparametric models. *The Annals of Statistics*, 13:689–705, 1985.

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

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

[140] A.F. Timan. *Theory of approximation of functions of a real variable.* Macmillan, New York, 1963.

[141] V. Tresp, J. Hollatz, and S. Ahmad. Network structuring and training using rule-based knowledge. In S. J. Hanson, J. D. Cowan, and C. L. Giles, editors, *Advances in Neural Information Processing Systems 5.* San Mateo, CA: Morgan Kaufmann Publishers, 1993.

[142] F. Utreras. Cross-validation techniques for smoothing spline functions in one or two dimensions. In T. Gasser and M. Rosenblatt, editors, *Smoothing techniques for Curve Estimation*, pages 196–231. Springer-Verlag, Heidelberg, 1979.

[143] V. N. Vapnik. *Estimation of Dependences Based on Empirical Data.* Springer-Verlag, Berlin, 1982.

[144] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequences of events to their probabilities. *Th. Prob. and its Applications*, 17(2):264–280, 1971.

[145] V.N. Vapnik and A. Ya. Chervonenkis. The necessary and sufficient conditions for the uniform convergence of averages to their expected values. *Teoriya Veroyatnostei i Ee Primeneniya*, 26(3):543–564, 1981.

[146] V.N. Vapnik and A. Ya. Chervonenkis. The necessary and sufficient conditions for consistency in the empirical risk minimization method. *Pattern Recognition and Image Analysis*, 1(3):283–305, 1991.

[147] V.N. Vapnik and A.R. Stefanyuk. Nonparametric methods for restoring probability densities. *Avtomatika i Telemekhanika*, (8):38–52, 1978.

[148] G. Wahba. Smoothing noisy data by spline functions. *Numer. Math*, 24:383–393, 1975.

[149] G. Wahba. Smoothing and ill-posed problems. In M. Golberg, editor, *Solutions methods for integral equations and applications*, pages 183–194. Plenum Press, New York, 1979.

[150] G. Wahba. Spline bases, regularization, and generalized cross-validation for solving approximation problems with large quantities of noisy data. In J. Ward and E. Cheney, editors, *Proceedings of the International Conference on Approximation theory in honour of George Lorenz*, Austin, TX, January 8–10 1980. Academic Press.

[151] G. Wahba. A comparison of GCV and GML for choosing the smoothing parameter in the generalized splines smoothing problem. *The Annals of Statistics*, 13:1378–1402, 1985.

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

[153] G. Wahba and S. Wold. A completely automatic french curve. *Commun. Statist.*, 4:1–17, 1975.

[154] G.S. Watson. Smooth regression analysis. *Sankhya A*, 26:359–372, 1964.

[155] H. White. Learning in artificial neural networks: a statistical perspective. *Neural Computation*, 1:425–464, 1989.

[156] H. White. Connectionist nonparametric regression: Multilayer perceptrons can learn arbitrary mappings. *Neural Networks*, 3(535-549), 1990.

[157] A. Yuille and N. Grzywacz. The motion coherence theory. In *Proceedings of the International Conference on Computer Vision*, pages 344–354, Washington, D.C., December 1988. IEEE Computer Society Press.

[158] W.P. Ziemer. *Weakly differentiable functions : Sobolev spaces and functions of bounded variation*. Springer-Verlag, New York, 1989.