MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No.1430 June 1993

C.B.C.L. Paper No. 75

Priors, Stabilizers and Basis Functions: from regularization to radial, tensor and additive splines

Federico Girosi, Michael Jones and Tomaso Poggio

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 we had discussed how standard smoothness functionals lead to a subclass of regularization networks, the well-known Radial Basis Functions approximation schemes. In this paper we show 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 extension that leads from 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 and some forms of Projection Pursuit Regression. 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 the final part of the paper, we show the relation between activation functions of the Gaussian and sigmoidal type by considering the simple case of the kernel $G(x) = |x|$.

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 generalize into Hyper Basis Functions, b) some tensor product splines, and c) additive splines that generalize into schemes of the type of ridge approximation, hinge functions and one-hidden-layer perceptrons.

© Massachusetts Institute of Technology, 1993

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. This research is sponsored by grants from the Office of Naval Research under contracts N00014-91-J-1270 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); and by a grant from the National Institutes of Health under contract NIH 2-S07-RR07047. Additional support is provided by the North Atlantic Treaty Organization, ATR Audio and Visual Perception Research Laboratories, Mitsubishi Electric Corporation, Sumitomo Metal Industries, and Siemens AG. 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.

1 Introduction

In recent papers 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 viceversa, has also been extensively discussed (Barron and Barron, 1988; Poggio and Girosi, 1989, 1990; Broomhead and Lowe, 1988).

In a series of papers we have explored a specific, albeit quite general, approach to the problem of function approximation. The approach is based on the recognition that the ill-posed problem of function approximation from sparse data must be constrained by assuming an appropriate prior on the class of approximating functions. Regularization techniques 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 in the case of classification where smoothness involves the classification boundaries rather than the input-output mapping itself. Our use of regularization, which follows the classical technique introduced by Tikhonov (1963, 1977), 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 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) 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; Micchelli, 1986; Dyn, 1987) that we have extended to Hyper Basis Functions (Poggio and Girosi, 1990, 1990a). The regularization networks with radial stabilizers we studied include all the classical one-dimensional as well as multidimensional splines and approximation techniques, such as radial and non-radial Gaussian or multiquadric functions. In Poggio and Girosi (1990, 1990a) we have extended this class of networks to Hyper Basis Functions (HBF). In this paper we show that an extension of Regularization Networks, that 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.

The plan of the paper is as follows. We first discuss the solution of the variational problems of regularization in a rather general form. 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-know 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 extension that leads from Radial Basis Functions to Hyper Basis Functions leads from additive models to ridge approximation, containing as special cases Breiman's hinge functions (1992) and ridge approximations of the type of Projection Pursuit Regression (PPR) (Friedman and Stuezle, 1981; Huber, 1985). Simple numerical experiments are then described to illustrate the theoretical arguments.

In summary, the chain of our arguments shows that ridge approximation schemes such as

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

where

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

are approximations of Regularization Networks with appropriate additive stabilizers. The form of G depends on the stabilizer, and includes in particular cubic splines (used in typical implementations of PPR) and one-dimensional Gaussians. It seems, however, impossible to directly derive from regularization principles the sigmoidal activation functions used in Multilayer Perceptrons. We discuss 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 of 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* knowedge 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:

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

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 terms.

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 memo, 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:

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

where $\tilde{}$ indicates the Fourier transform, $\tilde{G}$ is some positive function that falls off to zero as $\|\mathbf{s}\| \rightarrow \infty$ (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) 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 eq. (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:

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

where $\{\psi_\alpha\}_{\alpha=1}^{k}$ is a basis in the k dimensional null space $\mathcal{N}$ and the coefficients d_α and c_i satisfy the following linear system:

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

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

where I is the identity matrix, and we have defined

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

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, and in this case the solvability of the linear system depends on the properties of the basis function G.

The approximation scheme of eq. form (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 simple extension to vector output scheme.

3 Classes of stabilizers

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

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

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 (4), corresponding to different properties of the basis functions G. Each of them corresponds to different *a priori* assumptions about the smoothness of the function that must be approximated.

3.1 Radial stabilizers

Most of the commonly used stabilizers have radial simmetry, 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 priviliged directions. Rotation invariant stabilizers correspond clearly 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 (Micchelli, 1986; Powell, 1987). The class of admissible Radial Basis Functions is the class of conditionally positive definite functions of any order, since it has been shown (Madych and Nelson, 1991; Dyn, 1991) that in this case the functional of eq. (4) 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 multivariate 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

$$G(\mathbf{x}) = \begin{cases} \|\mathbf{x}\|^{2m-d} \ln \|\mathbf{x}\| & \text{if } 2m > d \text{ and } d \text{ is even} \\ \|\mathbf{x}\|^{2m-d} & \text{otherwise.} \end{cases} \tag{5}$$

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

$$k = \begin{pmatrix} d+m-1 \\ d \end{pmatrix} .$$

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 $m = 2$, eq. (5) yields the so called "thin plate" basis function $G(\mathbf{x}) = \|\mathbf{x}\|^2 \ln \|\mathbf{x}\|$ (Harder and Desmarais, 1972), depicted in figure (1).

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, represented in figure (2). The Gaussian function is positive definite, and it is well known from the theory of reproducing kernels that positive definite functions can be used to define *norms* of the type (4). Since $\phi[f]$ is a norm, its null space contains only the zero element, and the additional null space terms of eq. (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.

$G(r) = e^{-\beta r^2}$	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.
$G(r) = r^{2n+1}$	multivariate splines, c.p.d. n
$G(r) = r^{2n} \ln r$	multivariate splines, c.p.d. n

3.2 Tensor product stabilizers

An alternative to choosing a radial function $\tilde{G}$ in the stabilizer (4) is a *tensor product* type of basis function, that is a function of the form

$$\tilde{G}(\mathbf{s}) = \Pi_{j=1}^d \tilde{g}(s_j) \tag{6}$$

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 (6) 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 in not very smooth, as shown in figure (3), and its performance in practical cases should first be tested experimentally.

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 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

$$f(\mathbf{x}) = \sum_{\mu=1}^{d} f_\mu(x^\mu) \tag{7}$$

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 (see Hastie and Tibshirani's book, 1990) and can be consider 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 a stabilizer that corresponds to an additive basis function (see fig. 4 for an example):

$$G(\mathbf{x}) = \sum_{\mu=1}^{n} \theta_\mu g(x^\mu) \tag{8}$$

where θ_μ are certain fixed parameters. Such a choice, in fact, leads to an approximation scheme of the form (7) in which the additive components f_μ have the form:

$$f_\mu(x) = \theta_\mu \sum_{i=1}^{N} c_i G(x^\mu - x_i^\mu) \tag{9}$$

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 to write stabilizers corresponding to the basis function (8) in the form (4), 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 (8) is a distribution. For example, in two dimensions we obtain

$$\tilde{G}(\mathbf{s}) = \theta_x \tilde{g}(s_x)\delta(s_y) + \theta_y \tilde{g}(s_y)\delta(s_x) \tag{10}$$

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 eq. (10) with Gaussians of very small variance. Consider, for example in two dimensions, the stabilizer:

$$\phi[f] = \int_{R^d} d\mathbf{s}\ \epsilon \frac{|\tilde{f}(\mathbf{s})|^2}{\theta_x \tilde{g}(s_x) e^{-(\frac{s_y}{\epsilon})^2} + \theta_y \tilde{g}(s_y) e^{-(\frac{s_x}{\epsilon})^2}} \tag{11}$$

This corresponds to a basis function of the form:

$$G(x, y) = \theta_x g(x) e^{-\epsilon^2 y^2} + \theta_y g(y) e^{-\epsilon^2 x^2}\ . \tag{12}$$

In the limit of ϵ going to zero the denominator in expression (11) approaches eq. (10), and the basis function (12) 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:

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

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

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i G(\mathbf{x} - \mathbf{x}_i) \tag{13}$$

where

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

as in eq. (8).

We notice that the additive component of eq. (13) 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 prior assumption of additivity and smoothness of the class of functions to be approximated. In the first technique the two assumptions are both in the choice of the stabilizer, (eq. 11); 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 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 $f(\mathbf{x}) = F(W\mathbf{x})$ for some smooth function F. The smoothness assumption is now made directly on F, through a smoothness functional $\phi[F]$ of the form (4). The regularization functional is now 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:

$$f(\mathbf{x}) = F(\mathbf{Wx}) = \sum_{i=1}^{N} c_i G(\mathbf{Wx} - \mathbf{W}\mathbf{x}_i) \qquad (14)$$

This argument is exact 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 probably 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 to replace the approximation scheme of eq. (14) with a similar one, in which he 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:

$$f(\mathbf{x}) = \sum_{\alpha=1}^{n} c_\alpha G(\mathbf{Wx} - \mathbf{W}\mathbf{t}_\alpha) \ . \qquad (15)$$

where $n < N$ and the *centers* $\mathbf{t}_\alpha$ are chosen according to some heuristic (Moody and Darken, 1989), or are considered as free parameters (Poggio and Girosi, 1989, 1990). The coefficients c_α and the elements of the matrix $\mathbf{W}$ are estimated accordingly to a least squares criterion. The elements of the matrix $\mathbf{W}$ could also be estimated through cross-validation, 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:

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

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).

The next sections discuss approximation schemes of the form (15) 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 eq. (15) 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:

$$\|\mathbf{x}\|_{\mathbf{W}} \equiv \mathbf{x} \cdot \mathbf{W}^T\mathbf{W}\mathbf{x} \ . \qquad (16)$$

The basis functions of eq. (15) are not radial anymore, or, more accurately, they are radial in the metric defined by eq. (16). This means that the level curves of the basis functions are not circles, but ellipses, whose axes 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 product matrix $\mathbf{W}^T\mathbf{W}$. Therefore, by the Cholesky decomposition, it is sufficient to take $\mathbf{W}$ upper triangular. The approximation scheme defined by eq. (15) has been discussed in detail in (Poggio and Girosi, 1990; Girosi, 1992), so we do will not discuss it further, and will consider, in the next section, 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).

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

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i \sum_{\mu=1}^{d} \theta_\mu G(x^\mu - x_i^\mu) \qquad (17)$$

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 eq. (17) 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 the overfitting problem. We then adopt the same idea presented in section (4), and consider an approximation scheme of the form

$$f(\mathbf{x}) = \sum_{\alpha=1}^{n} \sum_{\mu=1}^{d} c_\alpha^\mu G(x^\mu - t_\alpha^\mu) \ , \qquad (18)$$

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 eq. (18) 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 $\mathbf{x} \rightarrow \mathbf{Wx}$, where $\mathbf{W}$ is a $d' \times d$ matrix. Calling $\mathbf{w}_\mu$ the μ-th column of W, and performing the substitution $\mathbf{x} \rightarrow \mathbf{Wx}$ in eq. (18), we obtain

$$f(\mathbf{x}) = \sum_{\alpha=1}^{n} \sum_{\mu=1}^{d'} c_\alpha^\mu G(\mathbf{w}_\mu \cdot \mathbf{x} - t_\alpha^\mu) \ . \qquad (19)$$

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 eq: (19) as

$$f(\mathbf{x}) = \sum_{i=1}^{d'} h_\mu(\mathbf{w}_\mu \cdot \mathbf{x}) \ . \qquad (20)$$

Notice the similarity between eq. (20) 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 $h_k(y)$ is left to the user. In our case the h_k 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_k have been indeed chosen to be cubic splines.

Let us briefly review the steps that bring us from the classical additive approximation scheme of eq. (9) to a Projection Pursuit Regression-like type of approximation:

1. the regularization parameters θ_μ of the classical approximation scheme (9) are considered as free parameters;
2. the number of centers is chosen to be smaller than the number of data points;
3. it is assumed that the true relevant variables are some unknown linear combination of the original variables;

We notice that in the special case in which each additive component has just one center ($n = 1$), the approximation scheme of eq. (19) becomes:

$$f(\mathbf{x}) = \sum_{\mu=1}^{d'} c^\mu G(\mathbf{w}_\mu \cdot \mathbf{x} - t^\mu) \ . \qquad (21)$$

If the basis function G were a sigmoidal function this would be clearly a standard Multilayer Perceptron with one layer of hidden units. Sigmoidal functions cannot be derived from regularization theory, but we will see in section (6) the relationship between a sigmoidal function and a basis function that can be derived from regularization, like the absolute value function.

There are clearly a number of computational issues related to how to find the parameters of an approximation scheme like the one of eq. (19), 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 Priors, stabilizers and basis functions

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 (Marroquin et al., 1987; Bertero et al., 1988, Wahba, 1990). In this section we illustrate this connection informally, without addressing the several deep mathematical issues of the problem.

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

$$f(\mathbf{x}_i) = y_i + \epsilon_i, \quad i = 1, \ldots, N \tag{22}$$

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, being 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 know, the posterior distribution $\mathcal{P}[f|g]$ can now be computed by applying the Bayes rule:

$$\mathcal{P}[f|g] \propto \mathcal{P}[g|f] \; \mathcal{P}[f]. \tag{23}$$

We now make the assumption that the noise variables in eq. (22) 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 rigorously formalized (Marroquin et al., 1987). The prior probability $\mathcal{P}[f]$ is written as

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

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 (23) the *a posteriori* probability of f is written as

$$\mathcal{P}[f|g] \propto e^{-\frac{1}{2\sigma^2}[\sum_{i=1}^N (y_i - f(\mathbf{x}_i))^2 + 2\alpha\sigma^2\phi[f]]}. \tag{24}$$

One simple estimate of the function f from the probability distribution (24) is the so called MAP (*Maximum A Posteriori*) estimate, that considers the function that maximizes the *a posteriori* probability $\mathcal{P}[f|g]$, or minimizes the exponent in equation (24). 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 eq. (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.

As we have pointed out (Poggio and Girosi, 1989), 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 in 1978). They connect data compression and coding with Bayesian inference, regularization, function approximation and learning.

5.1 The Bayesian interpretation of Generalized Regularization Networks

In the probabilistic interpretation of standard regularization the term $\lambda\phi[f]$ in the regularization functional corresponds to the following prior probability in a Bayesian formulation in which the MAP estimate is sought:

$$\mathcal{P}[f] \propto e^{-\lambda\phi[f]}.$$

From this point of view, the extension of section (4) corresponds (again informally) to choose an *a priori* probability of the form

$$\mathcal{P}[f] \propto \int \; \delta g e^{-\lambda\phi[g]}\delta(f(\mathbf{x}) - g(\mathbf{W}\mathbf{x}))$$

where δg means that a functional integration is being performed. This restricts the space of functions on which the probability distribution is defined to the class of functions that can be written as $f(\mathbf{x}) = g(\mathbf{W}\mathbf{x})$, and assume a prior probability distribution $e^{-\lambda\phi[g((\mathbf{x}))]}$ for the functions g, where ϕ is now a radially symmetric stabilizer.

In a similar manner, in the case of additive approximation the prior probability of f is concentrated on those functions f that can be written as sums of additive components, and corresponding priors are of the form:

$$\mathcal{P}[f] \propto \int \; \delta f_1 \ldots \delta f_d \; \Pi_{\mu=1}^d e^{-\frac{1}{\theta_\mu}\phi[f_\mu]}\delta\left(f(\mathbf{x}) - \sum_{\mu=1}^d f_\mu(x^\mu)\right) \; .$$

This is equivalent to saying that we know *a priori* that the underlying function is additive.

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

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

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. Multidimensional additive approximations consist of one-dimensional terms (once the $\mathbf{W}$ has been fixed). We consider the approximation with the lowest possible degree of smoothness: piecewise linear. The associated basis function $G(x) = |x|$ is shown in figure 5 top left, and the associated stabilizer is given by

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

Its use in approximating a one-dimensional function consists of 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 $\sigma_L(x)$ shown in figure 5. 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 5. 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 (1992), 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 $|x| = x_+ + x_-$.

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 $f(x) = \sum_\alpha^n c_\alpha G(x - t_\alpha)$ to approximate the function $h(x) = \sin(2\pi x)$ on $[0,1]$, where $G(x)$ is one of the basis functions of figure 5. The function $\sin(2\pi x)$ is plotted in figure 6. Fifty training points and 10,000 test points were chosen uniformly on $[0,1]$. The parameters were learned using the iterative backfitting algorithm that will be described in section 7. We looked at the function learned after fitting 1, 2, 4, 8 and 16 basis functions. The resulting approximations are plotted in the following figures and the errors are summarized in table 1.

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 better and the "Gaussian" function did slightly worse. 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 axes. 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 axes. 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

$$G_1 = e^{-(\frac{x^2}{\sigma_1} + \frac{y^2}{\sigma_2})}$$

and

$$G_2 = e^{-(\frac{x^2}{\sigma_2}+\frac{y^2}{\sigma_1})}$$

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

The fourth model is an additive model of the form (18), in which the number of centers is smaller than the number of data points, but the additive components are independent, and can be written as:

$$f(x, y) = \sum_{\alpha=1}^{n} b_\alpha G(x - t_x^\alpha) + \sum_{\beta=1}^{n} c_\beta G(y - t_y^\beta)$$

where the basis function is the Gaussian:

$$G(x) = e^{-2x^2}.$$

In this model, the centers were also fixed in the learning algorithm, and were a proper subset of the training examples, so that there were fewer centers than examples. In the experiments that follow, 7 centers were used with this model, and the coefficients b_α and c_α were determined by least squares.

The fifth model is a Generalized Regularization Network model, of the form (21), 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 the weight vectors, centers, and coefficients are all learned.

The coefficients of the first four models were set by solving the linear system of equations by using the pseudo-inverse, which finds the best mean squared fit of the linear model to the data.

The fifth model was trained by fitting one basis function at a time according to the following algorithm:

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

The random step algorithm (Caprile and Girosi, 1990) for optimizing a set of parameters 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:

$$h(x, y) = \sin(2\pi x) + 4(y - 0.5)^2$$

and the two-dimensional Gabor function:

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

The graphs of these functions are shown in figure 10. The training data for the additive function consisted of 20 points picked from a uniform distribution on $[0, 1] \times [0, 1]$. Another 10,000 points were randomly chosen to serve as test data. The training data for the Gabor function consisted of 20 points picked from a uniform distribution on $[-1, 1] \times [-1, 1]$ with an additional 10,000 points used as test data.

In order to see how sensitive were the performances to the choice of basis function, we also repeated the experiments for the models 3, 4 and 5 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:

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

These models (6, 7 and 8) are shown in table 2 together with models 1 to 5. Notice that only model 8 is a Multilayer Perceptron in the standard sense. The results are summarized in table 3.

Plots of some of the approximations are shown in figures 11, 12, 13 and 14. As expected, the results show that the additive model was able to approximate the additive function, $h(x, y)$ better than both the RBF model and the elliptical Gaussian model. Also, there seems to be a smooth degradation of performance as the model changes from the additive to the Radial Basis Function (figure 11). Just the opposite results are seen in approximating the non-additive Gabor function, $g(x, y)$. The RBF model did very well, while the additive model did a very poor job in approximating the Gabor function (figures 12 and 13a). However, we see that the GRN scheme (model 5), gives a fairly good approximation (figure 13b). This is due to the fact that the learning algorithm was able to find better directions to project the data than the x and y axes as in the pure additive model. We can also see from table 3 that the additive model with fewer centers than examples (model 4) has a larger training error than the purely additive model 3, but a much smaller test error. The results for the sigmoidal additive model learning the additive function h (figure

14) show that it is comparable to the Gaussian additive model. 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. This is due to the ill-conditioning of the associated linear system, which is a common problem in Radial Basis Functions (Dyn, Levin and Rippa, 1986).

8 Summary and remarks

A large number of approximation techniques can be written as multilayer networks with one hidden layer, as shown in figure (16). In past papers (Poggio and Girosi, 1989; Poggio and Girosi, 1990, 1990b; Maruyama, Girosi and Poggio, 1992) we showed how to derive RBF, HBF and several types of multidimensional splines from regularization principles of the form used to deal with the ill-posed problem of function approximation. 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 PPR 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 and tensor products splines. Figure (15) shows a diagram of the relationships. Notice that HBF networks and Ridge Regression networks are directly related in the special case of normalized inputs (Maruyama, Girosi and Poggio, 1992). Also note that Gaussian HBF networks, as described by Poggio and Girosi (1990) contain in the limit the additive models we describe here.

We feel that there is now a theoretical framework that 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 all the different networks and corresponding approximation schemes can be justified in terms of the variational principle

$$H[f] = \sum_{i=1}^{N} (f(\mathbf{x}_i) - y_i)^2 + \lambda\phi[f] \ . \qquad (25)$$

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 (see our numerical experiments in this paper, and Maruyama et al. 1992; see also Donohue and Johnstone, 1989).

Of the several points suggested by our results we will discuss one here: it regards 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 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 – 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. 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 independently accessible 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 report here only for the sake of curiosity. It may be argued that humans possibly 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"*.

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:

$$H[f] = \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i))^2 + \lambda\phi[f] \ . \qquad (26)$$

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 (26) all have the same form. A detailed and rigorous derivation of the solution of the variational principle associated with eq. (26) 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:

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

obtaining the functional

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

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

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

so that the functional can be rewritten as:

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

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

$$\frac{\delta H[\tilde{f}]}{\delta \tilde{f}(\mathbf{t})} = 0 \quad \forall \mathbf{t} \in R^d\ . \tag{27}$$

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

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

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 eq. (27) 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:

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

We now remember 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

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

where $p(\mathbf{x})$ is a term that lies in the null space of $\phi[f]$.

B Approximation of vector fields through multioutput regularization networks

Consider the problem of approximating a vector field $\mathbf{y}(\mathbf{x})$ from a set of sparse data, the examples, which are pairs $(\mathbf{y}_i, \mathbf{x}_i)$ for $i = 1\cdots N$. 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 17). Then the approximation is

$$\mathbf{y}(\mathbf{x}) = \sum_{i=1}^{n}\mathbf{c}_iG(\mathbf{x}-\mathbf{x}_i)$$

with G being the chosen Green function. The equation can be rewritten in matrix notation as

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

where $\mathbf{g}$ is the vector with elements $g_i = G(\mathbf{x} - \mathbf{x}_i)$.
Let us define as $\mathbf{G}$ the matrix of the chosen Green function evaluated at the examples, that is, the matrix with elements $G_{i,j} = G(\mathbf{x}_i - \mathbf{x}_j)$. Then the "weights" $\mathbf{c}$ are "learned" from the examples by solving

$$\mathbf{Y} = \mathbf{C}\mathbf{G}$$

where $\mathbf{Y}$ is defined as the matrix in which column l is the example $\mathbf{y}_l$. $\mathbf{C}$ is defined as the matrix in which row m is the vector $\mathbf{c}_m$. This means that $\mathbf{x}$ is a $d \times 1$ matrix, $\mathbf{C}$ is a $q \times n$ matrix, $\mathbf{Y}$ is a $q \times N$ matrix and $\mathbf{G}$ is a $n \times N$ matrix. Then the set of weights C is given by

$$\mathbf{C} = \mathbf{Y}\mathbf{G}^{+}$$

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

$$\mathbf{y}(\mathbf{x}) = \mathbf{Y}\mathbf{G}^{+}\mathbf{g}(\mathbf{x})$$

which can be rewritten as

$$\mathbf{y}(\mathbf{x}) = \sum_{l=1}^{N} b_l(\mathbf{x})\mathbf{y}_l$$

where the b_l depend on the chosen G, according to

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

Thus *for any choice of the regularization network – even HBF – and any choice of the Green function – including Green functions corresponding to additive splines and tensor product splines – the estimated output vector is always a linear combination of example vectors with coefficients* $\mathbf{b}$ *that depend (nonlinearly) on the input value.* The result is valid for all networks with one hidden layer and linear outputs, provided that a L^2 criterion is used for training. Thus, for all types of regularization networks and all Green functions the output is always a linear combination of output examples (see Poggio and Girosi 1989).

References

[1] 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.

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

[3] L. Breiman. Hinging hyperplanes for regression, classification, and function approximation. 1992. (submitted for publication).

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

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

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

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

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

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

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

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

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

[19] 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.

[20] 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.

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

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

[23] 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.

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

[25] 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.

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

[27] 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.

[28] M.J.D. Powell. The theory of radial basis functions approximation in 1990. Technical Report NA11, Department of Applied Mathematics and Theoretical Physics, Cambridge, England, December 1990.

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

[30] 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.

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

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

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

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

	1 basis function	2 basis functions	4 basis functions	8 basis functions	16 basis functions
Absolute value	train: 0.798076	0.160382	0.011687	0.000555	0.000056
	test: 0.762225	0.127020	0.012427	0.001179	0.000144
"Sigmoidal"	train: 0.161108	0.131835	0.001599	0.000427	0.000037
	test: 0.128057	0.106780	0.001972	0.000787	0.000163
"Gaussian"	train: 0.497329	0.072549	0.002880	0.000524	0.000024
	test: 0.546142	0.087254	0.003820	0.001211	0.000306

Table 1: L_2 training and test error for each of the 3 piecewise linear models using different numbers of basis functions.

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	$f(x,y)=\sum_{\alpha=1}^{7} b_\alpha e^{-\frac{(x-t_x^\alpha)^2}{\sigma}}+\sum_{\beta=1}^{7} c_\beta e^{-\frac{(y-t_y^\beta)^2}{\sigma}}$	$\sigma=0.5$
Model 5	$f(x,y)=\sum_{\alpha=1}^{n} c_\alpha e^{-(\mathbf{W}_\alpha\cdot\mathbf{X}-t_\alpha)^2}$	–
Model 6	$f(x,y)=\sum_{i=1}^{20} c_i[\sigma(x-x_i)+\sigma(y-y_i)]$	–
Model 7	$f(x,y)=\sum_{\alpha=1}^{7} b_\alpha\sigma(x-t_x^\alpha)+\sum_{\beta=1}^{7} c_\beta\sigma(y-t_y^\beta)$	–
Model 8	$f(x,y)=\sum_{\alpha=1}^{n} c_\alpha\sigma(\mathbf{w}_\alpha\cdot\mathbf{x}-t_\alpha)$	–

Table 2: The eight models we tested in our numerical experiments.

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8
$h(x,y)$	train: 0.000036	0.000067	0.000001	0.000001	0.000170	0.000001	0.000003	0.000743
	test: 0.011717	0.001598	0.000007	0.000009	0.001422	0.000015	0.000020	0.026699
$g(x,y)$	train: 0.000000	0.000000	0.000000	0.345423	0.000001	0.000000	0.456822	0.000044
	test: 0.003818	0.344881	67.95237	1.222111	0.033964	98.419816	1.397397	0.191055

Table 3: 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.

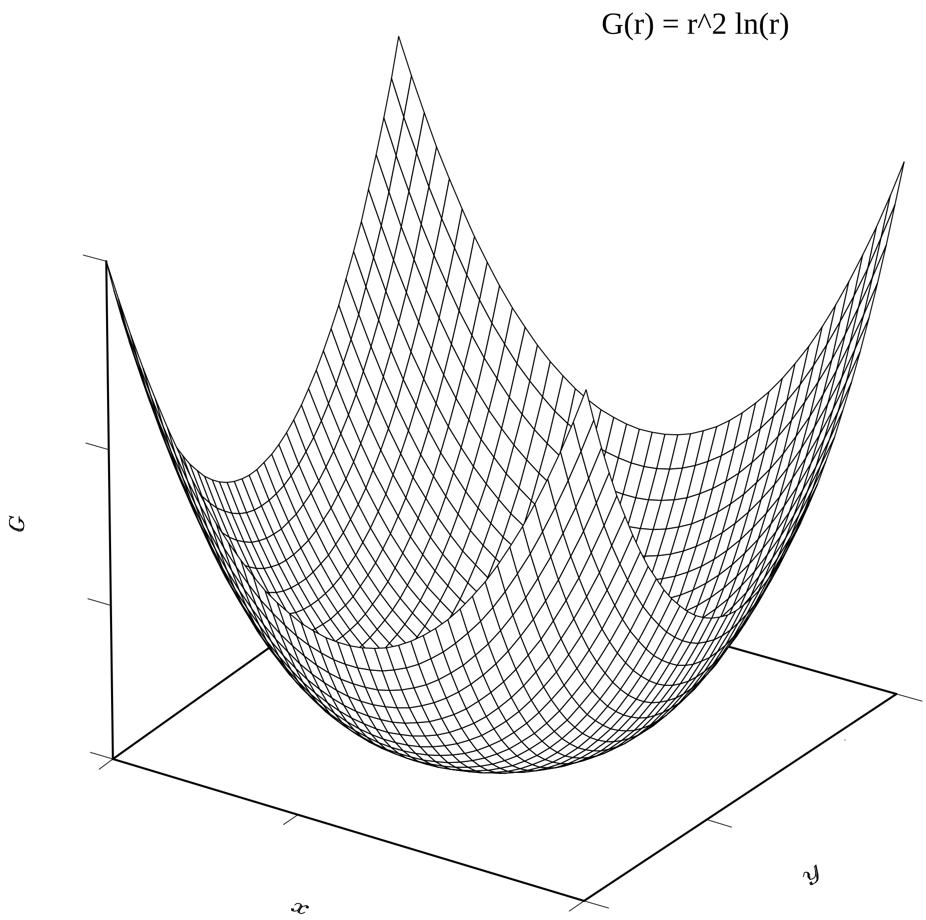

Figure 1: The "thin plate" radial basis function $G(r) = r^2 \ln(r)$, where $r = \|\mathbf{x}\|$.

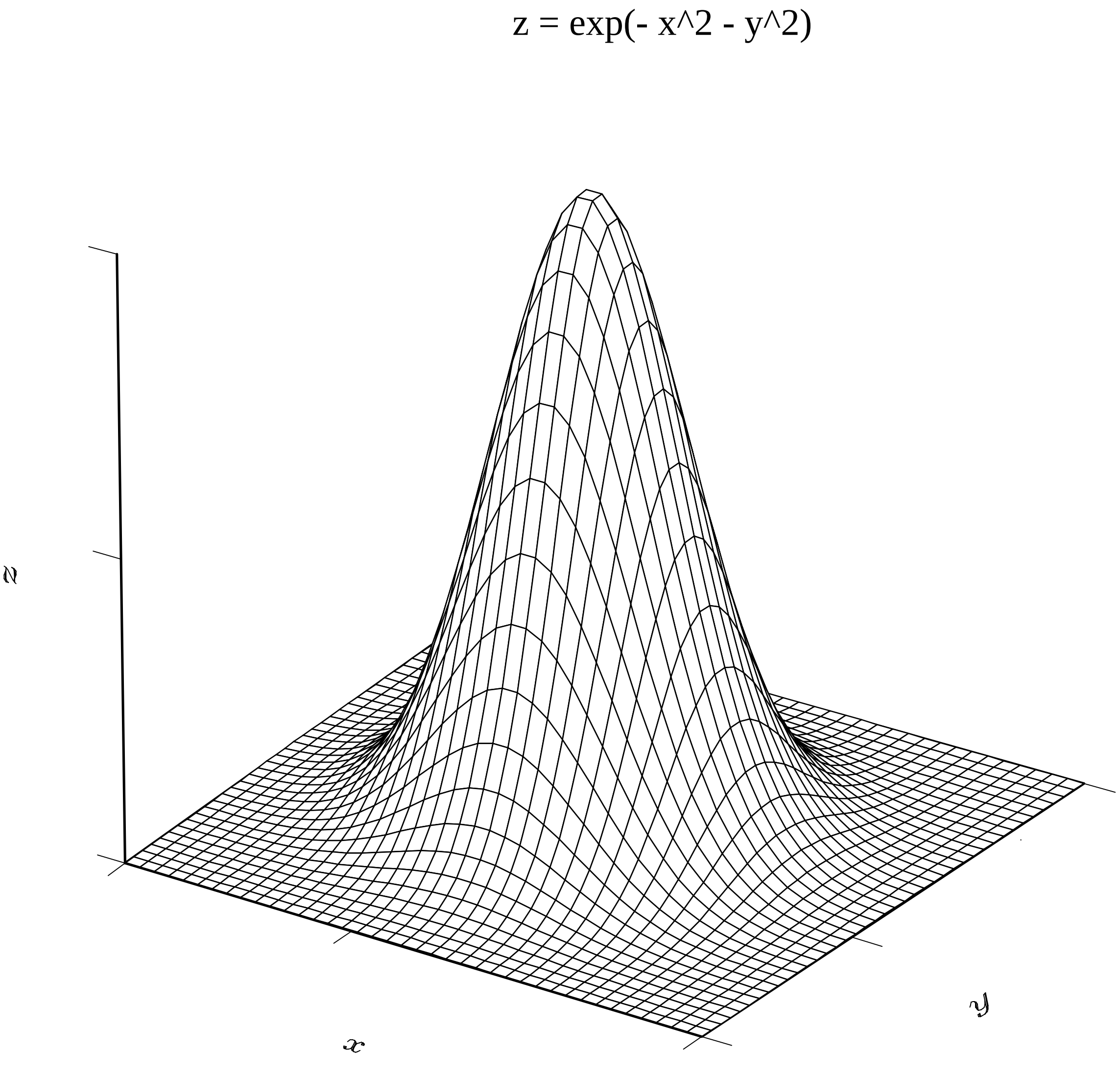

Figure 2: The Gaussian basis function $G(r) = e^{-r^2}$, where $r = \|\mathbf{x}\|$.

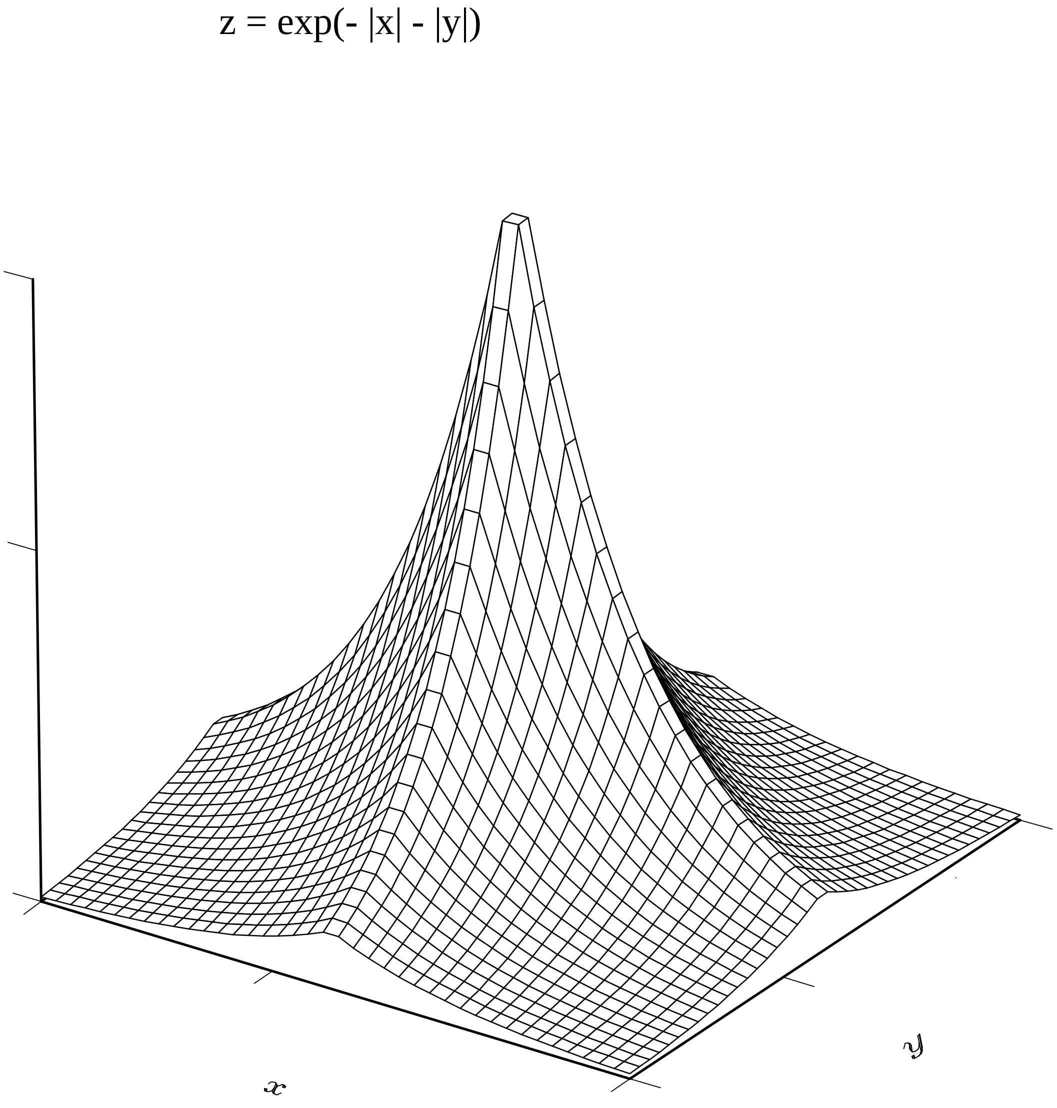

Figure 3: The basis function $G(\mathbf{x}) = e^{-\|\mathbf{x}\|_{L_1}}$

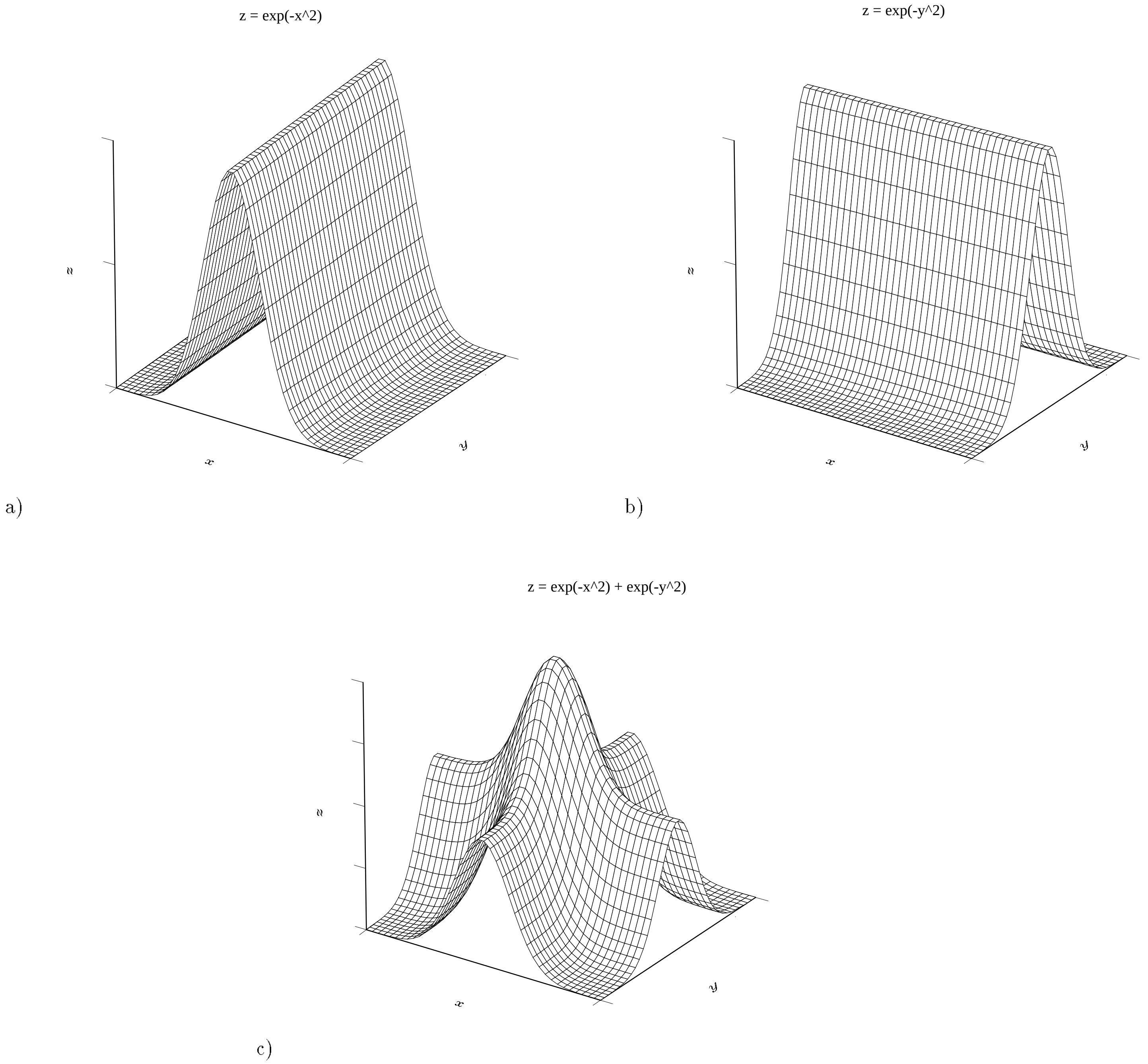

Figure 4: In (c) it is shown an additive basis function, in the case in which the additive component of the basis functions (a and b) are gaussian.

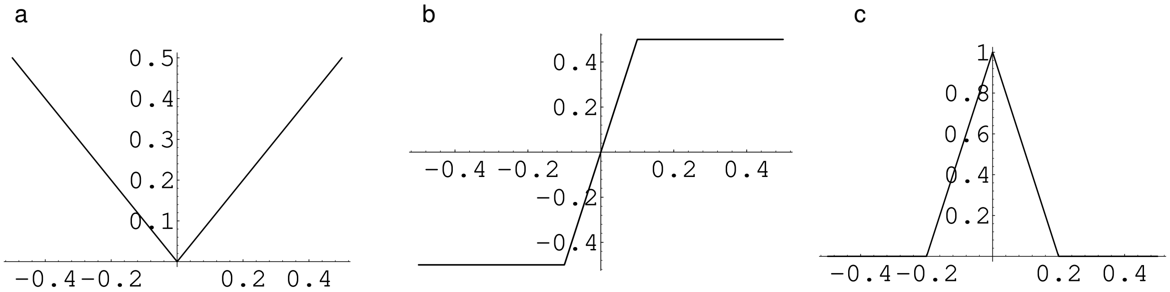

Figure 5: a) Absolute value basis function, $|x|$, b) "Sigmoidal" basis function $\sigma_L(x)$ c) Gaussian-like basis function $g_L(x)$

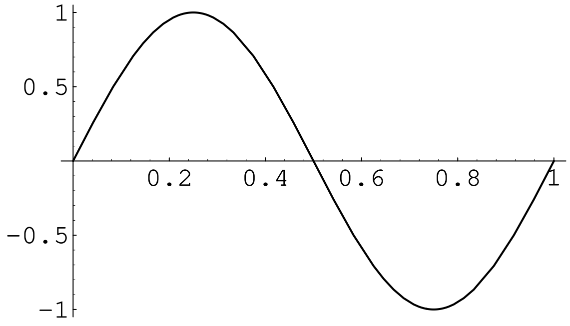

Figure 6: $Sin(2\pi x)$

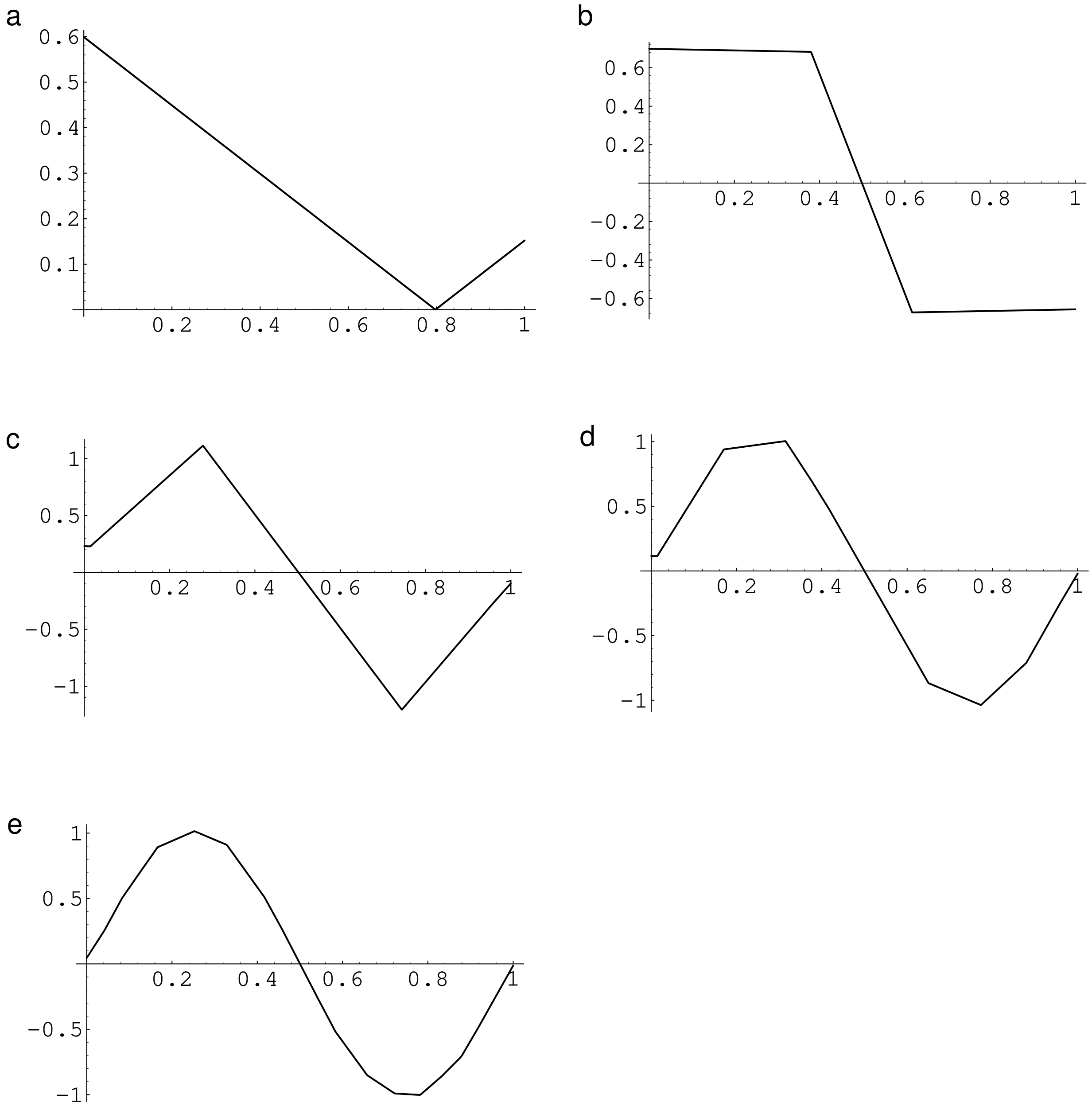

Figure 7: a) Approximation using one absolute value basis function b) 2 basis functions c) 4 basis functions d) 8 basis functions e) 16 basis functions

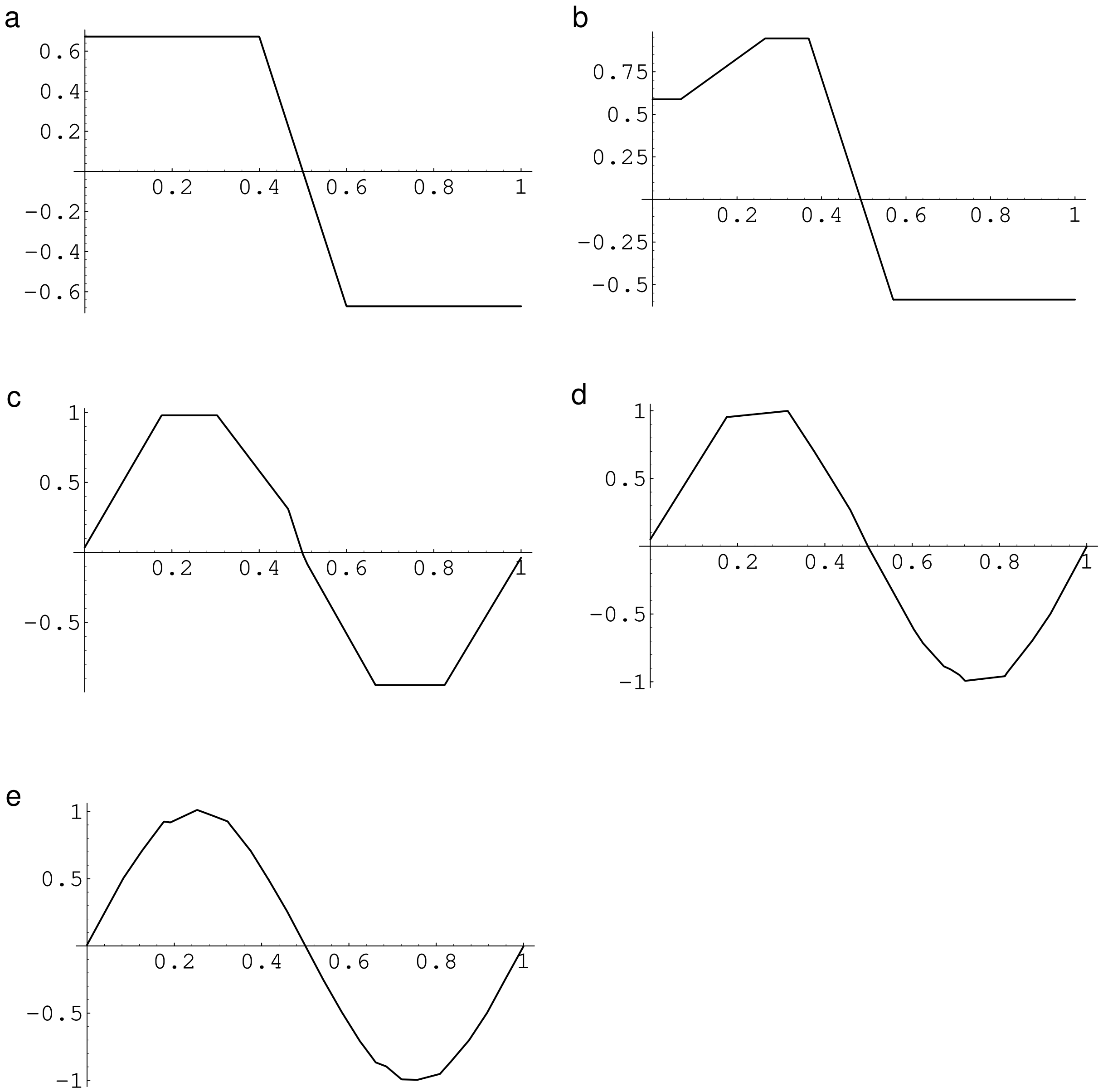

Figure 8: a) Approximation using one "sigmoidal" basis function b) 2 basis functions c) 4 basis functions d) 8 basis functions e) 16 basis functions

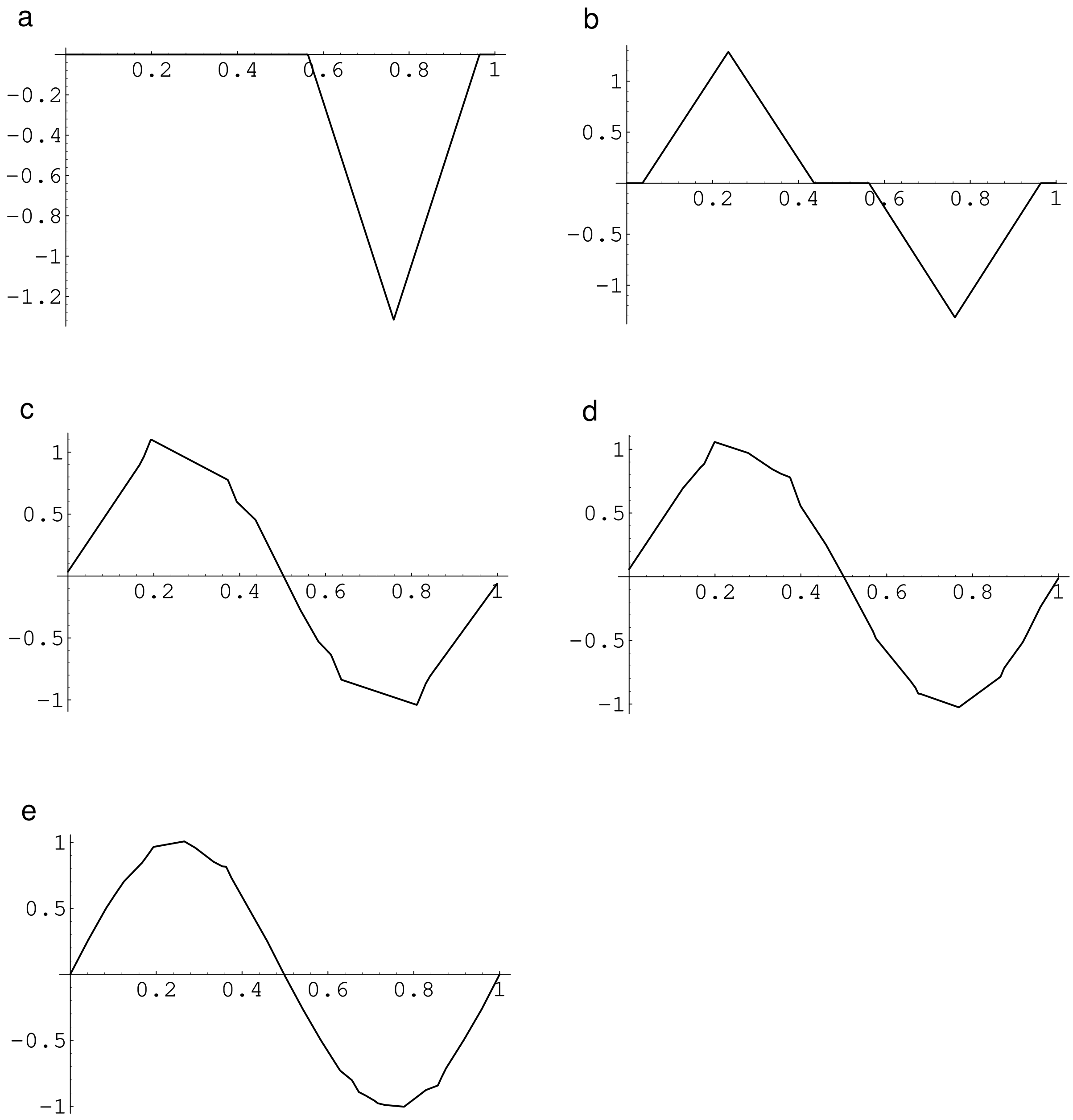

Figure 9: a) Approximation using one "Gaussian" basis function b) 2 basis functions c) 4 basis functions d) 8 basis functions e) 16 basis functions

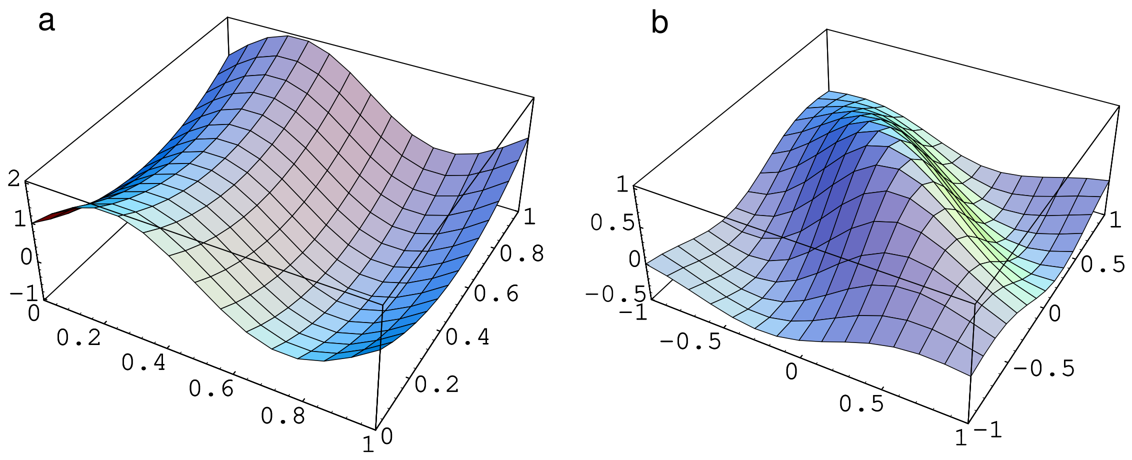

Figure 10: a) Graph of $h(x, y)$. b) Graph of $g(x, y)$.

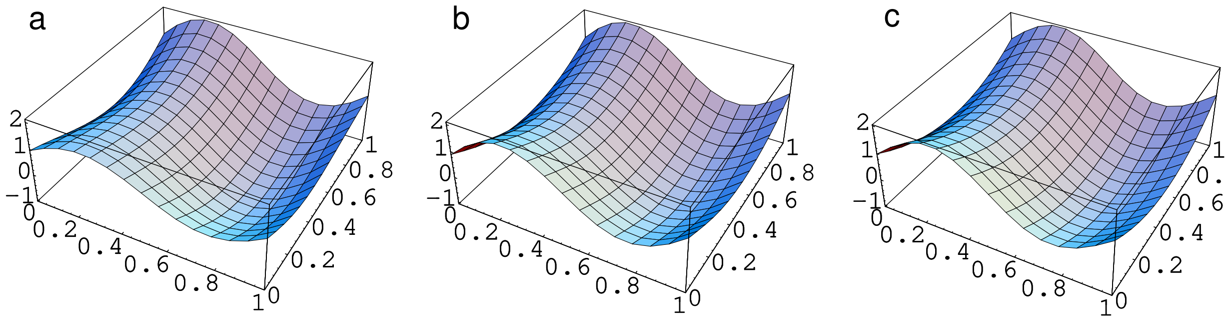

Figure 11: a) RBF Gaussian model approximation of $h(x, y)$ (model 1). b) Elliptical Gaussian model approximation of $h(x, y)$ (model 2). c) Additive Gaussian model approximation of $h(x, y)$ (model 3).

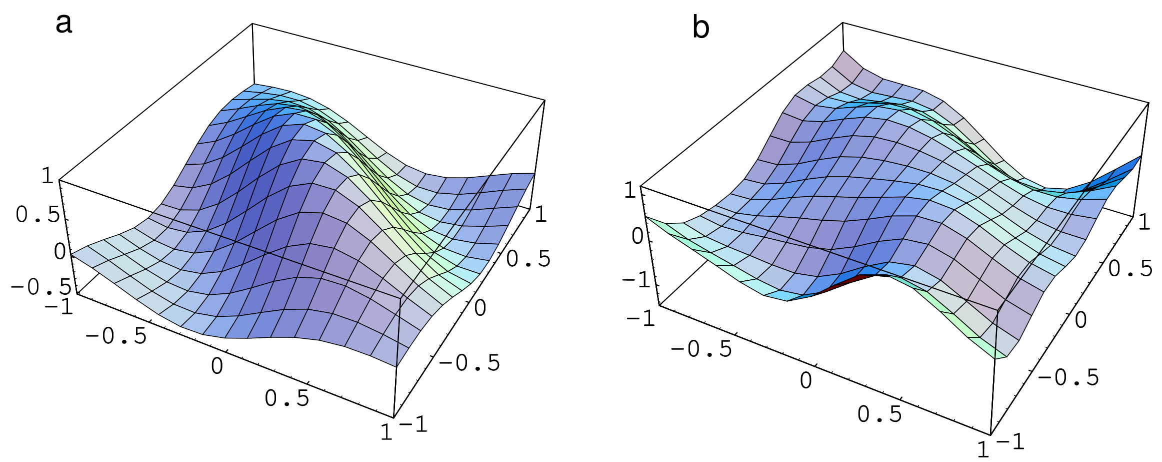

Figure 12: a) RBF Gaussian model approximation of $g(x, y)$ (model 1). b) Elliptical Gaussian model approximation of $g(x, y)$ (model 2).

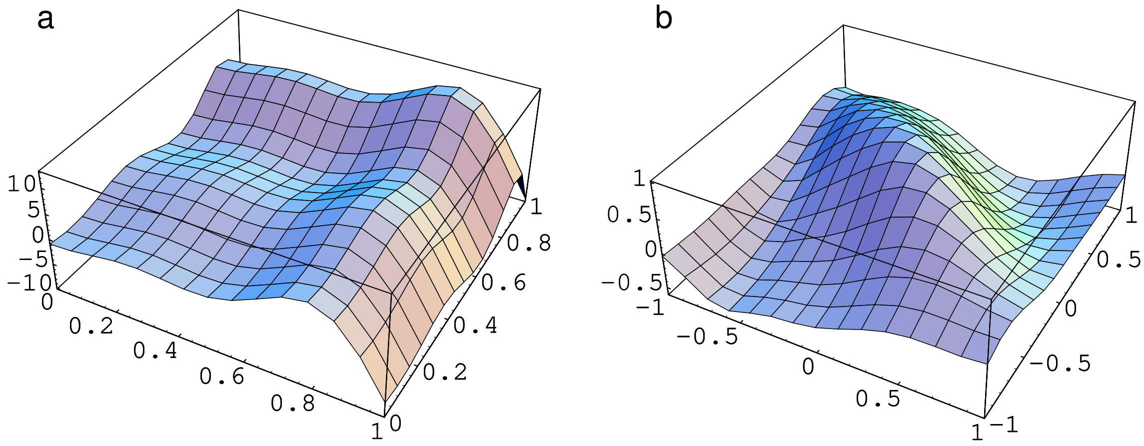

Figure 13: a) Additive Gaussian model approximation of $g(x, y)$ (model 3). b) GRN Approximation of $g(x, y)$ (model 5).

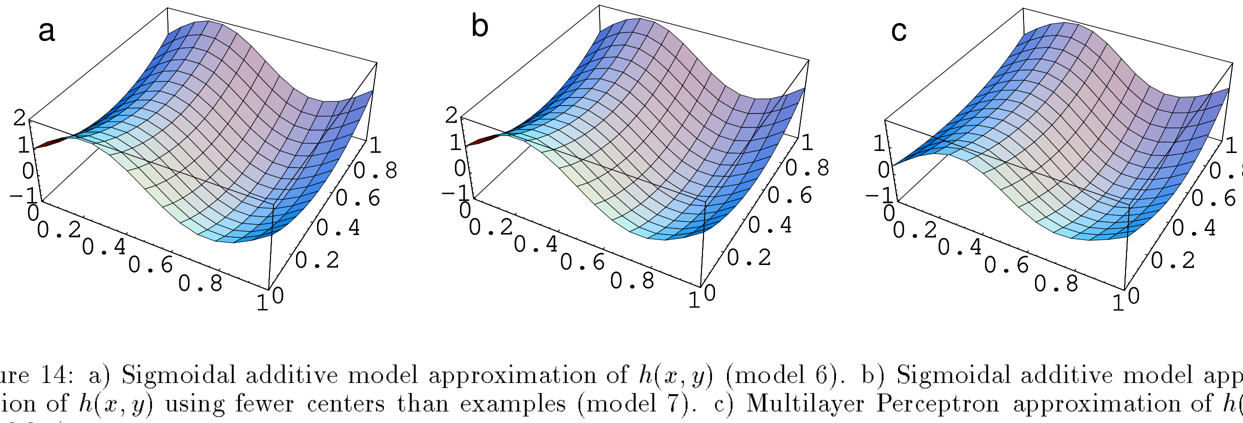

Figure 14: a) Sigmoidal additive model approximation of $h(x, y)$ (model 6). b) Sigmoidal additive model approximation of $h(x, y)$ using fewer centers than examples (model 7). c) Multilayer Perceptron approximation of $h(x, y)$ (model 8).

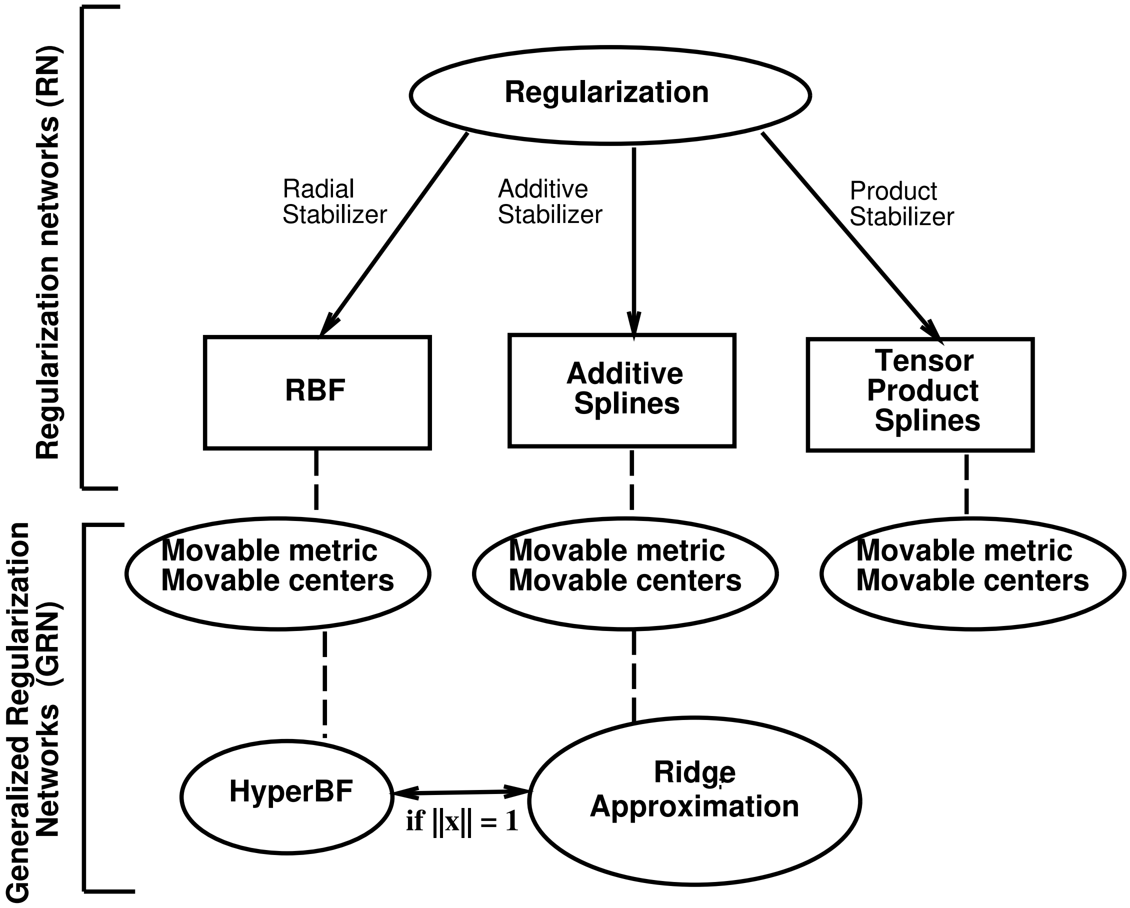

Figure 15: Several classes of approximation schemes and associated network architectures can be derived from regularization with the appropriate choice of smoothness priors and corresponding stabilizers and Greens functions

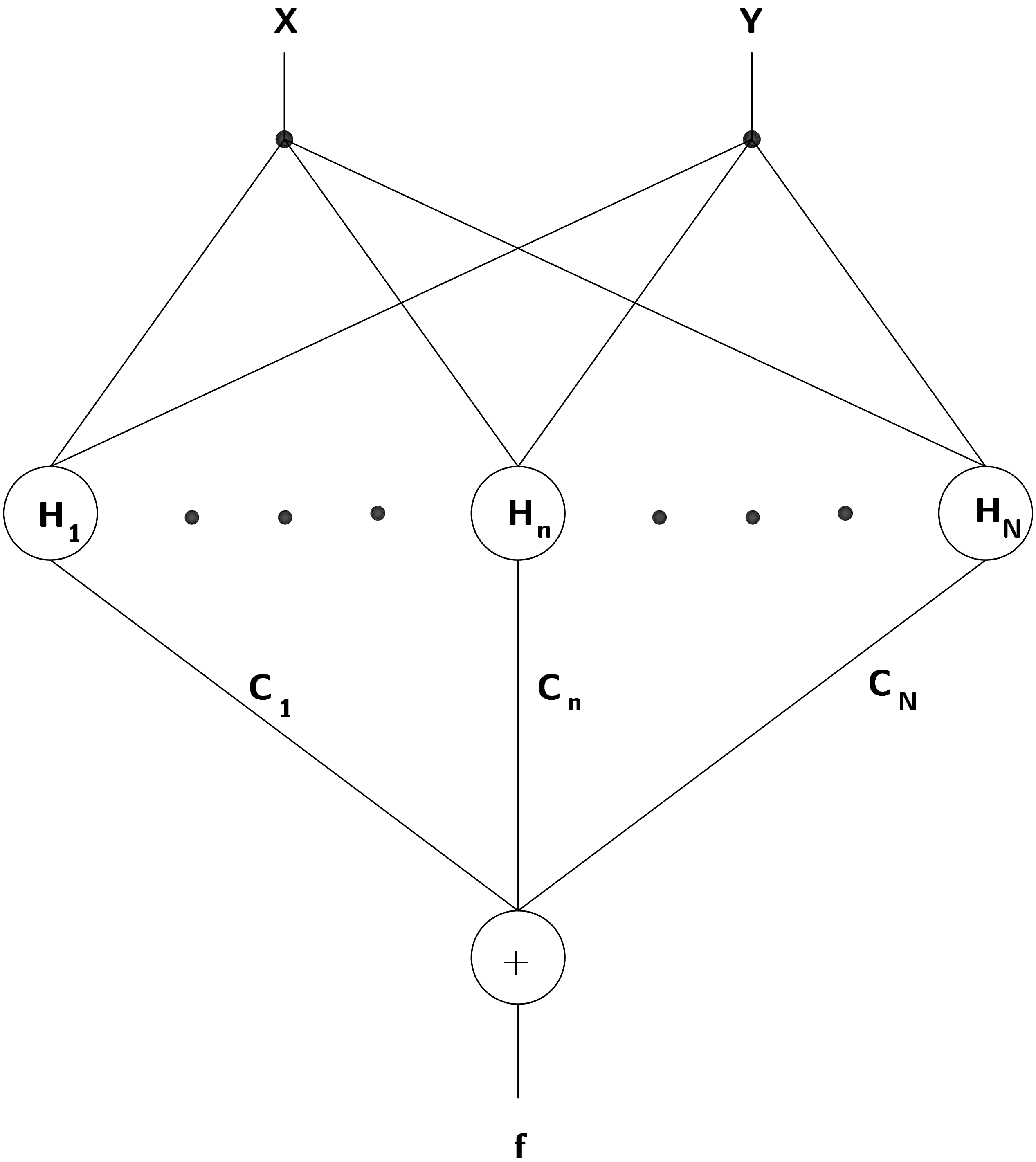

Figure 16: The most general network with one hidden layer and scalar output.

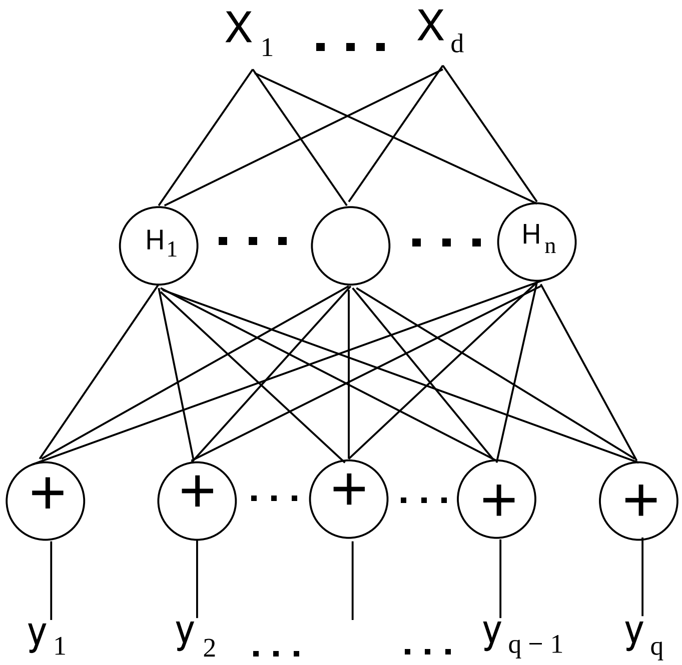

Figure 17: The most general network with one hidden layer and vector output. Notice that this approximation of a q-dimensional vector field has in general fewer parameters than the alternative representation consisting of q networks with one-dimensional outputs. If the only free parameters are the weights from the hidden layer to the output (as for simple RBF with if $n = N$) the two representations are equivalent.