EMPIRICAL LEARNING THROUGH NEURAL NETWORKS: THE WAVE-NET SOLUTION

Alexandros Koulouris, Bhavik R. Bakshi,[1] and George Stephanopoulos

Laboratory for Intelligent Systems in Process Engineering
Department of Chemical Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Empirical learning is an ever-lasting and ever-improving procedure. Although neural networks (NN) captured the imagination of many researchers as an outgrowth of activities in artificial intelligence (AI), most of the progress was accomplished when empirical learning through NNs was cast within the rigorous analytical framework of the *functional estimation problem*, or *regression*, or *model realization*. Independently of the

[1]Present address: Ohio State University, Department of Chemical Engineering, Columbus, OH 43210.

name, it has been long recognized that, due to the inductive nature of the learning problem, to achieve the desired accuracy and generalization (with respect to the available data) in a dynamic sense (as more data become available) one needs to seek the unknown approximating function(s) in functional spaces of varying structure. Consequently, a recursive construction of the approximating functions at multiple resolutions emerges as a central requirement and leads to the utilization of wavelets as the basis functions for the recursively expanding functional spaces. This chapter fuses the most attractive features of a NN: representational simplicity, ability for universal approximation, and ease in dynamic adaptation, with the theoretical soundness of a recursive functional estimation problem, using wavelets as basis functions. The result is the *Wave-Net* (wavelets, network of), a multiresolution hierarchical NN with localized learning. Within the framework of a *Wave-Net* where adaptation of the approximating function is allowed, we have explored the use of the L^∞ error measure as the design criterion. Within the framework of a *Wave-Net* one may cast any form of data-driven empirical learning to address a variety of modeling situations encountered in engineering problems such as design of process controllers, diagnosis of process faults, and planning and scheduling of process operations. This chapter will discuss the properties of a *Wave-Net* and will illustrate its use on a series of examples.

I. Introduction

Estimating an unknown function from its examples is a central problem in statistics, where it is known as the problem of *regression*. Under a different name but following the same spirit, the problem of *learning* has attracted interest in the AI research since the discovery that many intelligent tasks can be represented through functional relationships and, therefore, learning those tasks involves the solution of a regression problem (Poggio and Girosi, 1989). The significance, however, of the problem is by no means limited to those two fields. *Modeling*, an essential part of science and engineering, is the process of deriving mathematical correlations from empirical observations and as such, obeys the same principles as the regression or learning problem. Many methods developed for the solution of these problems are extensively used as tools to advance understanding and allow prediction of physical behavior.

In recent years the merging between the statistical and AI points of view on the same problem has benefited both approaches. Statistical regression techniques have been enriched by the addition of new methods

and learning techniques have found in statistics the framework under which their properties can be studied and proved. This is especially true for neural networks (NNs), which are the main product of AI research in the field and the most promising solution to the problem of functional estimation. Despite their origination as biologically motivated models of the brain, NNs have been established as nonlinear regression techniques and studied as such. Their main features are their capability to represent nonlinear mappings, their ability to learn from data and finally their inherent parallelism which allows fast implementation. Every application where an unknown nonlinear function has to be reconstructed from examples is amenable to a NN solution. This explains the wide spread of NNs outside the AI community and the explosion of their applications in numerous disciplines and for a variety of tasks. In chemical engineering and especially in process systems engineering, NNs have found ground for applications in process control as models of nonlinear systems behavior (Narendra and Parthasarathy, 1990; Ungar *et al.*, 1990; Ydstie, 1990; Hernandez and Arkun, 1992; Bhat and McAvoy, 1990; Psichogios and Ungar, 1991; Lee and Park, 1992), fault diagnosis (Hoskins and Himmelblau, 1988; Leonard and Kramer, 1991), operation trend analysis (Rengaswamy and Venkatasubramaniam, 1991), and many other areas. In all these applications, the exclusive task, NNs are required to perform, is *prediction*. From the interpretation point of view, NNs are blackbox models. They fail to provide explicitly any physical insight simply because their representation does not coincide and is not motivated by the underlying phenomenon they model. The only expectation from their use is accuracy of their predictions, and this is the criterion on which the merits of their use can be judged.

The mathematical point of view on the analysis of NNs did not only aim at depriving them from their mystery related to their biological origin, but was also supposed to provide guarantees, if any, on their performance and guide the user with their implementation. Many theorems have been recruited to support the approximation capabilities of NNs. The universal approximation property has been proved for many different activation functions used in NNs such as sigmoids (Hornik *et al.*, 1989), or radial basis functions (RBFs) (Hartman *et al.*, 1990). At the same time, it was shown that such a property is quite general and not difficult to prove for many other sets of basis functions not used in NNs. Rates of convergence to the real function have also been derived (Barron, 1994), but these are limited to special conditions on the space of functions where the unknown function belongs. The central theme of all these theorems is that given a large enough network and enough data points any unknown function can be approximated arbitrarily well. Despite these theorems, the potential

user is still puzzled by issues such as what type of network to implement, how many nodes or hidden layers to use, and how to interconnect them. The issues to these questions are still derived on empirical grounds or through a trial-and-error procedure. The theorems have, however, clearly shown that it is exactly these choices related to the above questions that determine the approximating capabilities and accuracy of a given network.

Under the new light that the mathematical analysis and practical considerations have brought, new directions in NN research have been revealed. The list of basis functions used in NNs expands steadily with new additions, and methods for overcoming the empiricism in determining the network architecture are explored (Bhat and McAvoy, 1992; Mavrovouniotis and Chang, 1992). Following this spirit, a novel NN architecture, the wavelet network or *Wave-Net* has been recently proposed by Bakshi and Stephanopoulos (1993). *Wave-Nets* use wavelets as their activation functions and exploit their remarkable properties. They apply a hierarchical localized procedure to evolve in their structure, guided by the local approximation error. *Wave-Nets* are data-centered, and all important decisions on their architecture are decided constructively by the data and that is their main attractive property.

In this study the problem of estimating an unknown function from its examples is revisited. Its mathematical description is attempted to map as closely as possible the practical problem that the potential NN user has to face. The objective of the chapter is twofold: (1) to draw the framework in which NN solutions to the problem can be developed and studied, and (2) to show how careful considerations on the fundamental issues naturally lead to the *Wave-Net* solution. The analysis will not only attempt to justify the development of the *Wave-Net*, but will also refine its operational characteristics. The motivation for studying the functional estimation problem is the derivation of a modeling framework suitable for process control. The applicability of the derived solution, however, is not limited to control implementations.

The remainder of this chapter is structured as follows. In Section II the problem of deriving an estimate of an unknown function from empirical data is posed and studied in a theoretical level. Then, following Vapnik's original work (Vapnik, 1982), the problem is formulated in mathematical terms and the sources of the error related to any proposed solution to the estimation problem are identified. Considerations on how to reduce these errors show the inadequacy of the NN solutions and lead in Section III to the formulation of the basic algorithm whose new element is the pointwise presentation of the data and the dynamic evolution of the solution itself. The algorithm is subsequently refined by incorporating the novel idea of structural adaptation guided by the use of the L^∞ error measure. The need

for a multiresolution framework in representing the unknown function is then recognized and the wavelet transform is proposed as the essential vehicle to satisfy this requirement. With this addition, the complete algorithm is presented and identified as the modified *Wave-Net* model. Modeling examples (Section IV) demonstrate the properties of the derived solution and the chapter concludes with some final thoughts (Section V).

II. Formulation of the Functional Estimation Problem

In its more abstract form, the problem is to estimate an unknown function $f(\mathbf{x})$ from a number of, possibly noisy, observations $(\mathbf{x}_i, y_i)$, where $\mathbf{x}$ and y correspond to the vector of *input* or *independent variables* and the *output variable* of that function, respectively. To avoid confusion later on, the terms *regression*, *functional estimation* or *modeling* will all equivalently refer to the process of obtaining a *solution* or *approximating function* or *model* from the set of available data. The function $f(\mathbf{x})$ will be referred as the *real* or *target* function. The objective is to use the derived estimate to make predictions on the behavior of the real function in new, unseen situations. In almost all cases, the function $f(\mathbf{x})$ provides a mathematical description of a physically sound, but otherwise unknown, relationship between $\mathbf{x}$ and y. However, the form of the approximating function sought is in no way motivated by the underlying physical phenomenon and, consequently, from the point of view of interpretability, the proposed solutions are "blackbox" models. The only basic requirements for $f(\mathbf{x})$ related to its physical origination are that (1) it exists and (2) it is continuous with respect to all its arguments and that (3) the dimensionality of the input space is known.

Before attempting to formalize and solve the problem in mathematical terms, it is instructive to recognize the characteristics that are inherent to the functional estimation task and not attributable to the particular tool used to solve it. This will also set the ground on which different solutions to the problem can be analyzed and compared.

First, it can be easily recognized that finite data are always available within a bounded region of the input space. *Extrapolation* of any approximating function beyond that region is meaningful only when the model is derived by physical considerations applicable in the entire input space. In any other case, extrapolation is equivalent to postulating hypotheses that cannot be supported by evidence and, therefore, are both meaningless and dangerous. For that reason, we will confine the search for the unknown function in the fraction of the input space where data are available.

A very important characteristic of the problem is its *inductiveness*. The task is to obtain globally valid models by generalizing partial, localized and, many times, incorrect information. Because of its inductive nature, the problem inherits some disturbing properties. No solution can ever be guaranteed and the validity of any model derived by any means is not amenable to mathematical proofs. The model validation can only be perceived as a *dynamic* procedure triggered by any available piece of data and leading to an endless cycle between model postulating and model testing. This cycle can potentially prove the inefficiencies of any proposed model, but can never establish unambiguously its correctness.

Additional complications result from the fact that the functional estimation problem is *ill-posed* (Poggio and Girosi, 1989). Our only way to check whether a given function is a potential solution, is by measuring the *accuracy* of the fit for the available data. It is, however, clear that, for every given set of data points, there exists an infinite number of, arbitrarily different from the real, functions that can approximate the data arbitrarily well. Indeed, the implementation of every available tool for data regression, including NNs, will result in an equal number of different approximating functions. All these functions are equally plausible solutions to the functional estimation problem. The question that naturally arises is whether there exists some criterion based on which potential solutions can be screened out. What is certain is that the requirement for accuracy (independently of how it is mathematically measured) is not adequate in defining a unique solution to the problem and definitely not an appropriate basis for comparing the performance of various regression methods.

Where mathematical logic fails, intuition takes over. Intuitively, not all approximating functions for a given data set are equally plausible solutions to the functional estimation problem. The attribute that distinguishes the intuitively plausible from the unfavorable solutions is the *smoothness* of the approximating curve (in this context, smoothness does not refer to the mathematical property of differentiability). It is natural to seek the "best" solution in the face of the simplest, smooth-looking function that approximates well the available data. This conforms with a celebrated philosophical principle known as *Occam's razor*, which has found wide applications in the AI area (Kearns and Vazirani, 1994) and which, in simple terms, favors the shortest hypotheses that can explain the observations. In our context, shortness is equivalent to smoothness where the latter has so far only an intuitive connotation. For example, the approximating curve in Fig. 1a is a perfectly reasonable model of the data points, despite the fact that it differs considerably from the real function that produced those points. It is also remarkable to notice that, for that example and the given data, the

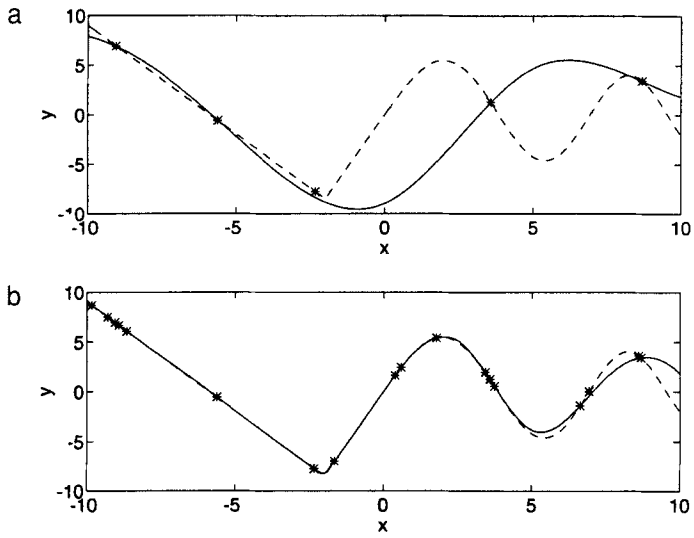

FIG. 1. Example of a functional estimation problem: evolution of the model (solid line) and comparison with the real function (dashed line) as more data (asterisks) become available.

real function is intuitively inferior (i.e., unnecessarily complicated) as an approximating function compared to its incorrect model. The real function becomes both the most accurate and smoothest (therefore, most favorable) solution when more reveal in data become available as it is the case in Fig. 1b. The fact that the model in Fig. 1a failed for additional data does not disqualify it as the "best" (in a weak, empirical sense) solution when only the initial points were available. The encouraging observation is that, by favoring intuitively simple approximating functions, the real function gradually emerges as it is evident by comparing the models in Figs. 1a and 1b.

The simple example in Fig. 1 indicates very clearly that the way the functional estimation problem should be attacked is by dynamically searching for the smoothest function that fits accurately the data. What we need is tools that will automatically produce approximating curves like the ones shown in Fig. 1. Of course, that will require us to first express the problem and its solution in mathematical terms and give precise mathematical meaning to the ambiguous attributes "accurate" and "smooth." The degree to which each proposed algorithm will prove able to mathematically represent those terms and appropriately satisfy the requirements for accuracy and smoothness will be the basis for judging its success.

A. Mathematical Description

Mathematically, the problem of deriving an estimate of a function from empirical data can be stated as follows (Vapnik, 1982):

1. *Functional Estimation Problem*

Given a set of data points $(\mathbf{x}_i, y_i) \in R^{k\times 1}$ $i = 1, 2, \ldots, l$ drawn with some unknown probability distribution, $P(\mathbf{x}, y)$, find a function $g^*(\mathbf{x})$: $X \to Y$ belonging to some class of functions G that minimizes the *expected risk functional*

$$I(g) = \int_{X\times Y} \mu[y, g(\mathbf{x})] P(\mathbf{x}, y)\, d\mathbf{x}\, dy, \tag{1}$$

where $\mu(\cdot, \cdot)$ is a metric in the space Y. We will restrict this definition to the case where y is given as a nonlinear function of $\mathbf{x}$ corrupted by some additive noise independent of $\mathbf{x}$:

$$y = f(\mathbf{x}) + d, \tag{2}$$

where d follows some unknown probability function $\mathbf{P}(d)$ and is also bounded: $|d| < \delta$. In that case, $P(\mathbf{x}, y) = P(\mathbf{x})P(y|\mathbf{x})$ and $P(y|\mathbf{x}) = \mathbf{P}[y - f(\mathbf{x})]$. The real function, $f(\mathbf{x})$, belongs to the space $C(X)$ of continuous functions defined in a closed hypercube $X = [a, b]^k$, where a and b are, respectively, the lower and upper bounds of the hypercube and k is the dimensionality of the input space. Therefore, $f(\mathbf{x})$ is bounded in both its L^2 and L^∞-norm:

$$\|f\|_2 = \int_X f^2(\mathbf{x})\, d\mathbf{x} < \infty,$$

$$\|f\|_\infty = \sup_x |f(\mathbf{x})| < \infty.$$

No other a priori assumptions about the form or the structure of the function will be made. For a given choice of g, $I(g)$ in Eq. (1) provides a measure of the real approximation error with respect to the data in the entire input space X. Its minimization will produce the function $g^*(\mathbf{x})$ that is closest to G to the real function, $f(\mathbf{x})$ with respect to the, weighted by the probability $P(\mathbf{x}, y)$ metric μ. The usual choice for μ is the Euclidean distance. Then $I(g)$ becomes the L^2-metric:

$$I(g) = \int_{X\times Y} [y - g(\mathbf{x})]^2 P(\mathbf{x}, y)\, d\mathbf{x}\, dy \tag{3}$$

In an analogous way, $I(g)$ can be defined in terms of all L^n norms with

$1 \leq n < \infty$. With a stretch of the notation, we can make Eq. (1) correspond to the L^{∞}-metric:

$$I(g) = \sup_{x} |y - g(\mathbf{x})|, \tag{4}$$

which is a meaningful definition [$I(g) < \infty$] only because we have assumed the noise to be bounded and, therefore, $y - g(\mathbf{x})$ is finite.

The minimization of the expected risk given by Eq. (1) cannot be explicitly performed, because $P(\mathbf{x}, y)$ is unknown and data are not available in the entire input space. In practice, an estimate of $I(g)$ based on the empirical observations is used instead with the hope that the function that minimizes the *empirical risk* $I_{\text{emp}}(g)$ (or *objective function*, as it is most commonly referred) will be close to the one that minimizes the real risk $I(g)$.

Let $(\mathbf{x}_i, y_i) i = 1, \ldots, l$ be the available observations. The empirical equivalents of Eqs. (3) and (4) are respectively

$$I_{\text{emp}}(g) = \sum_{i=1}^{l} [y_i - g(\mathbf{x}_i)]^2, \tag{5}$$

$$I_{\text{emp}}(g) = \max_{i=1,\ldots,l} |y_i - g(\mathbf{x}_i)|. \tag{6}$$

The magnitude of $I_{\text{emp}}(g)$ will be referred as the *empirical error*. All regression algorithms, by minimizing the empirical risk $I_{\text{emp}}(g)$, produce an estimate, $\hat{g}(\mathbf{x})$, which is the solution to the functional estimation problem.

a. Decisions Involved in the Functional Estimation Problem. As it was theoretically explained earlier, due to the inductive nature of the problem, the only requirements we can impose on the model, $\hat{g}(\mathbf{x})$ are accuracy and smoothness with respect to the data. Equations (5) and (6) provide two of the possible mathematical descriptions of the accuracy requirement. The smoothness requirement is more difficult to describe mathematically and, in fact, does not appear anywhere in the problem definition. We will postpone the mathematical description of smoothness for later. The important question now is what tools are available for controlling the accuracy and smoothness of the approximating function, $\hat{g}(\mathbf{x})$.

The mathematical description of the problem involves the following elements:

(a) The data, $(\mathbf{x}_i, y_i) i = 1, \ldots, l$
(b) The function space, G, where the solution is sought
(c) The empirical risk, $I_{\text{emp}}(g)$
(d) The algorithm for the minimization of $I_{\text{emp}}(g)$

The problem definition itself does not pose any restrictions on the decisions described above. For all practical purposes all these choices are arbitrary. That, in fact, demonstrates the ill-posedness of the problem, which results in a variety of solutions depending on the particular tool used and the specific choices that are forced.

The data are usually given a priori. Even when experimentation is tolerated, there exist very few cases where it is known how to construct good experiments to produce useful knowledge suitable for particular model forms. Such a case is the identification of linear systems and the related issue on data quality is known under the term *persistency of excitation* (Ljung, 1987).

The critical decisions in the modeling problem are related to the other three elements. The space G is most often defined as the linear span of a finite number, m, of basis functions, $\theta(\mathbf{x})$, each parametrized by a set of unknown *coefficients* $\mathbf{w}$ according to the formula

$$G = \left\{ g(\mathbf{x}) | g(\mathbf{x}) = \sum_{k=1}^{m} c_k \theta_k(\mathbf{w}, \mathbf{x}), \mathbf{c} \in R^k, \mathbf{w} \in R^n \right\}, \tag{7}$$

where $\mathbf{c}$ is the vector of independent *weights*. All models we consider will be of the form given by Eq. (7). The number of basis functions in G will be referred as the *size of* G. The choice of the basis functions $\theta(\mathbf{x})$ is a problem of *representation*. Any a priori knowledge on the unknown function, $f(\mathbf{x})$, could effectively be exploited to reduce the space G where the model is sought. In most practical situations, however, it is unlikely that such a condition is available. Indeed, in the problem definition we have only assumed $f(\mathbf{x})$ to be continuous and bounded. That sets G equal to the L^∞ space, which, however, is not helpful at all, since the L^∞ space is already too large.

The important question is what restrictions the accuracy and smoothness requirements impose on the function space G, i.e., if smoothness and accuracy (a) can be achieved with any choice of G. (b) Can "optimally" be satisfied in *all* cases for a specific choice of G.

First, with respect to the type of basis functions used in G, smoothness is by no means restrictive. As it is intuitively clear and proved in practice, weird nonsmooth basis functions have to be excluded from consideration but beyond that, all "normal" bases are able to create smooth approximations of the available data. Accuracy is not a constraint either. Given enough basis functions, arbitrary accuracy for the prediction on the data is possible.

Both smoothness and accuracy are possible for given selection of the basis functions. It is, however, clear that a tradeoff between the two

appears with respect to the size of G. Each set of basis functions resolves optimally this tradeoff for different sizes of G. This gives rise to an interesting interplay between the size of G and the number of data points. This is appropriately exemplified by the so-called *bias vs. variance* tradeoff in statistics (Barron, 1994), or *generalization vs. generality* tradeoff in the AI literature (Barto, 1991).

If G is "small," good generalization can be achieved (in the sense that with few degrees of freedom, the behavior of the approximating function in between the data points is restricted) but that introduces bias in the model selection. The solution might look smooth, but it is likely to exhibit poor accuracy of approximation even for the data used to derive it. This is, for example, usually the case when linear approximations are sought to nonlinear functions. Increased accuracy can be achieved by making G large enough. If that is the case, generality can also be achieved, since a large set of functions can potentially approximate well different sets of data. The problem is, however, that in a large G more than one and probably quite different functions in G are able to approximate well or even interpolate perfectly the data set. The variance of the potential solutions is large and the problem can be computationally ill-posed. With such a variety of choices, the risk of deriving a nonsmooth model is large. This is the problem most often encountered with the space of polynomials as basis functions and is also related to the problem of *overfitting*. Data are overfitted when smoothness is sacrificed for the sake of increased accuracy. Because overfitting usually occurs when a large number of basis functions are used for fitting the data, the two have become almost synonymous. It should be noted, however, that this might not always be the case. There can be sets of basis functions (like the piecewise linear Haar basis) that are so poor as approximation schemes, that many of them are required to fit decently (but not overfit) a given set of data. On the other hand, the newly developed fractal-like functions (see, for example, the wavelets created by Daubechies, 1992) provide examples of structures that even in small numbers give rise to approximating functions that make the data look definitely overfitted.

The conclusion is that for every particular set of basis functions and given data, there exists an "appropriate" size of G that can approximate both accurately and smoothly this data set. A decisive advantage would be if there existed a set of basis functions, which could probably represent *any* data set or function with *minimal complexity* (as measured by the number of basis functions for given accuracy). It is, however, straightforward to construct different examples that acquire minimal representations with respect to different types of basis functions. Each basis function for itself is the most obvious positive example. A Gaussian (or discrete points

from a Gaussian) can minimally be represented by the Gaussian itself as the basis function and so on. It is, consequently, hopeless to search for the unique *universal approximator*. The question boils down to defining for each selected basis and data the appropriate size of G that yields both an accurate and smooth approximation of the data.

2. *Sources of the Generalization Error*

The model, $\hat{g}(\mathbf{x})$, that any regression algorithm produces will, in general, be different from the target function, $f(\mathbf{x})$. This difference is expressed by the *generalization error* $\|f(\mathbf{x}) - \hat{g}(\mathbf{x})\|$, where $\|\cdot\|$ is a selected functional norm. The generalization error is a measure of the prediction accuracy of the model, $\hat{g}(\mathbf{x})$, for unseen inputs. Although this error cannot in practice by measured (due to the inductive character of the problem), the identification of its sources is important in formulating the solution to the problem.

There are two sources to the generalization error (Girosi and Anzellotti, 1993):

1. The *approximation error* that stems from the "finiteness" of the function space, G
2. The *estimation error* due to the finiteness of the data.

The unknown function, $f(\mathbf{x})$ is an infinite-dimensional object requiring an infinite number of basis functions for an exact representation. In Fourier analysis, any square-integrable function is exactly represented by an infinite number of trigonometric functions. Similarly, a polynomial of infinite degree is needed for an exact reconstruction of any generic function. On the other hand, the approximating space G, for computations to be feasible, can be spanned only by a finite set of basis functions and characterized only by a finite number of adjustable parameters. Independently of the basis used, the size o G has to be finite. The selection of any function from G [including the "best" approximating function $g^*(\mathbf{x})$] will inevitably introduce some approximation error compared to the target function. Even worse, due to the finiteness of the available data that does not allow the minimization of $I(g)$, a suboptimal solution, $\hat{g}(\mathbf{x})$, rather than $g^*(\mathbf{x})$, will be derived as the model. The difference between $\hat{g}(\mathbf{x})$ and $g^*(\mathbf{x})$ corresponds to the estimation error which further increases the generalization error. To eliminate the approximation error, G has to be infinite-dimensional. Similarly, infinite number of data are required to avoid the estimation error. Both requirements are impractical and that forces us to an important conclusion that the expectation to perfectly reconstruct the real function has somehow to be relaxed.

B. Neural Network Solution to the Functional Estimation Problem

For an introduction to NNs and their functionality, the reader is referred to the rich literature on the subject (e.g., Rumelhart *et al.*, 1986; Barron and Barron, 1988). For our purposes it suffices to say that NNs represent nonlinear mappings formulated inductively from the data. In doing so, they offer potential solutions to the functional estimation problem and will be studied as such.

Mathematically, NNs are weighted combinations of activation functions (units), $\theta(\mathbf{x})$, parametrized by a set of unknown coefficients. In that respect, they fall exactly under the general model form given by Eq. (7). The use of all NNs, for deriving an approximating function $g(\mathbf{x})$ given a set of data, follows a common pattern summarized by the following general NN algorithm:

Step 1. Select a family of basis functions, $\theta(\mathbf{x})$.

Step 2. Use some procedure to define G (the number of basis functions, m, and possibly the coefficients, $\mathbf{w}$).

Step 3. Learning Step: Solve the objective function minimization problem with respect to the weights $\mathbf{c}$ (and the coefficients, $\mathbf{w}$, if not defined in the previous step).

Although the minimization of the objective function might run to convergence problems for different NN structures (such as backpropagation for multilayer perceptrons), here we will assume that step 3 of the NN algorithm unambiguously produces the best, unique model, $\hat{g}(\mathbf{x})$. The question we would like to address is what properties this model inherits from the NN algorithm and the specific choices that are forced.

1. Properties of the Approximations

In recent years some theoretical results have seemed to defeat the basic principle of induction that no mathematical proofs on the validity of the model can be derived. More specifically, the *universal approximation* property has been proved for different sets of basis functions (Hornik *et al.*, 1989, for sigmoids; Hartman *et al.*, 1990, for Gaussians) in order to justify the bias of NN developers to these types of basis functions. This property basically establishes that, for every function, there exists a NN model that exhibits arbitrarily small generalization error. This property, however, should not be erroneously interpreted as a guarantee for small generalization error. Even though there might exist a NN that could

approximate well any given function, once the NN structure is fixed (step 2), the universality of the approximation is lost. Any given NN can approximate well only a limited set of functions, and it is only sheer luck that determines whether the unknown function belongs in that set. On the other hand, the universal approximation property is quite general and can be proved for different sets of basis functions, such as polynomials or sinusoids, even if they are not used in NNs. Basically, every nonlinearity can be used as a universal approximator (Kreinovich, 1991). This conforms with our earlier conclusion that all "normal" basis functions are eligible as approximation schemes, and no one can be singled out as an outstanding tool. As the study of the NN history reveals, the use of most bases in NNs has been motivated by other than mathematical reasons.

2. *Error Bounds*

Furthermore, there exist some theoretical results (Barron, 1994; Girosi, 1993) which provide a priori order of magnitude bounds on the generalization error as a function of the size of the network and the number of the data points. For all these results, however, an assumption on the function space, where the target function belongs, is invariably made. Since it is unlikely that the user will have a priori knowledge on issues like the magnitude of the first Fourier moment of the real function (Barron, 1994) or if the same function is of a Sobolev type (Girosi, 1993), the convergence proofs and error bounds derived under those assumptions have limited practical value. It can be safely concluded that NNs offer no more guarantees with respect to the magnitude of the generalization error than any other regression technique.

The next question is whether the construction of NNs at least guarantees accurate and smooth approximating functions with respect to the available data. The critical issue here is, given a specific selection of the basis functions, how their number is determined (step 2). The established practice reveals that the selection of the NN size is based on either empirical grounds or statistical techniques. These techniques use different aspects of the data, such as the size of the sample or its distribution in the input space, to guide the determination of the network structure. Some techniques, such as the k-means clustering algorithm (Moody and Darken, 1989), are essentially reflections of the empirical rule that the basis functions should be placed in proportion to the data density in the input space. That is an intuitive argument, but also easy to defy. Large density in the input space does not necessarily entail the existence of a complex feature of the underlying function requiring many basis functions to be represented.

It is interesting to notice that the decision on the network size is taken in the absence of any consideration on both the accuracy and the smoothness of the resulting model. On one hand, this might lead to unnecessary complexity and nonsmoothness of the model if the size is larger than needed. This is the case with the regularization approach to radial basis function networks (RBFNs) (Poggio and Girosi, 1989), which assigns a basis function to every data point available. On the other hand, because of the arbitrariness in structuring the network, the accuracy can be poor. The discovery of the network inefficiency, though some validation technique using additional empirical (testing) data, results to a repeated cycle of trial-and-error tests between steps 2 and 3 of the NN algorithm. This is a strong indication that the NN algorithm does not provide in the first place any guarantees of accuracy and smoothness for the approximating function.

Finally, it would be interesting to see if, with an increasing number of data and allowing the NN to find the optimal coefficient values, the real function can ever be closely approximated. Because of the arbitrariness of the NN structuring, the approximation error for the selected network can be arbitrarily large. As long as adaptation of the structure is not allowed, the gap between the model and the real function can never close. The adaptation of the coefficient values (reduction of the estimation error) is certainly not the remedy, if the initial selection of G is inappropriate. It should be noted that there exists some work (e.g., Bhat and McAvoy, 1992) on the problem of optimally structuring the network. However, the problem is dealt at the initial stage (step 1) and the proposed solutions do not include any considerations for structural adaptation at the learning stage (step 3). In most of the cases, the definition of the network architecture involves few initial data points and the solution of a considerable optimization problem that is difficult to duplicate in an on-line fashion when more data are available. As long as the network structuring becomes a static (with respect to the data) decision, the preceding considerations still hold independently of the method used for determining the NN structure.

III. Solution to the Functional Estimation Problem

A. Formulation of the Learning Problem

The analysis of NNs has shown that, in order to assure both accuracy and smoothness for the approximating function, the solution algorithm will have to allow the model (essentially its size) to evolve *dynamically* with the

data. This calls not only for a new algorithm but also for a new problem definition. The new element is the stepwise presentation of the data to the modeling algorithm and the dynamic evolution of the solution itself. All elements in the problem have to be indexed by a time-like integer l, which will correspond to the amount of data available and that, by abuse of notation, will be referred as the *instant*. The new problem is called the *learning problem* and is defined as follows:

Learning Problem. At every instant l let $Z_l = \{(\mathbf{x}_i, y_i) \in R^{k\times 1} i = 1, 2, \ldots, l\}$ be a set of data drawn with some unknown probability distribution, $P(\mathbf{x}, y)$ and G_l, a space of functions. Find a function $\hat{g}_l(\mathbf{x})$: $X \to Y$ belonging to G_l that minimizes the empirical risk $I_{\text{emp}}(g)$.

There is a "time" continuity of all the variables in the problem, which is expressed by the following relations:

(a) $Z_l = Z_{l-1} \cup \{(\mathbf{x}_l, y_l)\}$
(b) $G_l \supseteq G_{l-1}$
(c) $G_l = A(G_{l-1})$, where A is the algorithm for the adaptation of the space, G_l
(d) $\hat{g}_l(\mathbf{x}) = M(Z_l, G_l)$, where M is the algorithm for the generation of the model, $\hat{g}_l(\mathbf{x})$

The first relation represents the data buildup, while the next two reassure that the space G_l evolves over its predecessor, G_{l-1} according to algorithm A. At every instant, the model results by applying algorithm M on the available data, Z_l, and the chosen input space, G_l.

With such an approach to the functional estimation problem, we escape from the cycle of deriving models and testing them. With every amount of data available, the simplest and smoothest model is sought. Every new data point builds constructively on the previous solution and does not defy it, as it is the case with statistical techniques such as cross-validation. The expectation is that eventually a function close to the real one will be acquired. The procedure is ever-continuing and is hoped to be ever-improving. Not unjustifibly, we call such a procedure *learning* and the computational scheme that realizes it, a *learning scheme*. The learning scheme consists of four elements:

1. A dynamic memory of previous data, Z_l
2. An approximating function or model, $\hat{g}_l(\mathbf{x})$
3. A built-in algorithm A for the adaptation of the space, G_l
4. A built-in algorithm M for the generation of the model, $\hat{g}(\mathbf{x})$

Independently of the amount of data and the way they are acquired, the definition of the learning problem simulates a continuous data flow. Such

a framework is of particular importance for the cases where the data are indeed available sequentially in time. Process modeling for adaptive control is one important application, which fits perfectly into this framework. The very nature of feedback control relies on and presupposes the explicit or implicit measurement of the process inputs and outputs; therefore, a continuous *teacher* to the process model is available. Accurate process modeling is essential for control. However, process identification is a hard problem. The derivation of a suitable mechanism which can fully exploit the continuous flow of useful information has been our main motivation for developing the learning framework itself and its solution.

1. Derivation of Basic Algorithm

The solution of the learning problem requires the construction of an algorithm, A, for adaptation of the structure and that in turn presupposes the selection of a proper criterion, which will indicate the need for adaptation. Unlike NNs, such a criterion will have to be based on the empirical error to avoid the dangers of overdetermining the size of the network, which might cause overfitting, or underdetermining it, which will result in large approximation error. The framework calls for an appropriately defined *threshold* on the empirical error, as the criterion which will serve as a guide for the structural adaptation.
We are now ready to propose the basic algorithm for the solution of the learning problem.

Basic Algorithm

Step 1. Select a family of basis functions, $\theta(\mathbf{x})$.
Step 2. Select a form for the empirical risk, $I_{\text{emp}}(g)$, and establish a threshold, ε, on the empirical error.
Step 3. Select a minimal space, G_0.
Step 4. Learning step: For every available data point, apply algorithm M to calculate the model and estimate the empirical error on all available data. If it exceeds the defined threshold, use algorithm A to update the structure of G until threshold is satisfied.

In the remaining sections, specific answers will be given for the selection of the

(a) Functional space, G.
(b) Basis functions $\theta(\mathbf{x})$.
(c) The empirical risk, $I_{\text{emp}}(g)$, and the corresponding error threshold, ε.

With every specification of the above parameters, the basic algorithm will be refined and its properties will be studied, until the complete algorithm is revealed. However, each of the presented algorithms can be considered as a point of departure, where different solutions from the finally proposed can be obtained.

2. *Selection of the Function Space*

The solution to the learning problem should provide the flexibility to search for the model in increasingly larger spaces, as the inadequacy of the smaller spaces to approximate well the given data are proved. This immediately calls for a hierarchy in the space of functions. Vapnik (1982) has introduced the notion of *structure* as an infinite ladder of finite-dimensional nested subspaces:

$$S_0 \subset S_1 \subset S_2 \subset \cdots S_n \subset \cdots. \tag{8}$$

For reasons that will become clear later on, we will refer to the index j of each subspace as the *scale*. For continuity to be satisfied, the basis functions that span each subspace S_j have to be a superset of the set of basis functions spanning the space S_{j-1}. Vapnik solves the minimization problem in each subspace and uses statistical arguments to choose one as the final solution to the approximation problem for a given set of data. He calls his method *structural minimization*. Although we have adopted his idea of the structure given by Eq. (8), we have implemented it in a fairly different spirit. At every instant, the space G_l, where the model is sought, is set equal to one of the predefined spaces S_j. With more data available and depending on the prediction accuracy, the model is sought in increasingly larger spaces. In this way, we establish a strictly forward move into the structure and allow the incoming data dictate the pace of the move.

One example of a structure (8) is the space of polynomials, where the ladder of subspaces corresponds to polynomials of increasing degree. As the index j of S_j increases, the subspaces become increasingly more complex where complexity is referred to the number of basis functions spanning each subspace. Since we seek the solution at the lowest index space, we express our bias toward simpler solutions. This is not, however, enough in guaranteeing smoothness for the approximating function. Additional restrictions will have to be imposed on the structure to accommodate better the notion of smoothness and that, in turn, depends on our ability to relate this intuitive requirement to mathematical descriptions.

Although other descriptions are possible, the mathematical concept that matches more closely the intuitive notion of smoothness is the *frequency* content of the function. Smooth functions are sluggish and coarse and characterized by very gradual changes on the value of the output as we scan the input space. This, in a Fourier analysis of the function, corresponds to high content of low frequencies. Furthermore, we expect the frequency content of the approximating function to vary with the position in the input space. Many functions contain high-frequency features dispersed in the input space that are very important to capture. The tool used to describe the function will have to support *local* features of *multiple resolutions* (variable frequencies) within the input space.

Smoothness of the approximating function with respect to the data forces analogous considerations. First, we expect the approximating function to be smoother where data are sparse. Complex behavior is not justified in a region of the input space where data are absent. On the other hand, smoothness has to be satisfied in a dynamic way with every new data point. The model adaptation, triggered by any point, should not extend beyond the neighborhood of "dominance" of the new data point and, at the same time, should not be limited to a smaller area. It is not justifiable to change the entire model every time one point is not predicted accurately by the model, nor is it meaningful to induce so local changes that only a single point is satisfied. Again the need for *localization both in space and frequency* shows up. This is not surprising, since the very nature of the problem is a game between the localization of the data and the globality of the sought model. The problem itself forces us to think in multiple resolutions.

Here is how we can achieve a multiresolution decomposition of the input space. Let $I_0 = [a, b]^k$ be the entire bounded input space according to the problem definition. In each dimension, the interval $[a, b]$ is divided into two parts that are combined into 2^k subregions, $I_{1,\mathbf{p}}$, where $\mathbf{p}$ is k-dimensional vector that signifies the position of the subregion and whose elements take the values 1 or 2. Let I_1 signify the set of all these subregions: $I_1 = \{I_{1,\mathbf{p}} | \mathbf{p} = (q, r), q, r = 1 \text{ or } 2\}$, which will be referred as the scale 1 *decomposition* of the input space. The union of all these subregions forms the entire input space: $\cup I_{1,\mathbf{p}} = I_0$. the subdivision of each subregion in I_1 follows the same pattern, resulting in a *hierarchy* of decompositions $I_0, I_1, I_2, \ldots$ each corresponding to a division of the input space in a different resolution. An example of the resulting ladder of subregions is given in Fig. 2 for the two-dimensional case.

There exists a one-to-one correspondence between the structure of subspaces and the hierarchy of input space decompositions. This already

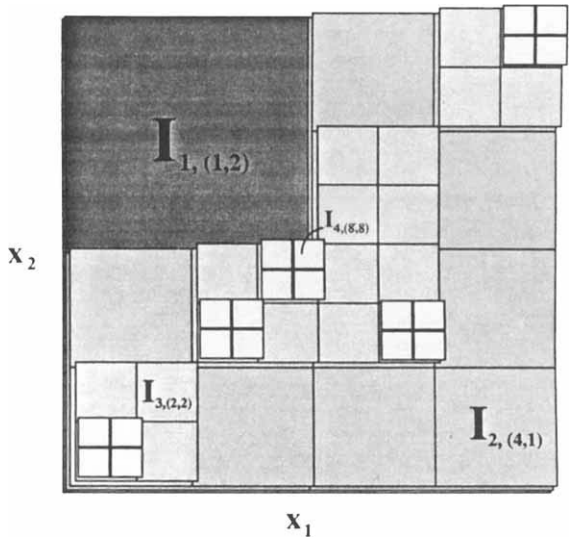

FIG. 2. Multiresolution decomposition of the input space.

poses a restriction on the basis functions that span the subspaces. They have to be local and have bounded support. Such a requirement excludes popular in regression functional bases such as polynomials and sigmoids that are global. Each subregion in I_j corresponds to the support of a basis function in S_j. In this way, the problem is decomposed into smaller subproblems where it is solved almost independently depending on the degree of overlap of the functions spanning those regions. The extreme case is when there is no overlap such as when piecewise approximation in each subregion is used. In this analysis, we have adopted for simplicity a predefined dyadic subdivision of the input space. However, this is not a unique choice. Different formulas can be invented to break the dyadic regularity and even formulate the grid of subregions not in a predefined way but in parallel with the data acquisition. The additional requirement for the basis functions used is the construction of the basis to support this localized multiresolution structure.

Given a multiresolution decomposition of the input space, the algorithm A for the adaptation of the model structure, identifies the region $I_{j+1,\mathbf{p}}$ in the next scale where the incoming data point belongs and adds the corresponding basis function that spans this region. With this addition, the basic algorithm can now be modified as follows.

Algorithm 1

Step 1. Select a family of basis functions, $\theta(\mathbf{x})$, supporting a multiresolution decomposition of the input space.
Step 2. Select a form for the empirical risk, $I_{\text{emp}}(g)$, and establish a threshold, ε, on the empirical error.
Step 3. Select a minimal space, G_0, and input space decomposition, I_0.
Step 4. Learning step: For every available data point, apply algorithm M to calculate the model and estimate the empirical error on all available data. If it exceeds the defined threshold, use algorithm A to update the structure of G, by inserting the lowest scale basis function that includes the new point. Repeat until threshold is satisfied.

3. *Selection of the Error Threshold*

Algorithm 1 requires the a priori selection of a threshold, ε, on the empirical risk, $I_{\text{emp}}(g)$, which will indicate whether the model needs adaptation to retain its accuracy, with respect to the data, at a minimum acceptable level. At the same time, this threshold will serve as a termination criterion for the adaptation of the approximating function. When (and if) a model is reached so that the generalization error is smaller than ε, learning will have concluded. For that reason, and since, as shown earlier, some error is unavoidable, the selection of the threshold should reflect our preference on how "close" and in what "sense" we would like the model to be with respect to the real function.

Given a space G, let $g^*(\mathbf{x})$ be the "closest" model in G to the real function, $f(\mathbf{x})$. As it is shown in Appendix 1, if $f \in G$ and the L^∞ error measure [Eq. (4)] is used, the real function is also the best function in G, $g^* = f$, independently of the statistics of the noise and as long as the noise is symmetrically bounded. In contrast, for the L^2 measure [Eq. (3)], the real function is not the best model in G if the noise is not zero-mean. This is a very important observation considering the fact that in many applications (e.g., process control), the data are corrupted by non-zero-mean (load) disturbances, in which cases, the L^2 error measure will fail to retrieve the real function even with infinite data. On the other hand, as it is also explained in Appendix 1, if $f \notin G$ (which is the most probable case), closeness of the real and "best" functions, $f(\mathbf{x})$ and $g^*(\mathbf{x})$, respectively, is guaranteed only in the metric that is used in the definition of $I(g)$. That is, if $I(g)$ is given by Eq. (3), $g^*(\mathbf{x})$ can be close to $f(\mathbf{x})$ only in the L^2-sense and similarly for the L^∞ definition of $I(g)$. As is clear, L^2

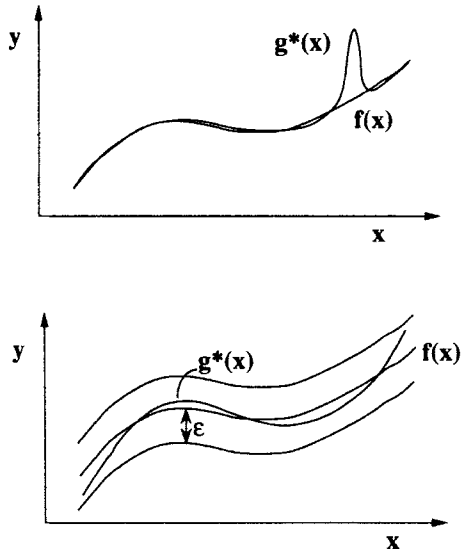

FIG. 3. L^2 vs. L^∞ closeness of functions.

closeness does not entail L^∞ closeness and vice versa. The question now is what type of closeness is preferable.

Figure 3 gives two examples of L^2 and L^∞ closeness of two functions. The L^2 closeness leaves open the possibility that in a small region of the input space (with, therefore, small contribution to the overall error) the two functions can be considerably different. This is not the case for L^∞ closeness, which guarantees some minimal proximity of the two functions. Such a proximity is important when, as in this case, one of the functions is used to predict the behavior of the other, and the accuracy of the prediction has to be established on a pointwise basis. In these cases, the L^∞ error criterion (4) and its equivalent [Eq. (6)] are superior. In fact, L^∞ closeness is a much stricter requirement than L^2 closeness. It should be noted that whereas the minimization of Eq. (3) is a quadratic problem and is guaranteed to have a unique solution, by minimizing the L^∞ expected risk [Eq. (4)], one may yield many solutions with the same minimum error. With respect to their predictive accuracy, however, all these solutions are equivalent and, in addition, we have already retreated from the requirement to find the one and only real function. Therefore, the multiplicity of the "best" solutions is not a problem.

There is one additional reason why the L^∞ empirical risk is a better objective function to use. With the empirical risk given by Eq. (6), which is by definition a pointwise measure, it is clear how to define in practice the

numerical value of the threshold that will be applied to every data point to assess the model accuracy. That would not be the case if the L^2 error were used whose absolute magnitude is difficult to correlate with the goodness of the approximation.

In light of the previous discussion and contrary to the established practice, we propose the use of the maximum absolute error (6) as the empirical risk to be minimized because it offers the following advantages:

1. As long as the noise is symmetrically bounded, the L^∞ error is minimized by the true function, $f(\mathbf{x})$, independently of the statistics of the noise.
2. Minimization of the L^∞ error ensures L^∞ closeness of the approximating function to the real function.
3. It is straightforward to define a numerical value for the threshold on the L^∞ error.

The first two points raise the question of convergence of the model to the real function, or its best model, $g^*(\mathbf{x})$. In the following analysis we will address the convergence question, first assuming that the model with the desired accuracy lies in a known space G and then when this space is attempted to be reached with the hierarchical adaptation procedure of Algorithm 1.

Given a choice for G, the question is, if and under which conditions the model, $\hat{g}(\mathbf{x})$, converges to $g^*(\mathbf{x})$ as the number of points increases to infinity, or, in other words, if it is possible to completely eliminate the estimation error. The answer will emerge after addressing the following questions.

4. As the Sample Size Increases, Does $I_{emp}(g)$ Converge to $I(g)$?

Under very general conditions, it is true that the empirical mean given by Eq. (5), converges to the mathematical expectation (3), as $l \to \infty$ (Vapnik, 1982). On the other hand, it is clear that

$$\max_{i=1,\ldots,l} |y_i - g(\mathbf{x}_i)| \to \sup_x |y - g(\mathbf{x})| \quad \text{as} \quad l \to \infty$$

and convergence of $I_{\text{emp}}(g)$, given by Eq. (4) to $I(g)$, given by Eq. (6), is also guaranteed.

5. Does Convergence of $I_{emp}(g)$ to $I(g)$ Entail Convergence of $\hat{g}$ to g^*?

If the answer were positive, that would ensure that, as we minimize the $I_{\text{emp}}(g)$ and more data become available, we approach the best approximating function, $g^*(\mathbf{x})$. Unfortunately, in general, this is not true and that

means that as more data become available the estimation error does not necessarily decrease. Convergence can, however, be proved in a weaker sense for the L^∞ case, if we retreat from the requirement to acquire $g^*(\mathbf{x})$ at infinity and be satisfied with *any* solution that satisfies an error bound in its L^∞-norm distance from the real function $f(\mathbf{x})$.

Theorem 1. *Let $\varepsilon > 0$ and $G_\varepsilon = \{g \in G | I(g) < \varepsilon\}$ be a nonempty subset of G and ε, a small positive number. The minimization of the empirical risk [Eq. (6)] will converge in G_ε as $l \to \infty$.*

The proof is given in Appendix 2. The theorem basically states that if there exists a function in G that satisfies $I(g) < \varepsilon$, then by minimizing the L^∞ error measure, this function will be found. The generalization error for this function is guaranteed to be less than $\varepsilon - \delta$. The theorem, however, presupposes the existence of the function with the desired accuracy. This, in general, cannot be guaranteed a priori. In the following, we will prove that the adaptation methodology developed for Algorithm 1 will converge to a subspace where the existence of such a function is guaranteed. For the proof, we have to make the following assumptions for the ladder of subspaces defined by Eq. (8):

Assumption 1. At every subspace S_{j2} there exists a better approximating function than any function in S_{j1} with $j_2 > j_1$:

$$\forall j_2 > j_1, \exists g' \in S_{j2} \quad \text{such that} \quad I(g') < I(g) \; \forall g \in S_{j1}.$$

Assumption 2. There exists a subspace S_{j^*} with $j^* < \infty$ where a function with the desired approximation accuracy exists:

$$\exists j^* \quad \text{and} \quad g^* \in S_{j^*} \quad \text{such that} \quad I(g^*) < \varepsilon,$$

or, in other words, G_ε in S_{j^*} is nonempty.

On the basis of these assumptions we can prove the following theorem:

Theorem 2. *Algorithm 1 will converge to S_{j^*}, and no further structural adaptation will be needed for any additional data points.*

The proof is given in Appendix 3. Theorem 2 indicates that by forcing $I_{\text{emp}}(g)$ to be less than ε at all times and by following Algorithm 1, we are guaranteed to reach a subspace S_{j^*} where a solution with $I(g) < \varepsilon$ exists. According to Theorem 1, the algorithm will converge into the set G_ε and all subsequent solutions to the minimization of $I_{\text{emp}}(g)$ will obey $I(g) < \varepsilon$ which is the criterion we want to be satisfied. The generalization error will

be $\varepsilon - \delta$. Therefore, by defining an error threshold $\varepsilon + \delta$, we will eventually reach a solution with a generalization error less than ε. The signficance of Algorithm 1 according to Theorem 2, is that it provides a formal way to screen out functional spaces where solutions with the desired accuracy are not present, and that it eventually concludes on the smaller subspace where such a solution exists. This result is possible by the use of the L^∞ error measure and the structural adaptation procedure. It should be reminded that if the space G were to be determined a priori and statically (as it is the case for NNs), such a convergence would not be possible unless, luckily, that space already contained a function with the desired accuracy.

It should be noted, however, that the convergence result does not entail a monotonic decrease of the generalization error. The algorithm moves to spaces where potentially better solutions can be obtained, but the solution that it chooses is not necessarily better than the previous one. This is too much to ask and indeed no approximation scheme, whether it applies structural adaptation or not, can guarantee strict improvement as the number of data points increases to infinity. On the other hand, no reassurance on the number of data needed for convergence in S_{j^*} and G_ε are provided. This issue is related to the data quality, i.e., the degree to which the incoming data reveal the inadequacies of the existing model. If the data are not "good enough," convergence can be delayed beyond the point where data are available. In this case, however, the solution is still satisfactory as long as, by construction, it serves properly the accuracy—smoothness tradeoff, which, as stated earlier, is the only criterion in assessing the goodness of the approximation.

We can now revise Algorithm 1 with the definition of the error threshold.

Algorithm 2

Step 1. Select a family of basis functions, $\theta(\mathbf{x})$, supporting a mutliresolution decomposition of the input space.

Step 2. Select a threshold ε, with $\varepsilon > 0$ and $\varepsilon > \delta$, on the empirical error defined by Eq. (6).

Step 3. Select a minimal space, G_0, and input space decomposition, I_0.

Step 4. Learning step: For every available data point, apply algorithm M to calculate the model by solving the minimization problem, $\hat{g} = \min_g I_{\text{emp}}(g) = \min_g \max_{i=1,\ldots,l} |y_i - g(x_i)|$, $g_j \in S_j$, and estimate the empirical error on all available data, $I_{\text{emp}}(g)$. If $I_{\text{emp}}(g) > \varepsilon$, use algorithm A to update the structure of G, by inserting the lowest scale basis function that includes that point. Repeat until threshold is satisfied.

6. Representation of the Functional Spaces

The only element of the learning algorithm that remains undetermined is the basis functions that span the subspaces according to Eq. (8). The basic requirement for the basis functions are to be local, smooth and to have adjustable support to fit the variable size subregions of the input space. The Gaussian function, for example, with variable mean and variance has all the required properties. The convergence theorems stated in the previous section, however, show that the basis used is also required to exhibit well-defined properties as an approximation scheme. This stems from the requirements (implied by Assumptions 1 and 2) that, as we move on to higher index subspaces, we increase our ability to fit better not only the data (which is the easy part) but the unknown function as well. This is a major difference of this approach with other algorithms, such as the resource-allocating network (RAN) algorithm (Platt, 1991) where on-line adaptation of the structure is performed but not in a well-defined framework that will guarantee improved approximation capabilities of the augmented network. The additional approximation properties are sought in the framework of *multiresolution analysis* and the *wavelet representation.* For details on both these tools, the reader is referred to the work of Mallat (1989), Daubechies (1992), Strang (1989), and also Bakshi and Stephanopoulos (1993) for a NN-motivated treatment of wavelets. Here, we will state briefly the basic facts that prove that the wavelet transform indeed fits the developed framework and will emphasize more its properties as an approximation scheme.

a. Multiresolution Analysis. Let $f(x) \in L^2(R)$ be a one-dimensional function of finite energy. Following Mallat (1989), a multiresolution analysis of $f(x)$ is defined as a sequence of successive approximations of $f(x)$ resulting from the projection of $f(x)$ on finer and finer subspaces of functions, S_j. Le $f(x)$ be the approximation of $f(x)$ at resolution 2^j or scale j. As the scale decreases, $f^j(x)$ looks coarser and smoother and devoid of any detail of high frequency. Inversely, with the scale increasing, the approximation looks finer and finer until the real function is recovered at infinite scale:

$$\lim_{j \to \infty} f^j(x) = f(x). \tag{9}$$

Multiresolution analysis conforms with the definition of structure (8). More importantly, it guarantees that by moving to higher subspaces (scales), better approximations of the unknown functions can potentially be obtained, which is the additional property sought.

Regarding the representation of the subspaces, S_j, there exists a unique function $\phi(x)$, called the *scaling function*, whose translations and dilations span those subspaces:

$$S_j = \left\{ \sum_n c_{jn} \phi_{jn}(x) | j, n \in Z, c_{jn} \in R \right\}, \quad \text{where}$$

$$\phi_{jn}(x) = \sqrt{2^j}\, \phi(2^j x - n) \tag{10}$$

The scaling function, $\phi(x)$, has either local support or decays very fast to zero. For all practical purposes, it is a local function. By translating and dilating that function we are able to cover the entire input space in multiple resolutions, as it is required.

The framework, however, as introduced so far is of little help for our purpose since the shift from any subspace to its immediate in hierarchy would require to change entirely the set of basis functions. Although $\phi_{jn}(x)$ are all created by the same function, they are different functions and, consequently, the approximation problem has to be solved from scratch with any change of subspace. The theory of wavelets and its relation to multiresolution analysis provides the ladder that allows the transition from one space to the other.

It can be proved that the approximation $f^{j+1}(x)$ of $f(x)$ in scale $j+1$ can be written as a combination of its approximation at the lower scale j with additional detail represented by the wavelet transform at the same scale:

$$S_{j+1} = \left\{ \sum_n c_{jn} \phi_{jn}(x) + \sum_n d_{jn} \psi_{jn}(x) | j, n \in Z, c_{jn}, d_{jn} \in R \right\}, \quad \text{where}$$

$$\psi_{jn}(x) = \sqrt{2^j}\, \psi(2^j x - n) \tag{11}$$

and $\psi(x)$ is the *wavelet function* determined in unique correspondence to the scaling function. With this addition, the shift to higher scales for improved approximation involves the addition of a new set of basis functions that are the wavelets at the same scale and whose purpose is to capture the high-frequency detail ignored in the previous scale. This analysis can be easily extended to dimensions higher than one.

There exist different pairs of wavelets and scaling functions. One such pair is shown in Fig. 4. This is the "Mexican hat" pair (Daubechies, 1992), which draws its name by the fact that the scaling function looks like the

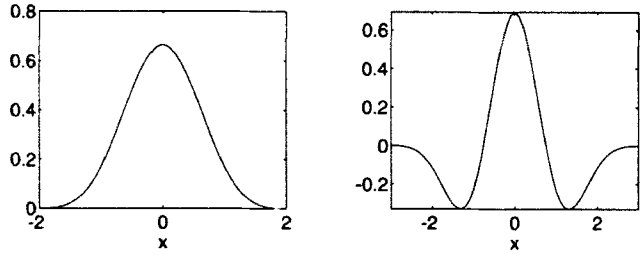

FIG. 4. "Mexican hat" scaling function and wavelet.

Gaussian and the wavelet like the Gaussian second derivative ("Mexican hat" function).

b. Wavelet Properties. All wavelets and scaling functions share the property of *localization in space and in frequency*, as required. Although such a property (in the weak sense that we are using it as a means to represent the smoothness requirement) can be satisfied with other functions as well, it is only with wavelets that it finds an exact mathematical description. The Fourier transform of the wavelets is a bandpass filter. The higher the scale, the more the passband extends over higher frequencies. That explains how the wavelets of high scales are able to capture the fine, high-frequency detail of any function they represent. The combination of space and frequency localization is a unique property for bases of $L^2(R)$ and it is what made wavelets a distinguished mathematical tool for applications where a multiresolution decomposition of a function or a signal is required.

As approximation schemes, wavelets trivially satisfy the Assumptions 1 and 2 of our framework. Both the L^2 and the L^∞ error of approximation is decreased as we move to higher index spaces. More specifically, recent work (Kon and Raphael, 1993) has proved that the wavelet transform converges uniformly according to the formula

$$\sup_x |f^j(x) - f(x)| \leq C2^{-jp}, \tag{12}$$

where j is the scale, C a constant, and p the order of the approximation. The latter corresponds to the number of vanishing moments of the wavelet (Strang, 1989):

$$\int x^m \psi(x)\, dx = 0 \qquad \text{for } m = 0, \ldots, p-1, \tag{13}$$

and is an important indicator of the accuracy of each wavelet as an

approximator. As can be easily verified by Eq. (12), the larger the number of vanishing moments, the faster the wavelet transform converges to the real function. The approximation error decreases with the scale and for any arbitrary small number ε there exists a scale $j^* = -(1/p)\log_2(\varepsilon/C)$, where the approximation error is less than ε. To use the NN terminology, wavelets are universal approximators since any function in $L^2(R^k)$ can be approximated arbitrarily well by its wavelet representation. Wavelets, however, offer more than any other universal approximator in that, they provide, by construction, a systematic mechanism (their hierarchical structure) to achieve the desired accuracy.

The different pairs of wavelet and scaling functions are not equivalent as approximators. Besides the number of vanishing moments, another distinguishable property is that of orthogonality. Orthogonality results in considerable simplifications in the structural adaptation procedure, since every new orthogonal wavelet inserted in the approximating function introduces independent information. From the approximation point of view, orthogonality is equivalent to more compact and, therefore, more economical representations of the unknown functions. There is, however, a price to pay. With the exception of the discontinuous Haar wavelet, there do not exist wavelets that are compactly supported, symmetrical and orthogonal. In addition, all orthogonal and compactly supported wavelets are highly nonsmooth and, therefore, are not suitable for approximation problems. For this reason, *biorthogonal* and *semiorthogonal* wavelets have been constructed (Cohen, 1992; Feauveau, 1992) that retain the advantages of orthogonality and, at the same time, satisfy the smoothness requirement.

The space–frequency localization of wavelets has lead other researchers as well (Pati, 1992; Zhang and Benveniste, 1992) in considering their use in a NN scheme. In their schemes, however, the determination of the network involves the solution of complicated optimization problem where not only the coefficients but also the wavelet scales and positions in the input space are unknown. Such an approach evidently defies the on-line character of the learning problem and renders any structural adaptation procedure impractical. In that case, those networks suffer from all the deficiencies of NNs for which the network structure is a static decision.

B. Learning Algorithm

With the selection of wavelets as the basis functions the learning algorithm can now be finalized.

Algorithm 3

Step 1. Select a family of scaling functions and wavelets.
Step 2. Select a threshold, ε, with $\varepsilon > 0$ and $\varepsilon > \delta$, on the empirical error defined by Eq. (6).
Step 3. Select an initial scale, j_0.
Step 4. Learning step: For every available data point, apply algorithm M to calculate the model by solving the minimization problem, $\hat{g} = \min_g I_{\text{emp}}(g) = \min_g \max_{i=1,\ldots,I} |y_i - g(\mathbf{x}_i)|$, $g_j \in S_j$, and estimate the empirical error on all available data, $I_{\text{emp}}(g)$. If $I_{\text{emp}}(g) > \varepsilon$, use algorithm A to update the structure of G, by inserting the lowest scale wavelet that includes that point. Repeat until threshold is satisfied.

The algorithm is schematically presented in Fig. 5. The implementation of the algorithm presupposes the a priori specification of

(a) The scaling function and wavelet pair.
(b) The initial scale j_0.
(c) The value of the error threshold, ε.

For the moment, there are no guidelines for the selection of the particular basis functions for any given application. The important issue here is that the properties of the wavelets will be inherited by the approximating

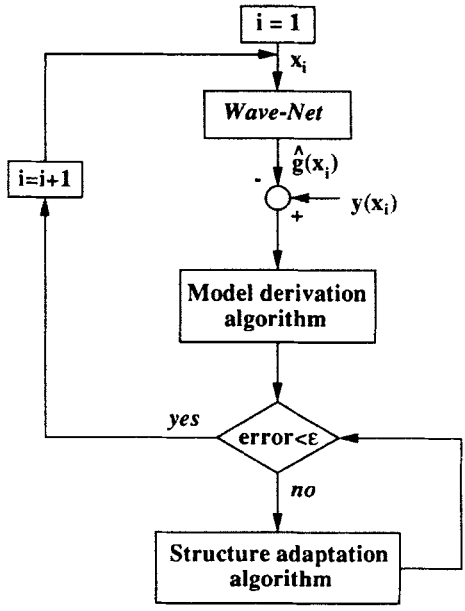

FIG. 5. Learning algorithm.

function and, therefore, if there exist prerequisites on the form of the approximating function (such as the number of existing derivatives), they should be taken into account before the selection of the wavelet. The initial scale, where the first approximation will be constructed, can be chosen as the scale in which few scaling functions (even one) fit within the input range, I_0, of the expected values of $\mathbf{x}$. Finally, the selection of the error bound depends on the accuracy of the approximation sought for the particular application. In any case, it should be equal or greater than the bound on the expected noise, δ.

1. The Model and Its Derivation

The approximating function constructed by the previous algorithm is of the form

$$g(x) = \sum_{k=1}^{m} c_k \theta_k(\mathbf{x}), \tag{14}$$

where $\theta(\mathbf{x})$ stands for both the scaling functions at scale j_0 and wavelets at the same or higher scales. Equation (14) can be easily identified as the mathematical representation of the *Wave-Net* (Bakshi and Stephanopoulos, 1993). With the use of the L^∞ error measure, the coefficients are calculated by solving the minimization problem:

$$\mathbf{c} = \arg\min \max_{i=1,\ldots,l} \left| y_i - \sum_{k=1}^{m} c_k \theta_k(\mathbf{x}_i) \right|. \tag{15}$$

The problem [Eq. (15)] is a minimax optimization problem. For the case (as it is here) where the approximating function depends linearly on the coefficients, the optimization problem [Eq. (15)] has the form of the *Chebyshev approximation* problem and has a known solution (Murty, 1983). Indeed, it can be easily shown that with the introduction of the dummy variables z, z_i, z_i^* the minimax problem can be transformed to the following linear program (LP):

$$\begin{aligned} &\min_{\mathbf{c}} z, \\ &z_i = -y_i + \sum_{k=1}^{m} c_k \theta_k(\mathbf{x}_i) + z \geq 0, \\ &z_i^* = y_i - \sum_{k=1}^{m} c_k \theta_k(\mathbf{x}_i) + z \geq 0. \end{aligned} \tag{16}$$

The transformation to this LP program is graphically depicted in Fig. 6 for the case when $\mathbf{c}$ is a scalar. For each data pair $(\mathbf{x}_i, y_i)$ the term $|y_i - \Sigma c_k \theta_k(\mathbf{x}_i)|$ represents two $(m+1)$-dimensional hyperplanes, $z = y_i - \Sigma c_k \theta_k(\mathbf{x}_i)$ and $z = -y_i + \Sigma c_k \theta_k(\mathbf{x}_i)$. For the scalar case, these correspond

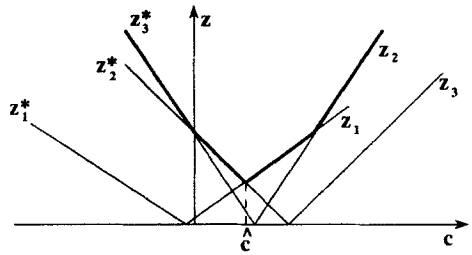

FIG. 6. Geometric interpretation of the Chebyshev approximation problem.

to two lines of opposite slope that meet on the c axis as shown in Fig. 6. For every value of c, according to problem (15), we need to identify the data pair whose corresponding hyperplane has the maximum value, z. The locus of these points is the polyhedron that inscribes the common space above each hyperplane. If an LP formulation, the space above every hyperplane corresponds to an inequality constraint according to Eq. (16), and the intersection of all these constraints is the feasible region. The bottom of the convex polyhedron that inscribes the feasible region is the value of c that minimizes both Eqs. (15) and (16). Its distance from c axis is the minimum value of z, or the minimum possible error. The optimization problem is convex and is guaranteed to have a solution. It can be easily solved using standard LP techniques based on the dual simplex algorithm.

The selection to minimize absolute error [Eq. (6)] calls for optimization algorithms different from those of the standard least-squares problem. Both problems have simple and extensively documented solutions. A slight advantage of the LP solution is that it does not need to be solved for the points for which the approximation error is less than the selected error threshold. In contrast, the least squares problem has to be solved with every newly acquired piece of data. The LP problem can effectively be solved with the dual simplex algorithm, which allows the solution to proceed recursively with the gradual introduction of constraints corresponding to the new data points.

2. *Variations on the Structural Adaptation Algorithm*

The multiresolution framework allows us to reconsider more constructively some of the features of the structure adaptation algorithm. First, a strictly forward move in the ladder of subspaces [Eq. (8)] is not necessary. Due to localization, the structural correction can be sought in higher spaces before all functions in the previous space are exhausted. This

results in an uneven distribution of variable scale features in the input space, which, in turn, means considerable savings in the number of basis functions used.

In a k-dimensional space, one way to construct the scaling function and the wavelets is by taking the k-term tensor products of the one-dimensional parts. For example, in two-dimensional (2D) space the scaling function is $\phi(x_1)\phi(x_2)$ and there are three wavelets: $\phi(x_1)\psi(x_2)$, $\psi(x_1)\phi(x_2)$, and $\psi(x_1)\psi(x_2)$. This result in $2^k - 1$ wavelets all sharing the same support, $I_j, \mathbf{p}$, in the input space. Every time adaptation of the structure is needed at a given position and scale, there are $2^k - 1$ choices and a selection problem arises. By solving locally a small optimization problem, we can identify the wavelet whose insertion in the model minimizes the empirical error and use this one to update the structure. The fact, however, that we are not seeking a complete wavelet representation allows us to consider alternatively empirical selection methods that are not computationally expensive. For example, we might choose to use only the one symmetrical wavelet $\psi(x_1)\psi(x_2)\ldots\psi(x_k)$ at every scale and ignore all the others. On the other hand, there exist methods for constructing high-dimensional wavelets that do not use the tensor products (*nonseparable* wavelets; Kovačević and Vetterli, 1992) and, therefore, automatically overcome this selection problem.

Another variation on the algorithm can be applied by allowing the freedom to select the initial space S_0 outside the multiresolution framework. In many cases, a model of the unknown function is already available by first-principles modeling, or there might exist some a priori knowledge (or bias) on the structure of the unknown function (linear, polynomial, etc.). In all these cases, we would like to combine the available knowledge with the NN model to reduce the blackbox character of the latter and increase its predictive power. There has been some recent work in this direction (Psichogios and Ungar, 1992; Kramer *et al.*, 1992). The multiresolution framework allows us to naturally unify the a priori model with the NN by allowing us to select the initial space S_0 in accordance to our preference. Then, the model in S_0 provides an initial global representation of the unknown function, and the learning algorithm is used to expand and correct locally this model. This procedure is in agreement with the notion of learning as a way to extend what is already known, and is also of considerable practical value.

3. *Derivation of Error Bounds*

As shown earlier, by imposing a threshold on the L^∞ empirical error and applying the learning algorithm, an approximating function with

generalization error less that the threshold will eventually be reached. Therefore, the selected threshold can be regarded as a *global* measure of the approximating capability of the derived model. The multiresolution decomposition of the input space, however, allows us to impose *local* error thresholds and, therefore, seek for variable approximation accuracy in different areas of the input space according to the needs of the problem.

The error threshold can be allowed to vary not only on a spatial but also on a temporal basis as well. Sometimes, the bound on the magnitude of the noise, δ, is not easy to be known, in which cases, the value of the threshold (which must be greater than δ) cannot be determined. In some other situations, it might be risky to impose a tight threshold from the beginning when relatively few data points are available because this might temporarily lead to overfitting. An empirical but efficient way to alleviate those problems is to enforce the error threshold gradually. In other words, we can devise some sequence of error thresholds that satisfy the relationship

$$\varepsilon_1 > \varepsilon_2 > \cdots > \varepsilon_n = \varepsilon$$

and apply them sequentially in time. With this approach, conservative models with respect to the prediction accuracy are initially derived. But as the threshold tightens and data become more plentiful, the model is allowed to follow the data more closely. Since the final value for the threshold is still ε, the theoretical results derived in previous section still hold.

The value of the threshold provides a global upper bound on the expected error of approximation. As stated earlier, guaranteeing any error bound is impossible unless some a priori assumption is imposed on the real function. Such an assumption could effectively be an upper and lower bound on the magnitude of the first derivative of the function. These bounds naturally arise when the unknown function describes a physical phenomenon, in which case, no steep gradients should be expected. If such bounds are available, the value of function between two neighboring data points is restricted between the intersection of two conic regions whose boundaries correspond to the upper and lower values of the first derivative. This conic region can effectively be considered as a guaranteed bound on the error, which, however, might or might not satisfy the desired accuracy as expressed by the value of the selected error threshold.

A tighter and local *estimate* on the generalization error bound can be derived by observing locally the maximum *encountered* empirical error. Consider a given dyadic multiresolution decomposition of the input space and, for simplicity, let us assume piecewise constant functions as approximators. In a given subregion of the input space, $I_{j,\mathbf{p}}$, let $Z_{l,\mathbf{p}}$ be the set of

data at instant l that are present into this region. The maximum encountered empirical error in this area is equal to

$$\eta_{l,\mathbf{p}} = \frac{\max(y_{l,\mathbf{p}}) - \min(y_{l,\mathbf{p}})}{2},$$

and also satisfies the relationship: $\eta_{l,\mathbf{p}} < \varepsilon$. This value constitutes a local error bound on the expected accuracy of the approximation unless otherwise proved by the incoming data. By observing the evolution of the local error bound, we can draw useful conclusions on the accuracy of the approximation in the different areas of the input space.

IV. Applications of the Learning Algorithm

In this section, the functionality and performance of the proposed algorithm are demonstrated through a set of examples. In all examples, the "Mexican hat" wavelet (Fig. 4) was used. An illustration of the effectiveness of the algorithm was given with the example of Fig. 1. The approximating curves were indeed derived by applying the algorithm using an error threshold equal to .5. The example was introduced in Zhang and Benveniste (1992) and its solution using *Wave-Nets* is more thoroughly explained in Bakshi *et al.* (1994).

A. Example 1

The first example is the estimation of the 2D function (from Narendra and Parthasarathy, 1990)

$$f(x_1, x_2) = \frac{x_1}{(1 + x_1^2)} + x_2^3. \tag{17}$$

A graph of the real function is presented in Fig. 7. A set of examples was created by random selections of the pairs (x_1, x_2) in the range $I = [0,1]^2$ and the error threshold, ε, was set equal to .1.

Figure 8 presents the evolution of the model during training. More specifically, the size of the network (number of basis functions) and the generalization error are plotted as a function of data presented to the learning algorithm. It can be observed that although the number of data is small, the generalization error is high and the network adaptation proceeds in high rates to compensate for the larger error. The generalization

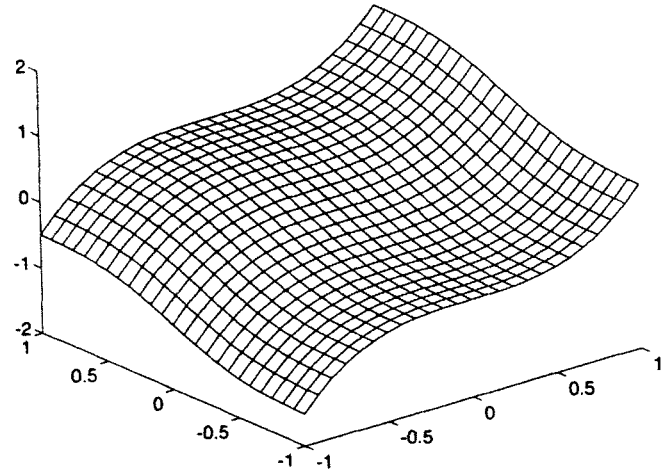

FIG. 7. Real function for Example 1.

error does not decrease monotonically, but it exhibits a clear descending trend. Large changes in the error coincide with data points that trigger structural adaptation (for example, after about 100 and 160 data points). That means that at those points the algorithm shifts to subspaces where better solutions are available and, most importantly, picks one of them resulting in drastic reductions of the error. After almost 160 points, no

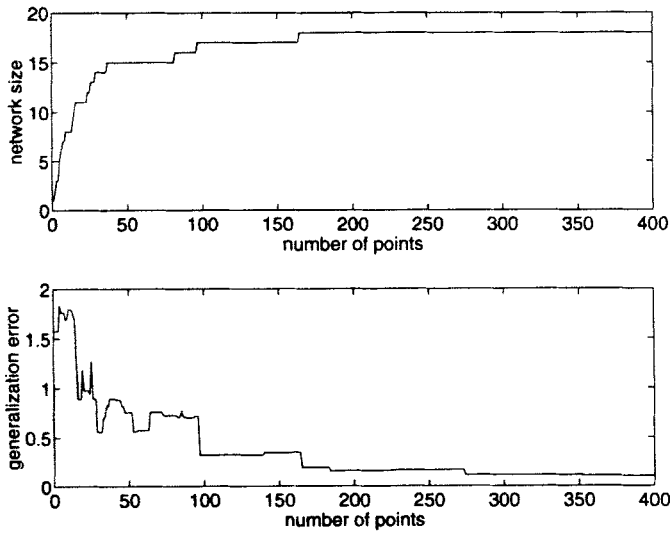

FIG. 8. Evolution of learning for Example 1.

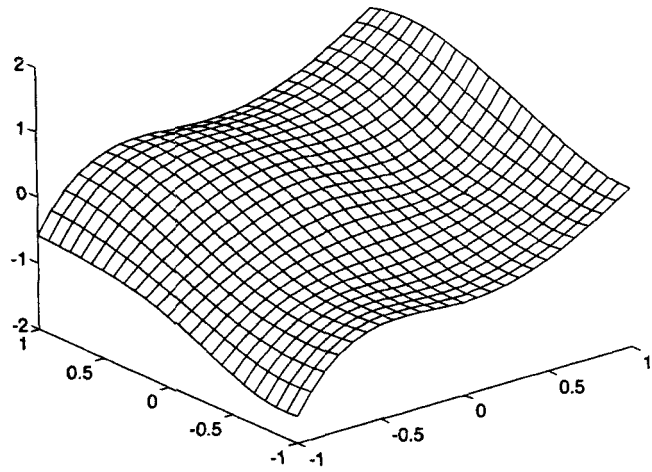

FIG. 9. Model surface for Example 1 after training with 400 points.

further structural adaptation is required and the network size reaches a steady state. The flat sections in the generalization error correspond to periods where no model adaptation is needed. After the 160th data point, coefficient adaptation is required only for three points until the 350th data when the generalization error drops for the first time below .1. At this point, learning has concluded, and a model with the specified accuracy has

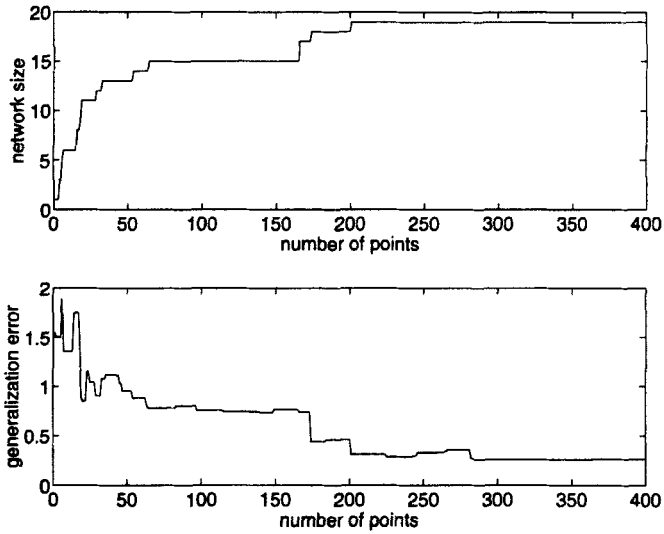

FIG. 10. Evolution of learning for Example 1: data corrupted with noise.

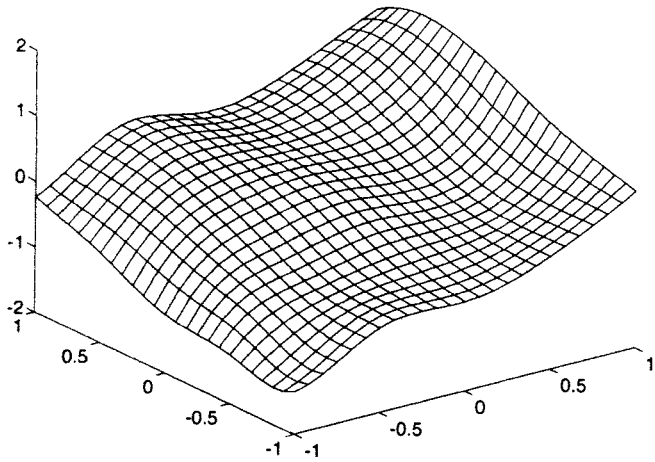

FIG. 11. Model surface for Example 1 after training with 400 noisy points.

been found. This model is shown in Fig. 9. It should be noticed, as is clear by comparing Fig. 7 and 9, that the error of .1 is the worst accuracy achieved. The approximation is much closer in the largest portion of the input space.

The same example was solved for the case when noise is present at the output, y. The maximum amplitude of the noise, δ, was .1 and the error threshold, ε, was set to .2 to reflect our desire to achieve a generalization error of $.1(=\varepsilon-\delta)$. The results (shown in Fig. 10) are similar with the no-noise case, except that the decrease of the generalization error and the derivation of the final model are slower. This is expected, since the presence of noise degrades the quality of information carried by the data as compared to equal amount of noise-free data. After 400 points the generalization error is .26 and, therefore, a model with the desired accuracy is yet to be achieved. The model (shown in Fig. 11), although not as accurate as before, provides a decent approximation of the real function. More importantly, the use of a larger threshold allows greater tolerance for the observed empirical error and, in this way, allows the model to avoid overfitting the noisy data.

B. EXAMPLE 2

Consider a continuous-stirred-tank reactor (CSTR) with cooling jacket where a first order exothermic reaction takes place. It is required to derive a model relating the extent of the reaction with the flowrate of the heat

transfer fluid. This is a nonlinear identification example very popular in chemical engineering literature. In this study we adopt the approach followed by Hernandez and Arkun (1992). The dimensionless differential equations that describe the material and energy balances in the reactor are

$$\frac{dx_1}{dt} = -x_1 + Da(1-x_1)\exp\left(\frac{x_2}{1+x_2/\gamma}\right),$$

$$\frac{dx_2}{dt} = -x_2 + BDa(1-x_1)\exp\left(\frac{x_2}{1+x_2/\gamma}\right) + b(u-x_1), \qquad (18)$$

$$y = x_1$$

where x_1 (or y) is the measured extent of the reaction, and x_2 the dimensionless temperature of the reactor. The input, u, is the dimensionless flowrate of the heat transfer fluid through the cooling jacket. The input (shown in Fig. 12) was constructed as a concatenation of step changes and random signals created by adding a pseudo random binary signal (PRBS) signal between -1 and 1 and a random variable with uniform distribution between $-.5$ and .5. The set of differential equations [Eq. (18)] was solved numerically with the initial conditions corresponding to

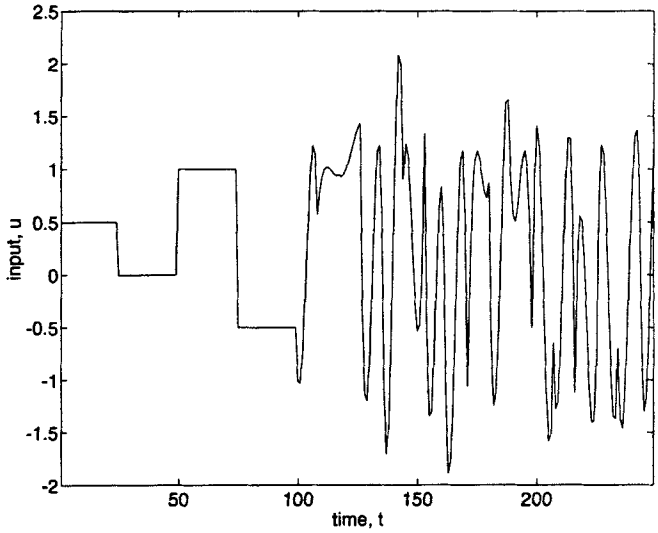

FIG. 12. Input data for identification example.

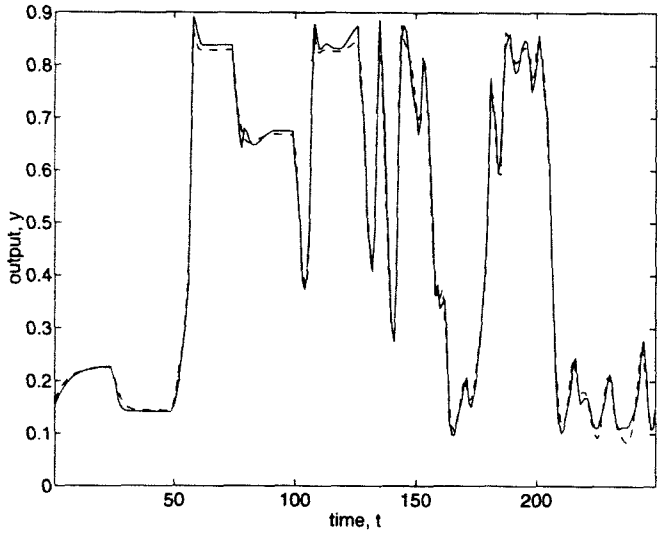

FIG. 13. Comparison of model predictions (solid line) with training data (dashed line) for identification example.

the equilibrium point $x_1 = .144$, $x_2 = .855$, and $u = .0$. The values of the output, created in this way, are represented by the dashed line in Fig. 13. The derived data were used to construct an input/output process model of the form

$$y(t+1) = f[(y(t), y(t-1), u(t))] \tag{19}$$

by using the *Wave-Net* algorithm to learn the unknown function $f(\cdot, \cdot, \cdot)$. The objective was to use the derived model to predict the behavior of the system in the future for different input values.

In Fig. 13, the model predictions are compared with the real output after training the network with 250 data points. The error threshold was set to .03 and the resulting model contained 37 basis functions with equal number of unknown coefficients. This model was used without further training to predict the system output for the next 750 time instants. The results are shown in Fig. 14, where a fairly good accuracy can be observed. The maximum error associated with the predictions is .11. It is greater than the defined threshold but it is much better than the maximum error ($= .28$) that would result by predicting the value of $y(t+1)$ equal to $y(t)$. This shows that the model has indeed captured the underlying dynamics of the physical phenomenon and is not merely predicting the future as a zero-order extrapolation of the present.

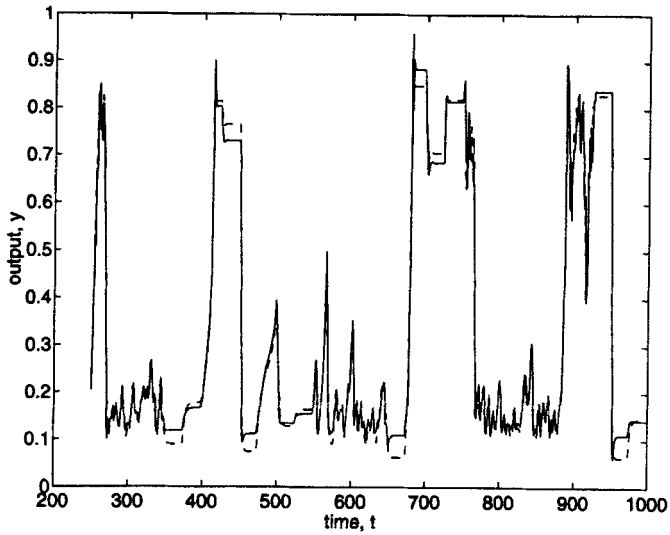

FIG. 14. Comparison of model predictions (solid line) with testing data (dashed line) for identification example.

C. EXAMPLE 3

The final example introduced in Zhang and Benveniste (1992) is a two-dimensional function with distinguished localized features. It is analytically represented by the formula

$$f(x_1, x_2) = (x_1^2 - x_2^2)\sin(5x_1) \tag{20}$$

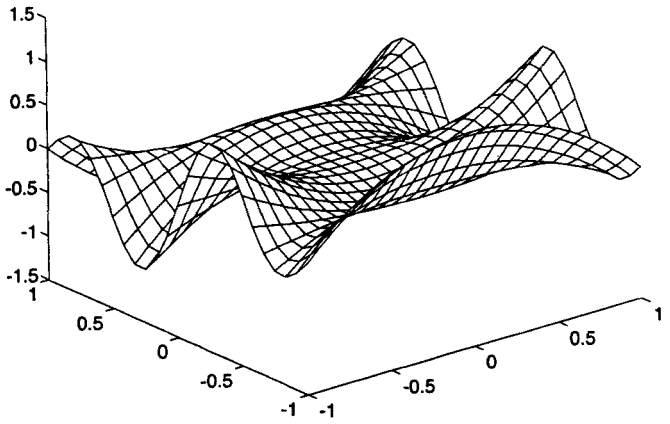

FIG. 15. Real function for example 3.

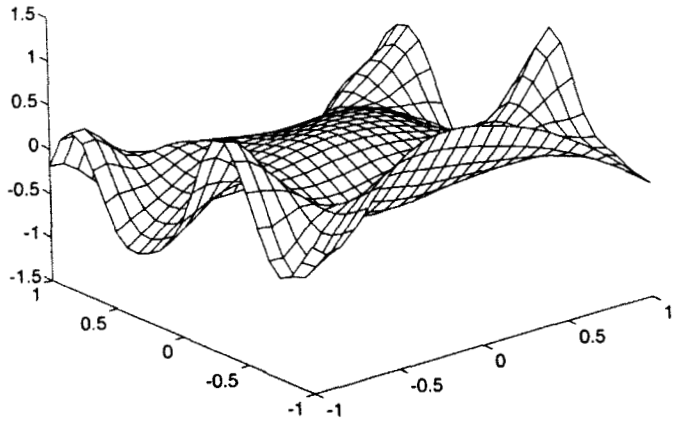

FIG. 16. Model surface for example 3 after training with 500 points.

and graphically represented in Fig. 15. After setting $\varepsilon = .2$ and using 500 data points for training, the resulting model exhibited a generalization error of .22. The model, shown in Fig. 16, has, within the predefined accuracy, captured the important features of the unknown function. This was possible by including during training more high-scale wavelets along the x_1 axis where the significant features lie. The coverage of the input space by the basis functions is shown in Fig. 17, where the circles

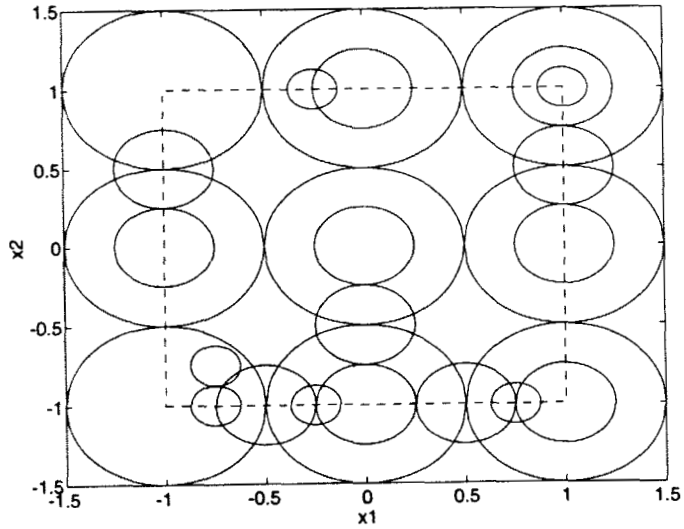

FIG. 17. Coverage of the input space (square) by the support (circles) of the basis functions for the model of Example 3.

correspond to the "dominant" support of each basis function and the square bounds the area where data were available. It is not surprising that most of the basis functions lie in the boundaries of the input space because this is where the biggest errors are most often encountered.

V. Conclusions

In this chapter the problem of estimating an unknown function from examples was studied. We emphasized the issues that stem from the practical point of view in posing the problem. In practice, one starts with a set of data and wants to construct an approximating function, for which, almost always, nothing is known a priori. The main point of the chapter is that the unknown function cannot be found but can be learned. Learning is an ever-lasting and ever-improving procedure. It requires the existence of a teacher that is the data and an evaluative measure. To ensure continuous improvement, the model has to be susceptible to new data and structural adaptation is an essential requirement to achieve this. NNs are neither data-centered nor flexible enough to enable one to learn from them since the important decisions for a given application are made in the absence of data. The derived approximating function has to be coarse and conservative where data are sparse and to follow closely the fluctuations of the data that might indicate the presence of a distinguished feature; localization and control of smoothness are important elements of the approximating scheme.

All the issues raised by these practical considerations have found rigorous answers within the *Wave-Net* framework. The chapter advocates the following points:

(a) The need for on-line model adaptation in parallel with the data acquisition
(b) The use of the local approximation error measured by the L^{∞} error to guide the model adaptation
(c) The use of the multiresolution framework to formalize the intuitive preference to simple and smooth approximating functions
(d) The use of bases, like wavelets, with well-defined properties as approximation schemes

The derivation of process models for adaptive control falls exactly within the framework of the estimation problem studied in this chapter. Control-related implementation are natural extensions to the current work and are

studied in another publication (Koulouris and Stephanopoulos, 1995). Some computational details and variations of the adaptation algorithm have also to be evaluated to improve the efficiency of the computations.

VI. Appendices

A. Appendix 1

1. Case 1: $f \in G$.

When $I(g)$ is given by Eq. (3), the solution to the minimization problem is the *regression function* (Vapnik, 1982):

$$g^* = \int yP(y|\mathbf{x})\, dy. \tag{21}$$

Since $y = f(\mathbf{x}) + \mathrm{d}$, $\mathrm{P(y|zx)} = \mathrm{P[y - f(x)]} = \mathbf{P}(d)$, and Eq. (21) yields

$$g^* = \int [f(\mathbf{x}) + d] P[y - f(\mathbf{x})]\, dy = \int f(\mathbf{x}) \mathbf{P}[y - f(\mathbf{x})]\, dy + \int d\mathbf{P}(d)\, dd$$

$$= f(x) + \bar{d}.$$

That is, the real function $f(\mathbf{x})$ is the solution to the minimization of Eq. (3) only in the absence of noise ($d = 0$) or when the noise has zero mean ($\bar{d} = 0$). This is, in fact, true for all L^n norms with $2 < n < \infty$ under the condition that n moments of the noise are zero. When the L^∞ metric given by Eq. (4) is used, we will prove that $f(\mathbf{x})$ yields the minimal value of $I(g)$, independently of the statistics of the noise and as long as the noise is symmetrically bounded.

$$\text{If } g = f, I(f) = \sup_x |y - f(\mathbf{x})| = \sup |d| = \delta,$$

$$\text{If } g \neq f, I(g) = \sup_x |y - g(\mathbf{x})| = \sup_x |f(\mathbf{x}) - g(\mathbf{x}) + d|$$

$$= \sup_x |f(\mathbf{x}) - g(\mathbf{x})| + \delta > \delta. \tag{22}$$

The reason for the last inequality is that for every $\mathbf{x}$ the maximum value of $|f(\mathbf{x}) - g(\mathbf{x}) + d|$ is $|f(\mathbf{x}) - g(\mathbf{x})| + \delta$ and the supremum value for all $\mathbf{x}$ is $\sup_x |f(\mathbf{x}) - g(\mathbf{x})| + \delta$.

2. *Case 2:* $f \notin G$.

It can be easily proved that when $f(\mathbf{x})$ is the regression function (Vapnik, 1982)

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

$$I(g) = I(f) + \mu_2(g, f),$$

$$\mu_2(g, f) = I(g) - I(f),$$

which implies L^2 closeness [$\mu_2(g, f)$ is small] if $I(g) - I(f)$ is small [$I(g) \geq I(f)$]. Similarly for the L^∞ case,

$$(22) \Rightarrow \quad \sup_x |f(\mathbf{x}) - g(\mathbf{x})| = I(g) - I(f) \qquad [\text{since } I(f) = \delta].$$

Again, the smaller the value of $I(g)[I(g) \geq I(f)]$, the closer $g(\mathbf{x})$ is to $f(\mathbf{x})$ in the L^∞ sense.

B. Appendix 2

The proof follows three intermediate steps.

(a) Let $I^{(l)}_{\text{emp}}(g) = \max_{i=1,\ldots,l} |y_i - g(\mathbf{x}_i)|$. Then

$$I^{(l)}_{\text{emp}}(g) \leq I^{(m)}_{\text{emp}}(g) \text{ for } m > l \text{ and } \forall g \in G, \tag{23}$$

which essentially states that every new point reveals for every function in G an empirical error that is equal or worse than the one already encountered. Equation (23) also conforms with the fact that $I_{\text{emp}}(g) < I(g)$ and is the basis of convergence of $I_{\text{emp}}(g)$ to $I(g)$.

(b) $(23) \Rightarrow \quad I(g) - I^{(l)}_{\text{emp}}(g) \geq I(g) - I^{(m)}_{\text{emp}}(g), \qquad m > l,$

$$\sup_g \left[I(g) - I^{(l)}_{\text{emp}}(g) \right] \geq \sup_g \left[I(g) - I^{(m)}_{\text{emp}}(g) \right],$$

$$\lim_{l \to \infty} \sup_g \left[I(g) - I^{(l)}_{\text{emp}}(g) \right] = 0.$$

This basically proves uniform convergence of $I^{(l)}_{\text{emp}}(g)$ to $I(g)$ and that, in turn, means

$$\forall k > 0, \exists N > 0 \quad \text{such that} \quad \sup_g \left[I(g) - I^{(l)}_{\text{emp}}(g) \right] < \kappa \quad \forall l > N. \tag{24}$$

(c) Let $\varepsilon > 0$ be a given positive number and $G_\varepsilon = \{g \in G | I(g) < \varepsilon\}$, which we assume to be nonempty. Let also $g^* = \text{argmin } I(g)$. Obviously, $g^* \epsilon G$ and $I(g^*) < \varepsilon$. If, in Eq. (24), we set $\kappa = \varepsilon - I(g^*)$, then

$$\exists N > 0 \quad \text{such that} \quad I(g) - I^{(m)}_{\text{emp}}(g) < \varepsilon - I(g^*) \quad \forall m > N$$

$$I^{(m)}_{\text{emp}}(g) > I(g) - \varepsilon + I(g^*) \tag{25}$$

We want to prove that, if this is the case, then only solutions with $I(g) < \varepsilon$ will be produced by the minimization of the empirical risk, and convergence in this weak sense will be guaranteed. Let g' be a function such that $I(g') > \varepsilon$. Then from Eq. (25)

$$I^{(m)}_{\text{emp}}(g) > I(g') - \varepsilon + I(g^*) > I(g^*),$$

but

$$I(g^*) > I^{(m)}_{\text{emp}}(g^*), \quad \text{so} \quad I^{(m)}_{\text{emp}}(g') > I^{(m)}_{\text{emp}}(g^*),$$

and, therefore, g' can never be the solution to the minimization of $I^{(m)}_{\text{emp}}(g)$ since there will always be at least be at least the best approximation g^*, with guaranteed lower value of the empirical risk.

C. Appendix 3

We first claim that if in some subspace $j < j^*$, $I_{\text{emp}}(g) > \varepsilon \; \forall g \in S_j$, there does not exist $g \in S_j$ so that $I(g) < \varepsilon$. That is straightforward since $I_{\text{emp}}(g) < I(g)$. By forcing a threshold on I_{emp}, we are guaranteed to find a data point for which no solution in S_j can satisfy the bound. The algorithm will then look for the solution at the immediate subspace S_{j+1} where, by Assumption 1, a better solution exists. Following this procedure, eventually subspace S^*_j is reached. Its existence is guaranteed by Assumption 2. The algorithm will never move to higher subspaces $j > j^*$ since $I_{\text{emp}}(g) < I(g) < \varepsilon \; \forall g \in G_\varepsilon$, and the requirement $I_{\text{emp}}(g) < \varepsilon$ will always be satisfied.

References

Bakshi, B., and Stephanopoulos, G., Wave-Net: A multiresolution, hierarchical neural network with localized learning. *AIChE J.* **39**, 57 (1993).

Bakshi, B., Koulouris, A., and Stepanopoulos, G., Learning at multiple resolutions: Wavelets as basis functions in artificial neural networks and inductive decision trees. *In* "Wavelet Applications in Chemical Engineering" (R. L. Motard and B. Joseph, eds.) Kluwer Academic Publishers, Dordrecht/Norwell, MA, p. 139 (1994).

Barron, A. R. Approximation and estimation bounds for artificial neural networks. *Mach. Learn.* **14**, 115 (1994).

Barron, A. R., and Barron, R. L., Statistical learning networks: A unifying view. *In* "Symposium on the Interface: Statistics and Computing Science." p. 192. Reston, VA, 1988.

Barto, A. G., Connectionist learning for control. *In* "Neural Networks for Control." (W. T. Miller, R. S. Sutton and P. J. Werbos, eds.) p. 5. MIT Press, Cambridge, MA, 1991.

Bhat, N. V., and McAvoy, T. J., Use of neural nets for dynamic modeling and control of chemical process systems. *Comput. Chem. Eng.* **14**, 573 (1990).

Bhat, N. V., and McAvoy, T. J., Determining the structure for neural models by network stripping. *Comput. Chem. Eng.* **16**, 271 (1992).

Cohen, A., Biorthogonal wavelets. *In* "Wavelets—A Tutorial in Theory and Applications," (C. K. Chui, ed.), Academic Press, San Diego, CA, p. 123. 1992.

Daubechies, I., "Ten Lectures on Wavelets." SIAM Philadelphia, 1992.

Feauveau, J. C., Nonorthogonal multiresolution analysis using wavelets. *In* "Wavelets—A Tutorial in Theory and Applications" (C. K. Chui, ed.), Academic Press, San Diego, CA, p. 153. 1992.

Girosi, F., "Rates of Convergence of Approximation by Translates and Dilates," AI Lab Memo, Massachusetts Institute of Technology, Cambridge, MA, 1993.

Girosi, F., and Anzellotti, G., Rates of convergence for radial basis functions and neural networks. "Artificial Neural Networks with Applications in Speech and Vision," (R. J. Mammone, ed.), p. 97. Chapman & Hall, London, 1993.

Hartman, E., Keeler, K., and Kowalski, J. K., Layered Neural Networks with Gaussian hidden units as universal approximators. *Neural Comput.* **2**, 210 (1990).

Hernandez, E., and Arkun, Y., A study of the control relevant properties of backpropagation neural net models of nonlinear dynamical systems. *Comput. Chem. Eng.* **16**, 227 (1992).

Hornik, K., Stinchcombe, M., and White, H., Multi-layer feedforward networks are universal approximators. *Neural Networks* **2**, 359 (1989).

Hoskins, J. C., and Himmelblau, D. M., Artificial neural network models of knowledge representation in chemical engineering. *Comput. Chem. Eng.* **12**, 881 (1988).

Kearns, M., and Vazirani, U., "An Introduction to Computational Learning Theory." MIT Press, Cambridge, MA. 1994.

Kon, M., and Raphael, L., "Convergence Rates of Wavelet Expansions," preprint 1993.

Koulouris, A., and Stepanopoulos, G., On-line empirical learning of process dynamics with Wave-Nets, submitted to *Comput. Chem. Eng.* (1995).

Kovačević, J., and Vetterli, M., Nonseparable multiresolutional perfect reconstruction banks and wavelet bases for R^n. *IEEE Trans. Inf. Theory*, **38**, 533 (1992).

Kramer, M. A., Thompson, M. L. and Bhagat, P. M., Embedding theoretical models in neural networks. *Proc. Am. Control Conf.* 475 (1992).

Kreinovich, V. Y., Arbitrary nonlinearity is sufficient to represent all functions by neural networks: A theorem. *Neural Networks* **4**, 381 (1991).

Lee, M., and Park, S., A new scheme combining neural feedforward control with model predictive control. *AIChE J.*, **38**, 193 (1992).

Leonard, J. A., and Kramer, M. A., Radial basis function networks for classifying process faults. *IEEE Control Syst.* **11,** pp. 31–38, (1991).

Ljung, L., "System Identification: Theory for the User." Prentice-Hall Englewood Cliffs, NJ. (1987).

Mallat, S. G., A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-11**, 674 (1989).

Mavrovouniotis, M. L., and Chang, S., Hierarchical Neural Networks. *Comput. Chem. Eng.* **16**, 347 (1992).

Moody, J., and Darken, C. J., Fast learning in networks of locally-tuned processing units. *Neural Comput.* **1**, 281 (1989).

Murty, K. G., "Linear Programming." Wiley, New York, 1983.

Narendra, K. S., and Parthasarathy, K., Identification and control of dynamical systems using neural networks. *IEEE Trans. Neural Networks* **1**, (1990).

Pati, Y. C., Wavelets and time-frequency methods in linear systems and neural networks. Ph.D. Thesis, University of Maryland, College Park (1992).

Platt, J., A resource-allocating network for function interpolation. *Neural Comput.* **3**, 213 (1991)

Poggio, T., and Girosi, F., "A Theory of Networks for Approximation and Learning," AI Lab. Memo. No. 1140. Massachusetts Institute of Technology, Cambridge, MA, 1989.

Psichogios, D. C., and Ungar, L. H., Direct and indirect model based control using artificial neural networks, *Ind. Eng. Chem. Res.* **30**, 2564 (1991).

Psichogios, D. C., and Ungar, L. H., A hybrid neural network-first principles approach to process modeling. *AIChE J.* **38**, 1499 (1992).

Rengaswamy, R., and Venkatasubramaniam, V., Extraction of qualitative trends from noisy process data using neural networks. *AIChE Annu. Meet. Los Angeles* (1991).

Rumelhart, D. E., McClelland, J. L., and the PDP Research Group, "Parallel Distributing Processing." MIT Press, Cambridge, MA, 1986.

Strang, G., Wavelets and dilation equations: A brief introduction. *SIAM Rev.* **31**, 614 (1989).

Ungar, L. H., Powell, B. A., and Kamens, S. N., Adaptive Networks for fault diagnosis and process control. *Comput. Chem. Eng.* **14**, 561 (1990).

Vapnik, V., "Estimation of Dependences Based on Empirical Data." Springer-Verlag, Berlin, 1982.

Ydstie, B. E., Forecasting and control using adaptive connectionist networks. *Comput. Chem. Eng.* **14**, 583 (1990).

Zhang, Q., and Benveniste, A., Wavelet networks. *IEEE Trans. Neural Networks* **3**, 889 (1992).