Wave-Net: a Multiresolution, Hierarchical Neural Network with Localized Learning

Bhavik R. Bakshi and George Stephanopoulos
Laboratory for Intelligent Systems in Process Engineering, Dept. of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139

A Wave-Net is an artificial neural network with one hidden layer of nodes, whose basis functions are drawn from a family of orthonormal wavelets. The good localization characteristics of the basis functions, both in the input and frequency domains, allow hierarchical, multiresolution learning of input-output maps from experimental data. Furthermore, Wave-Nets allow explicit estimation for global and local prediction error-bounds, and thus lend themselves to a rigorous and explicit design of the network. This article presents the mathematical framework for the development of Wave-Nets and discusses the various aspects of their practical implementation. Computational complexity arguments prove that the training and adaptation efficiency of Wave-Nets is at least an order of magnitude better than other networks. In addition, it presents two examples on the application of Wave-Nets; (a) the prediction of a chaotic time-series, representing population dynamics, and (b) the classification of experimental data for process fault diagnosis.

Introduction

Artificial neural networks have been widely used to solve learning problems in a variety of fields. Using available data, the neural networks "learn" relationships between given inputs and outputs. These nonlinear relationships are completely empirical and are not based on any fundamental physical theories. As such, neural networks are simply, "... complex, nonlinear regression models whose structure is determined empirically" (Leonard et al., 1991). Although the initial inspiration came from the networks of nerve cells in the brain, current developments in applied neural networks (or connectionist networks) are primarily driven by advances in functional analysis, as will become clearer in subsequent sections.

In the area of process systems engineering we have witnessed an explosion in academic and industrial interest in neural networks, whose applicability in process design, product design, and especially process operations and control has been explored in a variety of ways. For example, neural networks have been employed to

- generate nonlinear models for the design of fixed or adaptive model-predictive control systems (Ungar et al., 1990; Ydstie, 1990; Bhat et al., 1990; Hernandez and Arkun, 1990; Haesloop and Holt, 1990);
- diagnose the occurrence of process faults and identify the root causes (Hoskins and Himmelblau, 1988; Watanabe et al., 1989; Kramer and Leonard, 1990; Venkatasubramaniam et al., 1990; Leonard and Kramer, 1991);
- monitor and interpret process trends, leading to an evaluation of the performance and/or product quality of continuous or batch operations (Rengaswamy and Venkatasubramaniam, 1991), or
- model chaotic behavior of deterministic dynamic systems (Levin, 1990; Adomaitis et al., 1990).

The richness of representations they can capture (Boolean, qualitative, semiquantitative, and/or analytic relationships), the high degree of parallelism in computations they afford, and the simplicity of their structure have made neural networks an extremely popular tool for solving many different types of engineering problems.

A neural network is typically composed of multiple layers of interconnected nodes with an activation function in each node and weights on the edges connecting the nodes of the network. The output of each node is a nonlinear function of all its inputs. Thus, the network represents an expansion of the unknown nonlinear relationship between inputs, x and outputs, F, into a space spanned by the functions represented by the activation functions of the network's nodes. Specifically, Poggio and Girosi (1989) have shown that learning by feedforward neural networks can be regarded as synthesizing

an approximation of a multidimensional function, over a space spanned by the activation functions, $\phi_i(x)$, $i=1, 2, \ldots, m$, that is,

$$F(x) = \sum_{i=1}^{m} c_i \phi_i(x) \tag{1}$$

Using empirical data, the activation function parameters, and the network parameters, c_i, $i=1, 2, \ldots, m$, are adjusted in such a way as to minimize the approximation error.

Two types of activation functions are commonly used: global and local. Global activation functions are active over a large range of input values, and provide a global approximation to the empirical data. Local activation functions are active only in the immediate vicinity of the given input value. Their effect drops off for input values away from the center of the activation function's receptive field. The two most commonly used global activation functions are

(a) the linear threshold unit used in perceptrons, and

(b) the sigmoid function, used in BackPropagation Networks (BPN).

From Figures 1a and b we see that these two functions are active for all input values greater than a given parameter. The functions which can be computed by a BPN with one hidden layer having m nodes constitute a very large set, S_m, defined by (Girosi and Poggio, 1989),

$$S_m \equiv \left\{ f(x): f(x) = \sum_{i=1}^{m} c_i \sigma(x w_i + \theta_i),\ w_i \in R^d,\ c_i,\ \theta_i \in \mathrm{R} \right\} \tag{2}$$

where, $\sigma(x)$ is the sigmoid function of Figure 1b, w_i, c_i and θ_i are adjustable parameters. As a matter of fact, Cybenko (1989) has shown that for large enough m, the set S_m includes any continuous function.

The activation function in Radial Basis Function Networks (RBFN) is local in character, as shown in Figure 1c. In general, a Radial Basis Function for the i-th node is given by

$$\phi_i(x) = h(\|x - x_i\|) \tag{3}$$

with Gaussian as the most common form of the function, h, that is

$$\phi_i(x) = \exp\left(-\frac{\|x - x_i\|^2)}{2\sigma_i^2} \right) \quad \text{if } x \in R \tag{3a}$$

$$\phi_i(x) = \frac{|W_i|}{\pi^{d/2}} \exp\left(-\frac{1}{2} (x - x_i)^T W_i^2 (x - x_i) \right) \quad \text{if } x \in R^n \tag{3b}$$

where, σ_i is the standard deviation for the one-dimensional case and W_i the $d \times d$ weight matrix formed by reciprocals of the covariance for the d-dimensional case. RBFNs are also capable of approximating any continuous function with arbitrary accuracy, given enough nodes (Girosi and Poggio, 1989; Poggio and Girosi, 1990; Stinchcombe and White, 1989; Hartman et al., 1990; Kreinovich, 1991). But, the approximation properties of BPNs and RBFNs are quite different.

Due to the global nature of the activation function, each node in a BPN influences the output over a large range of

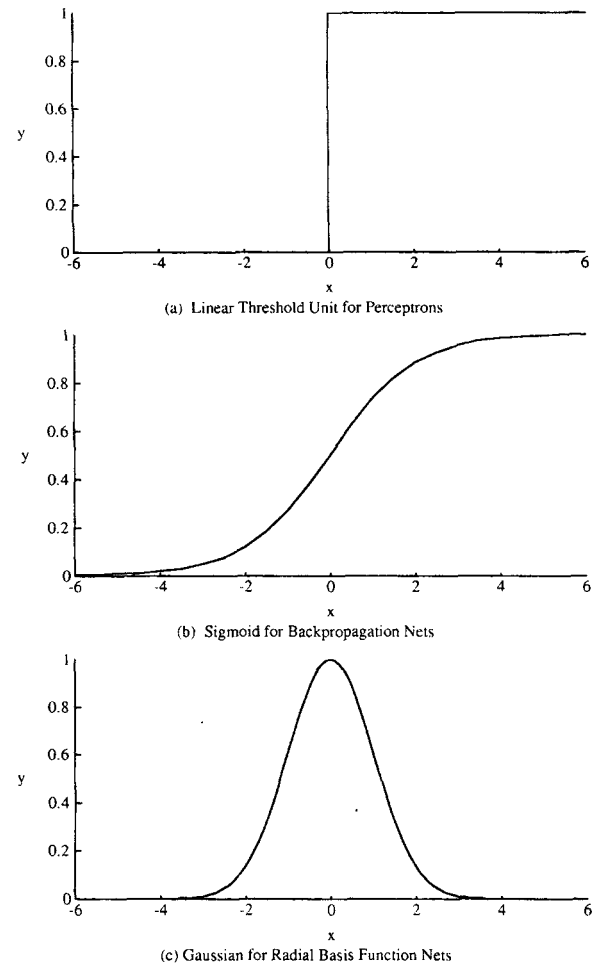

Figure 1. Activation functions for various artificial neural networks.

input values. Furthermore, all the activation functions overlap over a large range of input values, and therefore interact with each other. Correction in the network output for a given input requires modification of parameters over all the interacting nodes. This makes adaptation and incremental learning with global approximators a slow process. Convergence of BPNs is also not guaranteed due to the nonlinear nature of the optimization problem, and they may get stuck in local minima on the error surface. In addition, global approximation networks provide a value for the output over the whole range of input values, independently of the availability or density of training data in given ranges of input values. Such a property could lead to large extrapolation errors without warning.

Neural networks with local activation functions overcome the above disadvantages of global approximation networks. For each input value, only a few nodes have nonzero activations. These nets are fast to train and adapt easily to new data since they require changes in only a small part of the net. The local nature of the activations allows extrapolation over a small range of input values for which the activations are nonzero. This property allows RBFNs to avoid large extrapolation errors and provides a measure of the reliability of the network output based on the density of the training data (Leonard et al., 1991). Also, convergence problems in RBFNs are less serious than

those in BPNs. Algorithms for fast learning using RBFNs were developed by Moody and Darken (Moody and Darken, 1988, 1989; Moody, 1989), and subsequent work has demonstrated the expected benefits of RBFNs for localized learning (Stokbro et al., 1990; Leonard and Kramer, 1991; Lee and Kil, 1991).

Equations 1 and 2 indicate that the neural network training procedure determines the parameters of the basis functions and the coefficients indicating their relative contribution to the overall input-output map. But, the set of the basis functions, which also determines the structure of the network, has to be determined empirically by trial and error. Furthermore, no physical interpretation can be easily attributed to the model generated by the trained network. This lack of formal procedures for the design and interpretation of neural networks gives them a "black box" character. Poggio and Girosi's (1989) theory of networks for approximation and learning reveals the relationship between the model learned by neural nets and functional approximation theory. They have shown that RBFNs provide a solution to the problem of approximating data by regularization, which is a commonly used method for functional approximation. Improved understanding of the relationship between neural networks, approximation theory and functional analysis has prompted several researchers to look for better ways to design neural networks.

From the theory of functional analysis it is well known that functions can be represented as a weighted sum of orthogonal basis functions. Such expansions can be easily represented as neural nets by having the selected basis functions as activation functions in each node, and the coefficients of the expansion as the weights on each output edge. Networks can be designed for the desired error rate using the properties of orthonormal expansions, thus decreasing the black box character of neural nets. Several orthogonal basis functions are known, for example, sinusoids, Legendre polynomials, Walsh functions, and so on. Lee and Kil (1991) have shown that sinusoidal activation functions provide better mapping capability than sigmoids or Gaussians. Unfortunately, most orthogonal functions are global approximators, and like sigmoids, suffer from the disadvantages of approximation using global functions. Holcomb and Morari (1991) give a method for designing an RBFN based on the principles of functional analysis. They use the properties of orthonormal expansions to determine the number of nodes required in the network by forcing orthogonality on Gaussians by prohibiting their overlap. Artificial imposition of orthogonality on the activation functions requires forced generalization of the learned model through an ad hoc "penalty function". These features result in ad hoc training and adaptation procedures with no clear termination criterion to decide how many nodes to train. This method, while an improvement over previous training methods, still retains a large degree of "black box" character in the neural net. Such a network does not suffer from the disadvantages of global approximation functions but cannot take full advantage of orthogonality either. What we need is a set of basis functions which are local and orthogonal.

It was believed until recently that it was not possible to build simple orthonormal bases with good localization properties. Meyer (1985) was the first to construct such a basis for all square-integrable functions, using a special class of functions which came to be known as wavelets. Subsequently, several other orthogonal wavelets with good localization properties have been developed (Daubechies, 1988; Mallat, 1989). In this article we propose the development of neural networks with activation functions derived from various classes of orthogonal wavelets. The resulting *Wave*let *Net*work, or *Wave-Net* has all the advantages of true localized learning.

Furthermore, in most learning problems the training data are often nonuniformly distributed in the input space, that is, data may be sparse in some ranges of input values and dense in others. An efficient way of solving such problems is by learning at multiple resolutions. A higher resolution of the input space may be used if data are dense and a lower resolution where they are sparse. Wavelets, as it will be explained in subsequent sections, in addition to forming an orthogonal basis are also capable of explicitly representing the behavior of a function at various resolutions of input variables. Consequently, a Wave-Net is first trained to learn the mapping between inputs and outputs at the coarsest resolution of input values. In subsequent stages, the network is trained to incorporate elements of the input-output mapping at higher and higher resolutions of the input variables until the desired level of trade-off between accuracy and generalization has been reached. Such hierarchical, multiresolution training has many attractive features for solving engineering problems, for example, a meaningful interpretation of the resulting mapping, estimation of mapping errors both in local ranges of input values and at various resolutions of input values, and finally, order(s) of magnitude more efficient for training and adaptation of the network, than those offered by earlier methods. In subsequent sections we will expand and substantiate all of these claims.

In this article we develop the fundamentals of Wave-Nets and describe their computer-aided implementation, observing the following structure: the second section discusses approximation theory as the essential theoretical basis for the design of neural networks. The material of this section is of central value in defining the framework for the construction of Wave-Nets. The third section introduces the concept of multiresolution, hierarchical learning, which constitutes the second leg on which the Wave-Net stands. The fourth section introduces wavelets and discusses their two principal properties; the fact that they form an orthonormal basis of all square-integrable functions, and their good local character. The resulting multiresolution characterization of input-output mappings is also discussed in this section. The fifth section presents how the various practical aspects in the design of Wave-Nets have been addressed and gives measures of the computational complexity of the resulting algorithms. Finally, in the sixth section we present some numerical examples, while the final section summarizes our conclusions on the design and use of Wave-Nets.

Networks and Approximation Theory

The task of learning the mapping between inputs and outputs from sets of input-output values is equivalent to approximating the underlying function representing the hypersurface in the input-output space. Mathematicians have studied the approximation of a real continuous function by an approximating function dependent on a fixed number of parameters under the name of Approximation Theory. Formally, the approximation problem may be stated as follows (Rice, 1964):

Approximation Problem Let $f(x)$ be a real-valued continuous function defined on a set X, and let $F(A, x)$ be a real-valued approximating function depending continuously on $x \in X$ and on n parameters, A. Given the distance function ρ, determine the parameters $A^* \in \mathfrak{A}$ such that

$$\rho[F(A^*, x), f(x)] \leq \rho[F(A, x), f(x)] \tag{4}$$

for all $A \in \mathfrak{A}$.

$\mathfrak{A}$ is the space in which the parameters lie and is usually the ordinary Euclidean space. The distance function ρ is a "measure of the approximation" and is generally given as the L^p norm of the difference $F(A^*, x) - f(x)$, that is

$$\rho = L^p[F(A^*, x) - f(x)]$$

$$= \left[\int_0^1 |F(A^*, x) - f(x)|^p dx \right]^{1/p}, \quad p \geq 1 \tag{5}$$

The solution to the approximation problem Eq. 4 is said to be a best approximation of the underlying function.

Poggio and Girosi have developed a theory of networks based on approximation theory (Poggio and Girosi, 1989; Girosi and Poggio, 1989; Poggio and Girosi, 1990). They have analyzed various networks for their approximation abilities and have shown that backpropagation networks with sigmoid activation functions are not best approximations, whereas networks with radial basis functions (RBFNs) are (Girosi and Poggio, 1989). Their work is described briefly in this section, since we use a similar framework to analyze best approximation properties of Wave-Nets.

Various networks may be represented as approximation schemes. For example, a linear approximation is given by

$$F(W, X) = WX \tag{6}$$

with W an $m \times n$ matrix of coefficients and X an $n \times 1$ vector of input variables. This corresponds to a network with n inputs, m outputs and no hidden units. Many well-known approximation schemes like spline fitting, expansion on orthogonal basis, single layer BPNs and so on can be represented by

$$F(W, X) = W\Phi(X) \tag{7}$$

that is, as a linear combination of a suitable set of basis functions $\Phi = \{\Phi_i(X), i = 1, 2, \ldots, m\}$. This corresponds to a network with one hidden layer. Backpropagation networks with multiple layers may be expressed as

$$F(W, X) = \sigma\left(\sum_n w_n \sigma\left(\sum_i v_i \sigma\left(\ldots \sigma\left(\sum_j u_j X_j\right)\ldots\right)\right)\right) \tag{8}$$

where σ is the sigmoidal function and w_n, v_i, u_j, ... are the adjustable coefficients.

The representation of networks as approximation schemes is useful for analyzing the theoretical properties of various networks. Radial Basis Functions Networks can be derived from the use of regularization theory for approximation.

Approximation by regularization theory and radial basis functions

Consider the classical interpolation problem defined as follows:

Interpolation problem. Given N different input vectors, $x_i, x_i \in R^n$, $i = 1, 2, \ldots, N$, and the corresponding N real outputs, $y_i, y_i \in R$, $i = 1, 2, \ldots, N$, find a function F, $F: R^n \rightarrow R$, satisfying the interpolation conditions

$$F(x_i) = y_i \quad i = 1, 2, \ldots, N \tag{9}$$

The solution to the above problem is a function which is a linear combination of N radial basis functions, that is

$$F(x) = \sum_{i=1}^{N} c_i h(\|x - x_i\|) \tag{10}$$

Gaussians, multiquadratics, and many others can be used as the specific radial basis functions to solve the interpolation problem defined above.

The interpolation problem is equivalent to approximation with zero error. Therefore, radial basis functions (RBF) can be used to provide a solution to the approximation problem (Broomhead and Lowe, 1988), by considering basis functions for K input points, t_k, $k = 1, 2, \ldots, K$, with $K < N$, that is

$$F(x) = \sum_{k=1}^{K} c_k h(\|x - t_k\|) \tag{11}$$

Using regularization theory, Poggio and Girosi (1989) solved the approximation problem by finding the function $F(x)$ which minimizes the following functional

$$H[F(x)] = \sum_{i=1}^{N} (y_i - F(x_i))^2 + \lambda \|PF(x)\|^2 \tag{12}$$

where, y_i are the measured values of the unknown function, $F(x)$ is the approximation of the unknown function, and λ is a positive real number called the regularization parameter. P is a constraint operator, called stabilizer, whose structure embodies the a priori knowledge about the solution, and therefore it depends on the characteristics of the particular problem to be solved. Normally, P is selected to be a differential operator so that it reflects the desired degree of smoothness of the unknown function. Therefore, $H[F(x)]$ expresses the compromise between the error of approximation and degree of smoothness of the unknown function. In other words, it expresses the trade-off between interpolation and generalization. The approximated solution to the problem given by Eq. 12 is a linear combination of what Poggio and Girosi (1988) called the Generalized Radial Basis Functions (GRBF), that is

$$F(x) = \sum_{k=1}^{K} c_k G(x; t_k) \tag{13}$$

where, $G(x; t_k)$ is the Green's function of the differential operator $\hat{P}P$, with $\hat{P}$ being the adjoint operator of P. If P is

an operator with radial symmetry, the Green's function G is radial and therefore the approximating function becomes

$$F(x) = \sum_{k=1}^{K} c_k G(\|x - t_k\|^2) \tag{14}$$

The coefficients, c_k, and the centers, t_k, are unknown and their values are found through numerical optimization.

All the radial basis functions considered above are at the same resolution. Thus, if the RBFs are Gaussian, they possess the same standard deviation, σ. Using GRBFs with different resolutions, we are led to the concept of Hyper Basis Functions (HyperBF) (Poggio and Girosi, 1990), which provide the solution to the following modified form of the regularization problem,

$$\underset{f_m}{\text{Minimize}}\ H[F(x)] = \sum_{k=1}^{N} \left[\sum_{m=1}^{L} f_m(x_k) - y_k \right]^2 + \sum_{m=1}^{L} \lambda_m \|P_m f_m\|^2 \tag{15}$$

where, the unknown function $F(x)$ has been regarded as the sum of L components, $f_m(x)$, $m = 1, 2, \ldots, L$, that is

$$F(x) = \sum_{m=1}^{L} f_m(x)$$

Thus, the approximated solution to Eq. 15 is given by,

$$F(x) = \sum_{m=1}^{L} \sum_{k=1}^{K} c_{mk} G_m(x; t_k) \tag{16}$$

where the radial basis functions G_m, $m = 1, 2, \ldots, L$, define the behavior of the unknown function, $F(x)$, at various resolutions of the input variables. The parameters c_{mk} and t_k are determined through global numerical optimization.

One of the most important results from the theory of networks for approximation of continuous functions is a theorem showing that a backpropagation network with sigmoid basis functions is not a best approximation, that is, the approximation of functions by BPNs does not minimize any norm. On the other hand, networks based on regularization theory minimize the L^∞, or the Tchebycheff norm when the number of basis functions is equal to the number of training data (Girosi and Poggio, 1989). This property does not hold when the number of basis functions is not equal to the number of training data and the centers, t_k of the expansion are unknown. Nevertheless, training procedures have been devised for regularization networks that preserve the property of best approximation, as will be discussed in the next paragraph.

Training procedure for RBFNs

Most commonly used RBFNs are not designed hierarchically. Therefore, designing an RBFN involves determination of the parameters c_k, t_k, and σ_k which minimize the global approximation error. This problem may be solved as a single global optimization problem by supervised learning, and was tried by Moody and Darken (1989). They also tried to determine the parameters by breaking the problem into two phases. The first phase determined the centers, t_k and standard deviations, σ_k in an unsupervised manner, while the second phase performed the optimization via supervised training to determine the c_k. They found the two-phase approach, described below, to be more efficient.

Phase 1. Self-organized learning. During this phase the centers of the K radial basis functions, t_k, and the extents, σ_k, of all basis functions are computed. The standard k-means clustering algorithm is used to find K receptive field centers in the input training examples. Each cluster gives a hidden node in the network. The center of the cluster determines the value of t_k for the basis function. This step allocates nodes only for regions where input data are present. The width (or variance) of each field is then determined by a contiguity heuristic. Various p-nearest neighbor heuristics may be used. For example, the width may be given by the geometric mean, $\sigma = \sqrt{d_1 d_2}$ where d_1 and d_2 are Euclidean distances from the k-th center to the two nearest centers. These heuristics achieve a certain amount of overlap between each unit and its neighbors to allow a smooth interpolation over the input space. This self-organized learning reduces the amount of work to be done by the supervised learning, since only the output weights have to be decided by error propagation.

Phase 2. Minimization of mean-squares error. The weights, c_k, on the radial basis functions are found from the minimization of the mean-squares error,

$$E = \sum_{k} [y_k - F(x_k)]^2$$

Determination of the weights is a linear problem, and convergence is guaranteed. Various improvements and alternate training methods have been suggested for RBFNs. Stokbro et al. (1990) have used coefficients that are linear functions of the input, allowing skewed Gaussians, and a quadratic cost function of the form,

$$E = \left[\frac{1}{2} \sum_i y_i^2 \right] - \sum_k c_k \left[\sum_i y_i \Phi_k(x_i) \right] + \frac{1}{2} \sum_k \sum_l c_k c_l \left[\sum_i \Phi_k(x_i) \Phi_l(x_i) \right]$$

where (x_i, y_i) are the inputs and the corresponding outputs, and $\Phi_k(x_i)$, $\Phi_l(x_i)$ denote the basis functions. This method is shown to be superior to Moody and Darken's method for predicting chaotic time series.

The heuristic Phase 1 of training RBFNs requires trial and error with different number of hidden units to design the optimal network. Holcomb and Morari (1991) give a method for estimating the contribution of each hidden unit to the approximation. This provides an insight into the number of hidden units to use. They give a local training method that trains each hidden unit individually. A multivariable Gaussian function is used that allows elliptical receptive fields. This is an improvement over spherical receptive fields because it provides an extra degree of freedom for assigning the basis functions appropriately on the training data. Their local training method is developed by drawing analogy with expansion of functions on an orthonormal basis. Since RBFs do not form an ortho-

normal basis, Holcomb and Morari (1991) try to maintain independence (orthogonality) between the basis functions by minimizing the overlap between the receptive fields. This condition leads to interpolation of the training data by the assignment of delta functions as receptive fields on each data point. Generalization is forced by introducing a penalty function to force the receptive fields to spread out. The value of the penalty function is determined empirically. Empirical determination of the number of nodes in the globally trained network is replaced by empirical determination of the penalty function value in the locally trained network. Nevertheless, this method gives a clearer measure of the contribution of each hidden unit which provides insight into when one has enough units. As pointed out by the authors, the assumption of nonoverlapping receptive fields may be satisfactory for classification problems, but is likely to break down if the approximated surface has to be smooth. The reason for this possible breakdown is that the nonoverlapping basis functions do not span the input space completely. This method gives us an indication of the potential benefits of using activation functions that are naturally orthonormal and have local receptive fields, in removing the arbitrariness in neural network design.

Approximation by expansion on an orthonormal basis

The previous paragraph indicated that if the basis functions were orthogonal to each other, while maintaining good localization of the corresponding receptive fields, the training of a neural network could be completely localized, while the number of hidden nodes would be directly determined by the added accuracy offered by a new node. Before we examine how wavelets can play the role of such "ideal" basis function for a network, let us briefly review the properties of approximation on an orthonormal basis.

Consider a function $F(x)$ which is assumed to be continuous in the range [0, 1]. Let $\phi_i(x)$, $i = 1, 2, \ldots, \infty$ be an orthonormal set of continuous functions in [0, 1]. Then, $F(x)$ possesses a unique L^2 approximation (Rice, 1964) of the form,

$$F(C, x) = \sum_{k=1}^{n} c_k \phi_k(x) \tag{17}$$

where the elements of the vector of coefficients $C = [c_1, c_2, \ldots, c_n]^T$ are given by the projection of $F(x)$ onto each basis function, that is

$$c_k = \int_0^1 F(x)\phi_k(x)dx \tag{18}$$

As we include more basis functions in the approximation in Eq. 17, the mean-squares error decreases and, in the limit, the error tends to zero, that is, the set $\{\phi_k(x)\}$ is complete. The error of approximation from considering only K terms in the expansion in Eq. 17 is given by

$$e_K^2 = \int_0^1 \left[F(x) - \sum_{k=1}^{K} c_k \phi_k(x) \right]^2 dx = \sum_{k=K+1}^{\infty} c_k^2 \tag{19}$$

The larger the value of the coefficient, c_k, the greater the contribution of the corresponding basis function, $\phi_k(x)$, in the approximating function. This observation provides a formal criterion for picking the most important activation function in each hidden unit of a network. The following theorem provides the criterion for generating the smallest network, that is, the smallest set of orthonormal basis functions, with a desired degree of approximation.

Theorem. Given an orthonormal set of functions, $\phi_k(x)$, $k = 1, 2, \ldots, \infty$, the smallest network for approximating a function $F(x)$ that is continuous in [0, 1], with a desired degree of accuracy is achieved by selecting basis functions corresponding to the largest coefficients c_k, calculated by Eq. 18. The resulting error rate is defined as the mean-squares, or L^2 error given by Eq. 19.

Proof. The smallest network is obtained by leaving out the maximum number of terms from the approximation. We prove that the number of terms left out, will be maximized by picking the smallest c_k for $k \in [K+1, \infty]$. K is the number of terms required to approximate the function $F(x)$ with an L^2 error of e_K. Since, e_K is given by Eq. 19, it is clear that smaller c_k's contribute less to e_K. Therefore, picking smaller c_k's will maximize the number of terms between $K+1$ and ∞, thus minimizing the number of terms from 1 to K.

The theoretical properties of expansion on orthonormal basis functions makes them very appealing for approximating relationships among data. Many techniques, like Fourier analysis, are based on these properties but suffer the disadvantages of global approximators, and the lack of input-frequency localization. The well-known Gibbs phenomenon in Fourier analysis is an example of the effect of global approximation using sinusoids. The availability of local orthonormal functions allows us to take advantage of the theoretical properties of orthonormal expansions, as well as bypass the problems of global approximators.

Networks With Multiresolution Hierarchies

In most learning problems training data are often nonuniformly distributed in the input space. Data may be sparse in some regions and dense in others. Approximating such data at a single coarse resolution may not bring out the fine details. A single fine resolution brings out the details, but no general picture may emerge. This tradeoff between the ability to capture fine detail and good generalization may be solved by learning at multiple resolutions. A higher resolution of the input space may be used if data are dense and lower resolution where they are sparse. Moody (1989) designed a multiresolution, hierarchical RBFN for localized learning. He uses ideas from the Cerebellar Model Articulation Controller (CMAC) (Albus, 1975) to allocate basis functions at multiple resolutions only to those regions of the input space where data are available. Such ideas have been used in several fields like solving differential equations by multigrid methods, image processing, and wavelet analysis as will be described in the fourth section.

Consider the function $F(x)$ to be expressed by its multiresolution components at L scales, that is

$$F_L(x) = \sum_{m=1}^{L} f_m(x) \tag{20}$$

where, the component at the m-th scale, $f_m(x)$, is given by

$$f_m(x) = \sum_{k=1}^{K_m} c_{mk}\phi_{mk}(x) \tag{21}$$

The basis functions ϕ_{mk} in Eq. 21 are all defined at scale m. Assume also that $m=\phi$ defines the lowest scale (finest resolution of input data) and $m=L$ the highest. A neural network is trained to learn the mapping between inputs and output at the coarsest resolution first. Then, the network is trained to learn the added detail as one moves from a coarser to a finer level of resolution. Clearly, the output of a network at the resolution level $m-1$ is given by,

$$F_{m-1}(x) = F_m(x) + f_{m-1}(x) \tag{22}$$

where $f_{m-1}(x)$ represents the difference in the approximation of function $F(x)$ at two adjacent scales. We will refer to $f_{m-1}(x)$ as the detail of the approximation at scale $m-1$. The error in the approximation at each resolution is given by

$$e_m^2 = \int_0^1 \left[f_m(x) - \sum_{k=1}^{K} c_{mk}\phi_{mk}(x) \right]^2 dx \tag{23}$$

which cannot be simplified as in Eq. 19 unless ϕ_{mk} are orthonormal. The training method guarantees that the mapping is the best possible, if the c_{mk} resulted from the minimization of the mean-squares error at each scale.

Moody (1989) has shown that a multiresolution, hierarchical training method is very efficient and well-suited for real-time adaptive learning. He used B-splines as activation functions, but also noted that they are not convenient for high-dimensional spaces. Poggio and Girosi's (1989) HyperBFs are the solution to approximation at multiple scales based on regularization theory. They have also pointed out the benefits of using multigrid techniques for training HyperBFs, instead of gradient descent in terms of improved efficiency. For a Gaussian HyperBF, the scales are determined by different σ values.

The basis functions used for these multiresolution networks are nonorthonormal. This makes the representations at different resolutions dependent and contain redundant information. Moody (1989) considers the approximation at each resolution to be independent and trains each level independently of the others, minimizing the error at each level while keeping the weights at the other levels constant. Due to the nonorthonormal activation functions, this procedure does guarantee minimum least-squares error at each level, but may not minimize the least-squares error of the overall approximation.

This redundancy may also cause an inefficient representation of the mapping. To understand this better, let us consider the structure of a "Moody-style" multiresolution network for interpolating regularly spaced training data. For a one-dimensional problem, we can interpolate data by assigning one unit for every data point in a single resolution network. Such a network will have N units, where N is the number of training examples. If we try to solve the same problem using a multiresolution network with nonorthonormal basis functions, we will need N units for the error signal at the finest scale, $N/2$ units for the error signal at the next coarser scale and so on until the coarsest signal. Therefore, the total number of units required will be (Daubechies, 1988):

$$N + \frac{N}{2} + \frac{N}{4} + \ldots + \frac{N}{2^{L-1}} + \frac{N}{2^L} = 2N\left(1 - \frac{1}{2^{L+1}}\right) \tag{24}$$

where L is the number of levels of resolution. This shows that for single-dimension systems, multiresolution networks could have twice the number of units as single resolution networks. This factor decreases for systems of higher dimensions. Thus, the number of units in a multiresolution network may be more than that in single-resolution networks. The redundancy of activation functions may be eliminated by choosing orthogonal activation functions, and extra units will not be required. Orthogonal wavelets generate such a multiresolution representation and are described in the next section.

Wavelets as Basis Functions for Neural Networks

A family of wavelets is derived from the translations and dilations of a single function. If $\psi(x)$ is the starting function, to be called a wavelet, the members of the family are given by

$$\frac{1}{\sqrt{s}}\psi\left(\frac{x-u}{s}\right) \quad \text{for } (s, u)\in R^2 \tag{25}$$

that is they are indexed by two labels s and u, with s indicating the dilation and u the translation of the base wavelet, $\psi(x)$. Although the family of wavelets given by Eq. 25 need not be orthogonal to each other, orthonormal families of wavelets have been constructed (Meyer, 1985; Daubechies, 1988; Mallat, 1989; Strang, 1989) and these are the only ones we will deal with in this article. They include the Meyer wavelet, the Haar wavelet, the Battle-Lemarie wavelets, and a class of orthonormal wavelets with compact support constructed by Daubechies.

If the input, x is defined in a discrete domain and if the dilation of the wavelet is always by a factor of 2, the resulting family of discrete dyadic wavelets is represented by

$$\sqrt{2^{-m}}\psi(2^{-m}x - k) \quad \text{for } (m, k)\in Z^2 \tag{26}$$

Note that m denotes the size of dilation (as a multiple of 2) and k the discrete-step translation of the wavelet, $\psi(x)$. They are related to the continuous parameters in Eq. 25 by,

$$s = 2^m, \quad u = k2^m$$

and indicate that both, the translation and dilation parameters are discretized dyadically. The translation and dilation of the Battle-Lemarie wavelet is shown in Figure 2.

Multiresolution analysis of functions

The pivotal concept, in the formulation and design of neural networks with wavelets as basis functions, has been Mallat's multiresolution representation of functions using wavelets. This theory provides the essential framework for the completely localized and hierarchical training afforded by Wave-Nets and will be discussed in some detail, following the style found in the articles of Mallat (1989) and Daubechies (1988).

Consider a continuous, square-integrable function, $F(x)\in L^2(R)$. Let $F_m(x) \equiv A_m F(x)$ denote the approximation of $F(x)$ at the resolution m, where 2^m is the sampling interval,

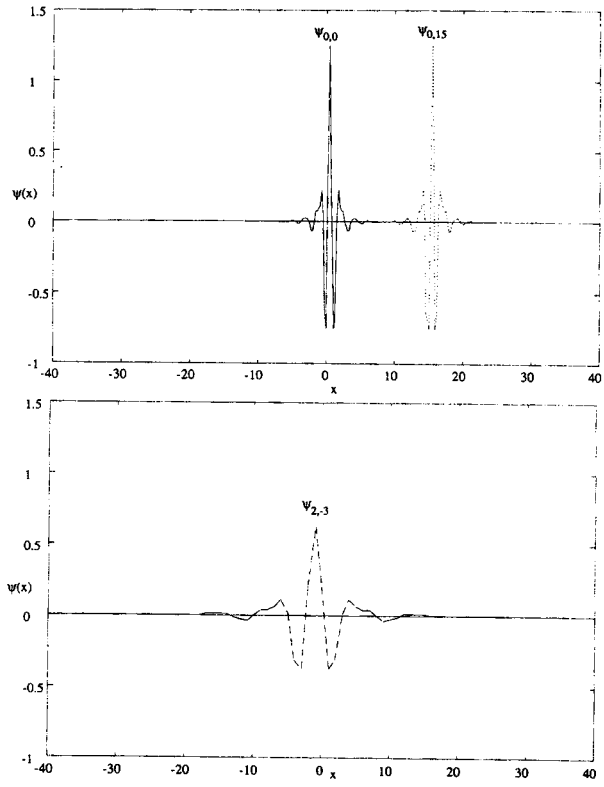

Figure 2. Translation and dilation of Battle-Lemarie wavelet.

that is, the interval between two consecutive sampled values used in the approximation. It is easy to see that 2^{-m} is the number of sampled values per unit length of input space. Consequently, as m increases the sampling interval increases, the number of samples per unit length decreases and the approximation $F_m(x)$ becomes coarser.

Let V_m be the vector space containing all possible approximations of $F(x)$ at the resolution 2^m. Then A_m is a projection operator on the space V_m. Furthermore, if $A_mF(x)$ is the best approximation of $F(x)$ at resolution 2^m, then operator A_m is an orthogonal projection on V_m.

Now, consider the approximations of $F(x)$ at all resolutions 2^m with $m=[-\infty, +\infty]$. The associated vector spaces, V_m, satisfy the following conditions:

(1) $\ldots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset \ldots$ (27)

This is a direct result of the causality property that requires the $A_{m-1}F(x)$ approximation (finer resolution) to contain all the information needed to compute the coarser approximation, $A_mF(x)$.

(2) If the approximations at the various resolutions are similar, then the spaces, V_m with $m \in Z$, of the approximated functions should be derivable from each other through scaling by the ratio of their resolutions, that is,

$$F_m(x) \equiv A_mF(x) \in V_m \Leftrightarrow F_{m-1} \equiv A_{m-1}F(2x) \in V_{m-1} \quad (28)$$

(3) As the resolution decreases, the resulting approximation of $F(x)$ contains less and less information and converges to zero, that is

$$\lim_{m \to +\infty} V_m = \bigcap_{m=-\infty}^{+\infty} V_m = \{0\} \quad (29)$$

(4) On the other hand, as the resolution increases, the resulting approximation converges to the original signal $F(x)$ where $F(x) \in L^2(R)$. Therefore,

$$\lim_{m \to -\infty} V_m = \bigcup_{m=-\infty}^{+\infty} V_m \text{ is dense in } L^2(R) \quad (30)$$

(5) For a given form of approximation, $A_mF(x)$ is represented by functions that are piece-wise constant, linear, quadratic, or cubic, and so on. It can be shown that (Mallat, 1989): "There exists a unique function, $\phi(x) \in V_0$, called a scaling function, such that for all $m \in Z$, the family of functions resulting from the dilation and translation of $\phi(x)$, that is:

$$\phi_{mk}(x) = \sqrt{2^{-m}}\phi(2^{-m}x - k) \quad (m, k) \in Z^2 \quad (31)$$

constitutes an unconditional orthonormal basis for V_m," that is:

$$V_m = \overline{\text{linear span}\{\phi_{mk}, k \in Z\}} \quad (32)$$

Mallat (1989) has also shown that there exists an isomorphism between any V_m and $l^2(Z)$, the vector space of all square-summable sequences of sampled values. This isomorphism implies that, if $F(x)$ is translated by a length proportional to the sampling interval 2^m, the $A_mF(x)$ is also translated by the same amount and is characterized by the same sampled values which have been translated. This translation invariance is not true if $F(x)$ is translated by a length not proportional to 2^m. The set of vector spaces, V_m, which satisfy the above properties has been called a multiresolution approximation of $L^2(R)$ (Mallat, 1989).

Since the approximation $A_mF(x) \in V_m$, then from property 5 we have

$$F_m(x) \equiv A_mF(x) = \sum_{k=-\infty}^{+\infty} a_{mk}\phi_{mk}(x) \quad (33)$$

with the basis functions given by Eq. 31, and the coefficients a_{mk} being the projections of $F(x)$ onto the orthonormal basis functions, ϕ_{mk}, that is,

$$a_{mk} = \int_{-\infty}^{+\infty} F(x)\phi_{mk}(x)dx \quad (34)$$

Let W_m be the orthogonal complement of V_m in V_{m-1}. Then, $V_{m-1} = V_m \oplus W_m$ with $V_m \perp W_m$. The $(m-1)$-th approximation of $F(x)$ can thus be written as

$$A_{m-1}F(x) = [A_m \oplus D_m]F(x) = A_mF(x) \oplus D_mF(x) \quad (35)$$

where D_m is a projection operator on the space W_m. Equation 35 indicates that the difference of information contained in the two approximations of $F(x)$ at the resolutions, m and $(m-1)$, is given by the orthogonal projection of $F(x)$ on the

vector space W_m. To quantify this difference, which from now on will be called the detail of $F(x)$ at the resolution m, we need to construct an orthonormal basis for the vector space W_m. Mallat (1989) has shown that there exists a unique function, $\psi(x)$, called a wavelet, whose dilations and translations form the family of functions

$$\psi_{mk}(x) = \sqrt{2^{-m}}\psi(2^{-m}x - k) \quad \text{for } (m, k) \in Z^2 \tag{36}$$

which in turn constitute an unconditional orthonormal basis of W_m. In view of the above result, it is clear from Eq. 35, that the detail of $F(x)$ at the resolution m will be given by

$$D_m F(x) = \sum_{k=-\infty}^{+\infty} d_{mk}\psi_{mk}(x) \tag{37}$$

where the coefficients d_{mk} are the projections of $F(x)$ onto the basis functions $\psi_{mk}(x)$, that is,

$$d_{mk} = \int_{-\infty}^{+\infty} F(x)\psi_{mk}(x)dx \tag{38}$$

Furthermore, from Eq. 35 and property 2 (see Eq. 28), we can easily see that the orthogonal union of all W_m spaces forms the vector space of all square-integrable functions, that is,

$$\bigoplus_{m \in Z} W_m = L^2(R) \tag{39}$$

Consequently, any $F(x) \in L^2(R)$ can be expanded into a set of orthonormal wavelets, that is,

$$F(x) = \sum_{m=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} d_{mk}\psi_{mk}(x) \tag{40}$$

Equation 40 describes what is known as the wavelet decomposition of a square-integrable function, and provides the theoretical framework for the design of Wave-Nets.

Finally, from Eqs. 35 and 37 we find that the approximation of $F(x)$ at scale $(m-1)$ is equal to

$$F_{m-1}(x) = F_m(x) + \sum_{k=-\infty}^{+\infty} d_{mk}\psi_{mk}(x) \tag{41}$$

Equation 41 summarizes the hierarchical, multiresolution representation of functions offered by the wavelet decomposition, and is identical in character to that discussed earlier in the third section (see Eq. 22).

Remarks on some practical aspects of the wavelet decomposition

Discrete functions. Instead of considering a continuous, square-integrable function, $F(x)$, let us focus on a sequence of discrete samples of $F(x)$, resulting from physical measurements. Call this sequence of measured samples, a_{0k}, $k = 1, 2, \ldots, n_0$. It represents an approximation of $F(x)$ at the highest physically measurable resolution. Let, V_0 be the vector space of this approximation. Then, from Eq. 33

$$F_0(x) = \sum_{k=-\infty}^{+\infty} a_{0k}\phi_{0k}(x)$$

and following the wavelet decomposition (Eqs. 35, 33 and 37) we have

$$\begin{aligned} F_0(x) &= \sum_{k=-\infty}^{+\infty} a_{0k}\phi_{0k}(x) \\ &= A_1F(x) + D_1F(x) \\ &= \sum_{k=-\infty}^{+\infty} a_{1k}\phi_{1k}(x) + \sum_{k=-\infty}^{+\infty} d_{1k}\psi_{1k}(x) \end{aligned} \tag{42}$$

The first term of Eq. 42 represents the first coarser approximation of the function $F_0(x)$, while the second term represents the included detail, that is, the information contained in $F_0(x)$ but filtered out in $A_1F(x)$. Let us now see how to compute the coefficients a_{1k} and d_{1k}.

Multiply both sides of Eq. 42 by $\phi_{1k}(x)$ and integrate. Since ϕ_{1k} is orthonormal to ϕ_{1n} and ψ_{1n}, the resulting equation yields

$$a_{1k} = \sum_{k=-\infty}^{+\infty} a_{0k} \int_{-\infty}^{+\infty} \phi_{0k}(x)\phi_{1k}(x)dx \tag{43a}$$

Similarly, by multiplying Eq. 42 by $\psi_{1k}(x)$ and integrating we get,

$$d_{1k} = \sum_{k=-\infty}^{+\infty} a_{0k} \int_{-\infty}^{+\infty} \phi_{0k}(x)\psi_{1k}(x)dx \tag{43b}$$

By defining filters H and G in such a way that their impulse responses are given by

$$h_k = \int_{-\infty}^{+\infty} \phi_{0k}(x)\phi_{1k}(x)dx \tag{44a}$$

$$g_k = \int_{-\infty}^{+\infty} \phi_{0k}(x)\psi_{1k}(x)dx \tag{44b}$$

Eqs. 43a and b yield

$$a_{1k} = \sum_{k=-\infty}^{+\infty} h_k a_{0k} \quad \text{and} \quad d_{1k} = \sum_{k=-\infty}^{+\infty} g_k a_{0k} \tag{45a, b}$$

Filters H and G form a pair of quadrature mirror filters (Esteban and Galand, 1977).

To summarize, from Eq. 42 we have the recursive decomposition of a given discrete sequence of samples characterized by

$$A_{m-1}F(x) = \sum_{k \in Z} a_{mk}\phi_{mk}(x) + \sum_{k \in Z} d_{mk}\psi_{mk}(x) \tag{46}$$

with the coefficients of the decomposition given by

$$a_m = Ha_{m-1} \quad \text{and} \quad d_m = Ga_{m-1} \tag{47}$$

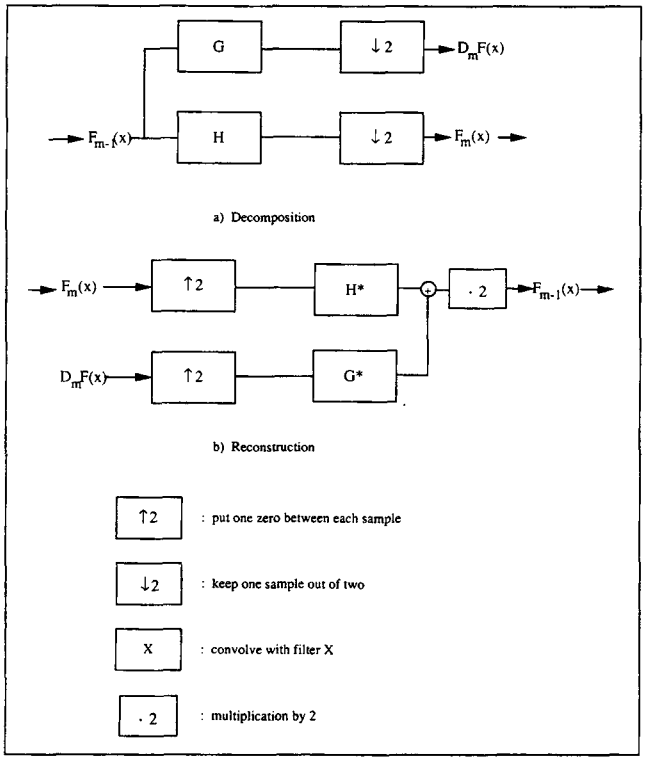

Figure 3. Multiscale decomposition of discrete data using wavelets (Mallat, 1989).

Figure 3a gives a schematic representation of the above decomposition. With 2 as the factor for the resolution of two successive approximations, each approximation contains half as many samples per unit of x as the immediately previous approximation. Then, the decomposition given by Eq. 46 leads to a grid of coefficients as shown in Figure 4.

Furthermore, the original data can be reconstructed through the recursive application of the following equation

$$a_{m-1} = H^* a_m + G^* d_m \tag{48}$$

where, H^* and G^* are the conjugates of H and G, respectively. Figure 3b presents a schematic of the reconstruction algorithm.

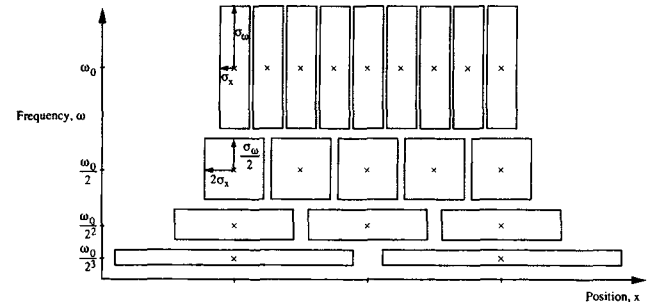

Figure 5. Input-frequency localization for wavelets at various translations and dilations.

Finite length. All the developments in the earlier paragraphs were based on infinite length sequences of sampled values. Adapting those equations to the case of finite sequences while preserving both the orthogonality and invertibility of the transformation (Chou, 1991) is not trivial. Mallat (1989) overcomes this problem of "end effects" by considering a mirror image of the trend beyond its end points. Chou indicates that this assumption violates the orthogonality of the filters, and has suggested three approaches which resolve this problem by defining appropriate H and G filters. For more details the reader is referred to Chou's PhD thesis. The design and training of Wave-Nets is based on examples of finite length with consistent wavelet decompositions. We do not make any assumptions about the behavior of the function beyond the end points in the input space, and the system "learns" the extrapolation based on the training data.

Localization in space and wave number. The principal benefit from the wavelet decomposition, which allows Wave-Nets to be locally trained, is the localized characterization of a continuous or discrete function in the input space, and wave number (or frequency, or scale). The input-frequency localization of wavelets at various translations and dilations is shown in Figure 5. Each rectangle indicates the input space and scale space localization of the corresponding wavelet. The size of each rectangle is determined by the standard deviation of the wavelet and its Fourier transform. The area of each rectangle is constant, indicating that as the frequency range increases, the input range decreases, as governed by the uncertainty principle. The information contained in the input and frequency

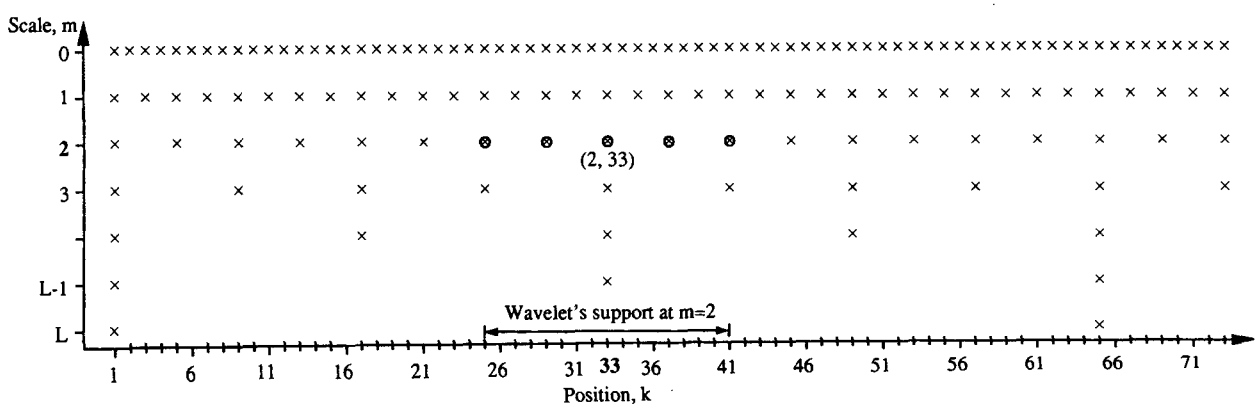

Figure 4. Grid for locating wavelets at various scales.

range covered by each wavelet or scaling function is captured by the coefficients d_{mk} and a_{mk} respectively.

Consider coefficient $d_{2,33}$, and the corresponding point, (2, 33) in the grid of Figure 4. The value of $d_{2,33}$ measures the content of the original signal in terms of the wavelet at the 2-nd dilation, when the input takes on values in the range $[33-q, 33+q]$. In other words, it measures the content of the original signal in the frequency range corresponding to the frequencies allowed at scale 2, and in the input range $[33-q, 33+q]$. This range is indicated by the encircled points in Figure 4. Here q is assumed to be 2 units. The value of q is determined by the extent of the wavelet's local support, or equivalently, the extent of the finite impulse response of H or G. Daubechies (1988) has presented a variety of wavelets with compact supports ranging from 4 to 20 samples. The Haar wavelet has a support of 2, while the Battle-Lemarie wavelet does not have compact support, but is exponentially decaying and may be considered to have a "practical" support of 23 sampled values.

Extension to higher dimensions. The multiresolution analysis, described in the section on multiresolution analysis of functions for a single input variable, can be easily extended to the multi-input case. In this paragraph, following Daubechies (1988), we present a brief description of this extension to two-dimensions, while its expansion to systems of higher dimensionality is straightforward.

Assume that we have a one-dimensional multiresolution analysis, that is, a set of vector spaces V_m and the associated scaling and wavelet functions ϕ and ψ. Define the set of vector spaces $\overline{V}_m$, $\overline{V}_m \in L^2(R^2)$ by

$$\overline{V}_m = V_m \oplus V_m \tag{49}$$

It is easy to show that the set $\{\overline{V}_m;\ m \in Z\}$ satisfies properties 1, 3, and 4 (see Eqs. 27, 29, and 30 respectively) of the multiresolution analysis. Defining

$$\Phi(x_1,x_2) = \phi(x_1)\phi(x_2) \tag{50}$$

one can show that

$$\overline{V}_m = \overline{\text{linear span}\{\Phi_{mn}(x_1,x_2); n \in Z^2\}}$$

where,

$$\begin{aligned}\Phi_{mn}(x_1,x_2) &= 2^{-m}\Phi(2^{-m}x_1 - n_1, 2^{-m}x_2 - n_2) \\ &= \phi_{mn_1}(x_1)\phi_{mn_2}(x_2)\end{aligned} \tag{51}$$

Recall that W_m is the orthogonal complement of V_m in V_{m-1}. Therefore,

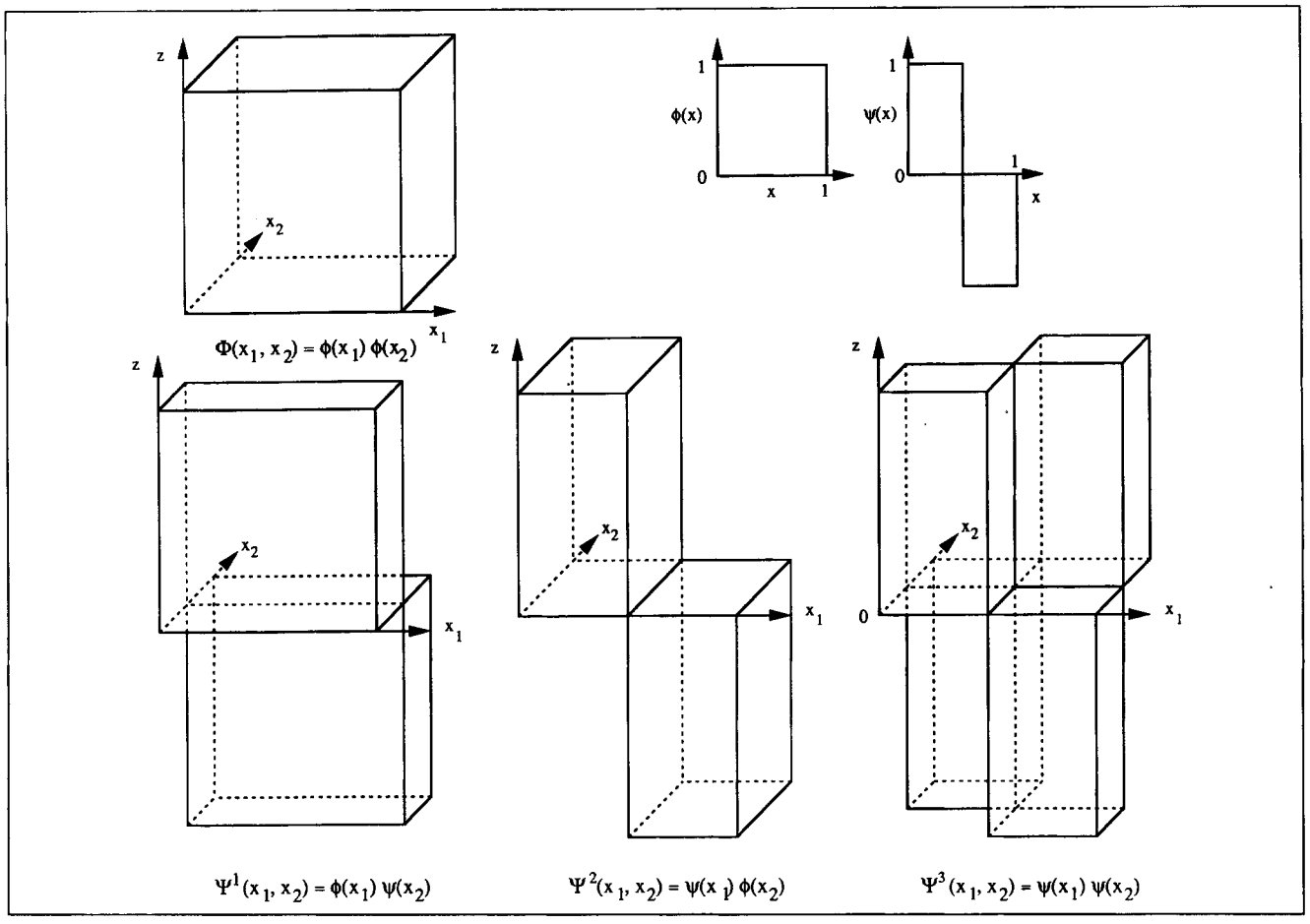

Figure 6. Haar scaling function and wavelets for two-dimensional problems.

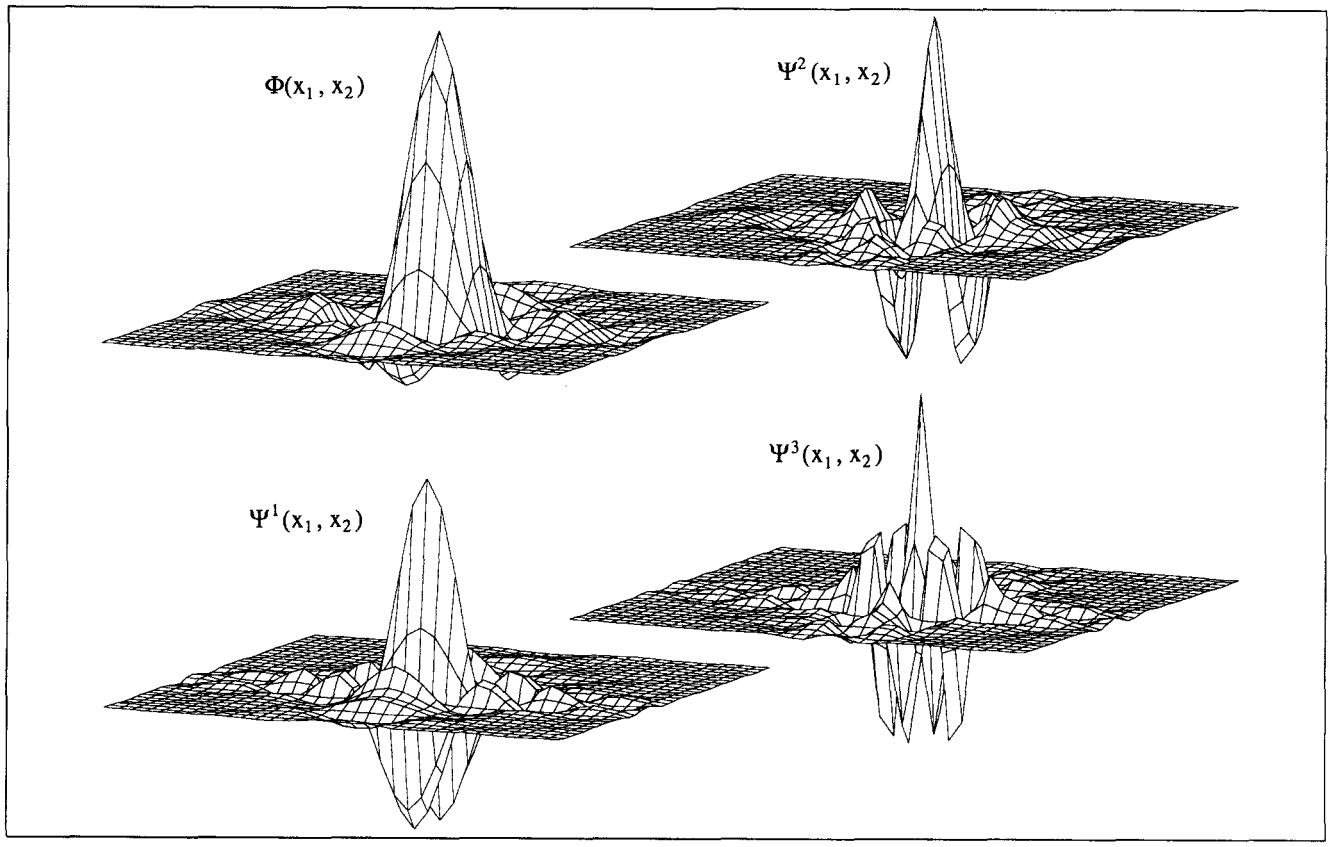

Figure 7. Battle-Lemarie scaling function and wavelets for two-dimensional problems.

$$\overline{V}_{m-1} = V_{m-1} \oplus V_{m-1} = [V_m \oplus W_m] \oplus [V_m \oplus W_m]$$
$$= \overline{V}_m \oplus [(V_m \oplus W_m) \oplus (W_m \oplus V_m) \oplus (W_m \oplus W_m)]$$

The last equation implies that the orthogonal complement $\overline{W}_m$ of $\overline{V}_m$ in $\overline{V}_{m-1}$ is given by

$$\overline{W}_m = (V_m \oplus W_m) \oplus (W_m \oplus V_m) \oplus (W_m \oplus W_m)$$

with an orthonormal basis given by the functions $\phi_{mn_1}\psi_{mn_2}$, $\psi_{mn_1}\phi_{mn_2}$, $\psi_{mn_1}\psi_{mn_2}$, where n_1, $n_2 \in Z$, or equivalently, by the two-dimensional wavelet Ψ^l_{mn},

$$\Psi^l_{mn}(x_1,x_2) = 2^{-m}\Psi^l(2^{-m}x_1 - n_1, 2^{-m}x_2 - n_2)$$

where $l = 1, 2, 3$, $n \in Z^2$ and

$$\Psi^1(x_1,x_2) = \phi(x_1)\psi(x_2) \tag{52a}$$

$$\Psi^2(x_1,x_2) = \psi(x_1)\phi(x_2) \tag{52b}$$

$$\Psi^3(x_1,x_2) = \psi(x_1)\psi(x_2) \tag{52c}$$

Therefore, the Ψ^l_{mn} with $l = 1, 2, 3$, $m \in Z$, $n \in Z^2$, constitute an orthonormal basis of wavelets for the $L^2(R^2)$ space. Each Ψ^l_{mn} captures information at different orientations as shown for the two-dimensional Haar and Battle-Lemarie wavelets in Figures 6 and 7 respectively.

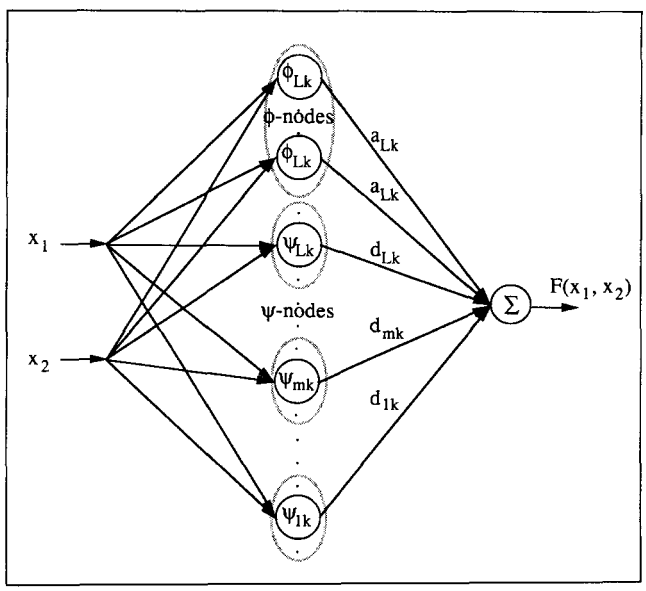

Figure 8. Structure of a typical Wave-Net.

Wavelets as basis functions for hierarchical, multiresolution networks

A Wave-Net is a network of nodes with structure shown in Figure 8. It consists of input and output nodes and two types of hidden-layer nodes: wavelet nodes, or ψ-nodes, and scaling

function nodes, or ϕ-nodes. The location and scale of the hidden-layer nodes are as shown in the grid of Figure 4. The basis functions associated with the hidden layer nodes are:

(a) Basis functions for ϕ-nodes: $\phi_{Lk}(x)$, $k=1, 2, \ldots, n_L$; the translates of the dilated scaling function, $\phi(x)$, which form the orthonormal basis for the approximation of the unknown function, $F(x)$, at the L-th (coarsest) resolution.

(b) Basis functions for ψ-nodes: $\psi_{mk}(x)$, with $m=1, 2, \ldots, L$ and $k=1, 2, \ldots, n_m$; the translates of the dilated wavelet, $\psi(x)$, at resolutions from 1 to L, which form the orthonormal basis for the detail of function $F(x)$ at each resolution.

Hierarchical, multiresolution learning. Consider the approximation of the unknown function, $F(x)$, at the L-th resolution; that is, $F(x)$ is represented by 2^{-L} samples per unit length of x. The lowest resolution possible along an input dimension for any function consists of two grid points, as shown at the L-th dilation level in Figure 4. For each dimension, $L=\inf\{\log_2 N_g\}$, where N_g is the size of the grid at the highest resolution. Then, the resulting network (see Figure 9a) has n_L ($=2$ for one dimension) nodes, which are all ϕ-nodes with their position given by the grid of Figure 4. The minimization of the mean-squares error (between prediction and measurements) yields the coefficients of the following approximation

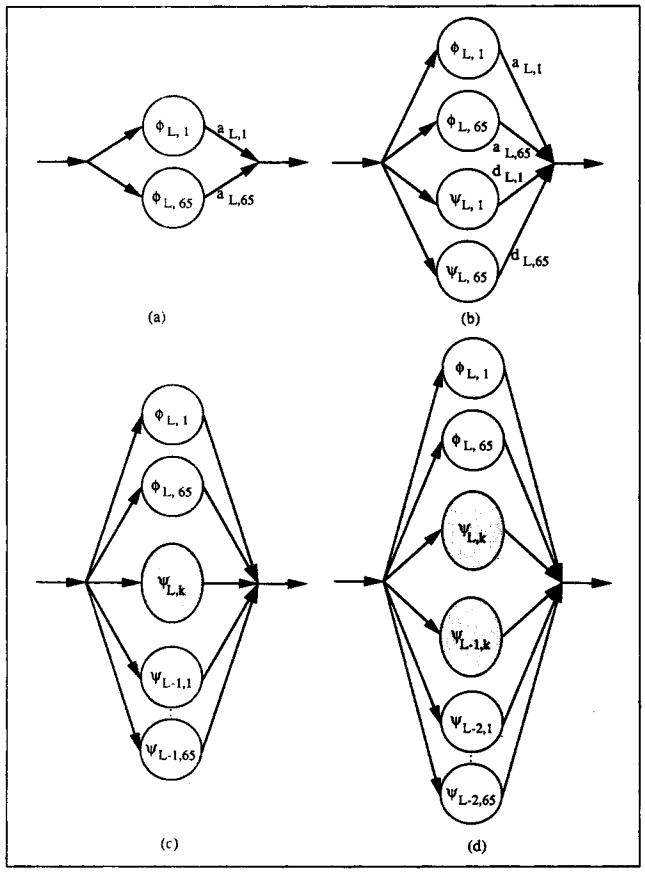

Figure 9. Construction of Wave-Net at different resolutions.

(a) ϕ-nodes at scale L; (b) with ψ-nodes at scale L; (c) with ψ-nodes at scale (L − 1); (d) with ψ-nodes at scale (L − 2) (Refer to grid of Figure 4).

$$F(x) \approx F_L(x) = \sum_{k=1}^{n_L} a_{Lk}\phi_{Lk}(x) \tag{53}$$

Equation 53 represents the first approximation of $F(x)$ at the coarsest, L-th resolution, and from the discussion on the multiresolution of functions we know that, $F_L(x) \in V_L$.

Now, suppose that we want a more refined approximation of $F(x)$, resulting in a lower global error. Such an approximation can be found from the orthogonal projection of the unknown function onto the vector space V_{L-1}, yielding

$$F(x) \approx F_{L-1}(x) = \sum_{k=1}^{2n_L} a_{L-1k}\phi_{L-1k}(x) \tag{54a}$$

In such a case we would have a network with $2n_L$ ϕ-nodes (twice as many as in the previous case), whose $2n_L$ coefficients would have to be computed through the minimization of the new mean-square error. Consequently, we have gained nothing by constructing the approximation in Eq. 53.

Instead, we observe that $F_{L-1}(x) \in V_{L-1}$ and we note from the multiresolution analysis that $V_{L-1} = V_L \oplus W_L$. Therefore,

$$F_{L-1} \in V_L \oplus W_L$$

and the approximation in Eq. 54a can be written in the following form

$$F(x) \approx F_{L-1}(x) = \sum_{k=1}^{n_L} a_{Lk}\phi_{Lk}(x) + \sum_{k=1}^{n_L} d_{Lk}\psi_{Lk}(x) \tag{54b}$$

where, $\psi_{Lk}(x)$; $k=1, 2, \ldots, n_L$ are the wavelet functions forming an orthonormal basis for the vector space, W_L.

Since $W_L \perp V_L$, the $\psi_{Lk}(x)$, $k=1, 2, \ldots, n_L$ are not only orthogonal to each other but are also orthogonal to the $\phi_{Lk}(x)$, $k=1, 2, \ldots, n_L$ functions. Therefore, we reach the following conclusions:

(a) The first term of the approximation (Eq. 54b), that is, $\sum_{k=1}^{n_L} a_{Lk}\phi_{Lk}(x)$, represents the approximation of the function at the L-th resolution and it has already been computed. In other words, the coefficients, a_{Lk}, are the same as those in Eq. 53.

(b) The second term of the approximation in Eq. 54b, that is, $\sum_{k=1}^{n_L} d_{Lk}\psi_{Lk}(x)$, represents the added information, as we move from the coarser approximation in Eq. 53 to the more detailed one in Eq. 54b.

(c) The approximation in Eq. 54b is represented by a network (see Figure 9b), which is an extension of the network constructed for approximation in Eq. 53 (see Figure 9a). Specifically, in the n_L ϕ-nodes of the network in Figure 9a we have added n_L ψ-nodes with basis functions, $\psi_{Lk}(x)$, $k=1, 2, \ldots, n_L$. The new network is trained only for the coefficients, d_{Lk}, $k=1, 2, \ldots, n_L$ (since a_{Lk}, $k=1, 2, \ldots, n_L$ are already known), using the errors from the first approximation as the data for training.

Continuing with the addition of ψ-nodes at the resolutions,

$(L-2)$, $(L-3)$, ... and so on, the resulting networks (see Figure 9c, d) approach the unknown function with progressively smaller global error. Specifically,

$$F_{L-2}(x) = \left\{ \sum_{k=1}^{n_L} a_{Lk}\phi_{Lk}(x) + \sum_{k=1}^{n_L} d_{Lk}\psi_{Lk}(x) \right\} + \sum_{k=1}^{2n_L} d_{L-1,k}\psi_{L-1,k}(x) \quad (55a)$$

$$= F_{L-1}(x) + \sum_{k=1}^{2n_L} d_{L-1,k}\psi_{L-1,k}(x)$$

$$F_{L-3}(x) = F_{L-2}(x) + \sum_{k=1}^{4n_L} d_{L-2,k}\psi_{L-2,k}(x) \quad (55b)$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$F_0(x) = F_1(x) + \sum_{k=1}^{2^{L-1}n_L} d_{1k}\psi_{1k}(x)$$

$$= \sum_{k=1}^{n_L} a_{Lk}\phi_{Lk}(x) + \sum_{m=1}^{L} \sum_{k=1}^{2^{L-m}n_L} d_{mk}\psi_{mk}(x) \quad (55c)$$

The above equations characterize the hierarchical, multiresolution learning inherent in a Wave-Net, which is identical in nature to that proposed by Moody (1989).

Computing Wave-Net coefficients. The location and size of the Wave-Net activation functions is given by the grid shown in Figure 4. If the data are regularly sampled in the input space, then the Wave-Net coefficients $\{a_{Lk}, d_{mk}\}$ for all (m, k) on the grid are easily computed by Eq. 47 and Mallat's multiscale decomposition procedure may be directly applied. Unfortunately, for most learning problems, data may not be regularly sampled in the input space. Then the Wave-Net coefficients are computed as described below.

At any level of approximation (see Eq. 55a, b, or c), the Wave-Net represents the input-output map by an expression of the form,

$$F(x) = \sum_i c_i\theta_i(x)$$

where, c_i are the weights and θ_i are the activation functions, representing the corresponding scaling or wavelet functions. Consider the case where we construct a Wave-Net with n nodes, and have m training data. In matrix form we have

$$\begin{pmatrix} F(x_1) \\ F(x_2) \\ \vdots \\ F(x_m) \end{pmatrix} = \begin{pmatrix} \theta_1(x_1) & \theta_2(x_1) & \dots & \theta_n(x_1) \\ \theta_1(x_2) & \theta_2(x_2) & \dots & \theta_n(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \theta_1(x_m) & \theta_2(x_m) & \dots & \theta_n(x_m) \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix}$$

or,

$$F = Ac \quad (56)$$

The least-squares solution of this equation is given by

$$c = ((A^TA)^{-1}A^T)F$$
$$= A^+F \quad (57)$$

where A^+ is called the pseudo-inverse of matrix A. Notice that

$$A^TA = \begin{pmatrix} \sum_i \theta_1(x_i)^2 & \sum_i \theta_1(x_i)\theta_2(x_i) & \dots & \sum_i \theta_1(x_i)\theta_n(x_i) \\ \sum_i \theta_1(x_i)\theta_2(x_i) & \sum_i \theta_2(x_i)^2 & \dots & \sum_i \theta_2(x_i)\theta_n(x_i) \\ \vdots & \vdots & \ddots & \vdots \\ \sum_i \theta_1(x_i)\theta_n(x_i) & \sum_i \theta_2(x_i)\theta_n(x_i) & \dots & \sum_i \theta_n(x_i)^2 \end{pmatrix} \quad (58)$$

If the samples x_i are uniformly spaced and $\theta_i(x)$ form an orthonormal basis, for $i = 1, 2, \dots, n$, then A^TA is an identity matrix,

$$A^TA = I_{n\times n}$$

Matrix inversion is not necessary to determine the coefficients, and

$$c = A^TF \quad (59)$$

Mallat's multiscale decomposition procedure, described earlier, solves Eq. 59 by defining the appropriate H and G filters. If the data are irregularly sampled, then the weights are determined by Eq. 57. In this case, the local nature of the activation functions causes the matrix of Eq. 58 to be a band diagonal matrix with a band of width $2q+1$. The need to invert a matrix increases the computational complexity of determining the weights for irregularly sampled data, than for regularly sampled data by one order of magnitude. The computational complexity aspects of a Wave-Net will be described in the next main section.

Localized learning. Learning at the coarsest, L-th, resolution (see Eq. 53) involves the determination of the coefficients a_{Lk} over the whole range of x-values. In other words, Eq. 53 provides a global regressor. The local support of each ϕ_{Lk} provides the basis for the localization of the learning process. Furthermore, as soon as we move to resolutions $(L-1)$, $(L-2)$, ..., 1 the local support of the wavelets, ψ_{mk}, implies that the coefficients of the ψ-nodes allow an efficient mechanism for localized learning within the scope of different resolutions.

As shown in Figure 5, since wavelets are localized in input and scale space, the nodes in a Wave-Net can be selected to reflect the distribution of data in an explicit manner. In the construction of a Wave-Net depicted in Figure 9, only those ψ-nodes are introduced that have data within its input and

scale domain. This localization in learning is very useful for estimating and reducing the local error of approximation during training and adaptation.

Global approximation error. Equation 55c yields the best approximation of the unknown function, but it represents an interpolation rather than a generalization of the training data. For generalization of the input-output mapping of the training data, we select to retain some of the terms appearing in Eq. 55c. Let S_+ represent the set of wavelet coefficients, d_{mk}, retained and S_- the set of coefficients neglected. Then, the variance of the global error of the resulting generalized mapping is given by (see discussion on approximation by expansion on an orthonormal basis and Eq. 19)

$$e^2_{\text{global}} = \sum_m \sum_k d^2_{mk} \quad \text{with } (m,k)\in S_- \tag{60a}$$

Clearly, the smaller the value of $|d_{mk}|$ the smaller the resulting error from the omission of the term (and the corresponding node in the Wave-Net), $d_{mk}\psi_{mk}(x)$. This result is the basis for the practical design of a reduced Wave-Net and indicates how one can select the hidden-layer nodes in a systematic and rigorous manner.

One additional note is in order. In all the previous analysis it has been assumed that the training data contain unique information between an input value, x_i and the corresponding output value, $F_i = F(x_i)$. It is possible though that in the training data, the same x_i has produced several different values of the output. In such cases, we take the mean output

$$F_i = F(x_i) = \frac{1}{N}\sum_{j=1}^{N} F_i^{(j)}$$

where N is the number of points at the input x_i. Then, the variance of the global error of Eq. 60a must be adapted with an additional term to account for the contribution of the interpolation error resulting from the use of mean output values, that is:

$$e^2_{\text{global}} = \left\{\sum_m \sum_k d^2_{mk}\right\}_{(m,k)\in S_-} + \left\{\sum_m \sum_k \frac{1}{N}\sum_{j=1}^{N} [d_{mk} - d^{(j)}_{mk}]^2\right\}_{(m,k)\in S_+} \tag{60b}$$

where $d^{(j)}_{mk}$ is the wavelet coefficient for the j-th sample at the same input value.

Local approximation error. In addition to the global error, it is very important that one has a measure of the induced approximation error over a local range of input values. Such a measure is very valuable in calculating the reliability of the approximation, especially when the training data are unevenly distributed in various ranges of input values. Regions in the input space with higher density of examples give a smaller error of approximation, that is, higher reliability. Leonard et al. (1991) have developed such a measure for RBFNs.

Consider the two-dimensional grid of points in Figure 10 illustrating the Wave-Net's nodes resulting from the wavelet decomposition. Let $2q+1$ be the support of the wavelet over the input space. In Figure 10, $q=2$ and the support of $2q+1=5$ sampled values is shown by the encircled points in the first row of the grid around position "l." In the same figure one may see the encircled grid points at higher resolutions falling in the same interval of support. Then, the local error at position "l" depends on the following two contributions:

(1) The errors induced by neglecting the coefficients of those terms in the support interval $[l-q,\ l+q]$ around "l," that is, those terms represented by the encircled grid points of Figure 10 which have been neglected in Eq. 55c.

(2) The interpolation error through the terms which were retained in Eq. 55c and which terms are in the interval $[l-q,\ l+q]$.

Consequently, the variance of the local error at position "l" is given by

$$e^2_{\text{local}} = \left\{\sum_m \sum_{k=l-q}^{l+q} d^2_{mk}\right\}_{(m,k)\in S_-} + \left\{\sum_m \sum_{k=l-q}^{l+q} \frac{1}{N}\sum_{j=1}^{N} [d_{mk} - d^{(j)}_{mk}]^2\right\}_{(m,k)\in S_+} \tag{61}$$

Size of the network. The output of the selected ϕ-units is independent of each other due to the orthonormality of ϕ_{Lk}'s for all k. Similarly, the output of each ψ-unit is also independent due to the orthonormality of ψ_{mk} for all $(m,\ k)\in Z^2$. Such nonredundant representation requires less nodes to reconstruct the data than those required by Moody's network. The advantage of using orthonormal activation functions in

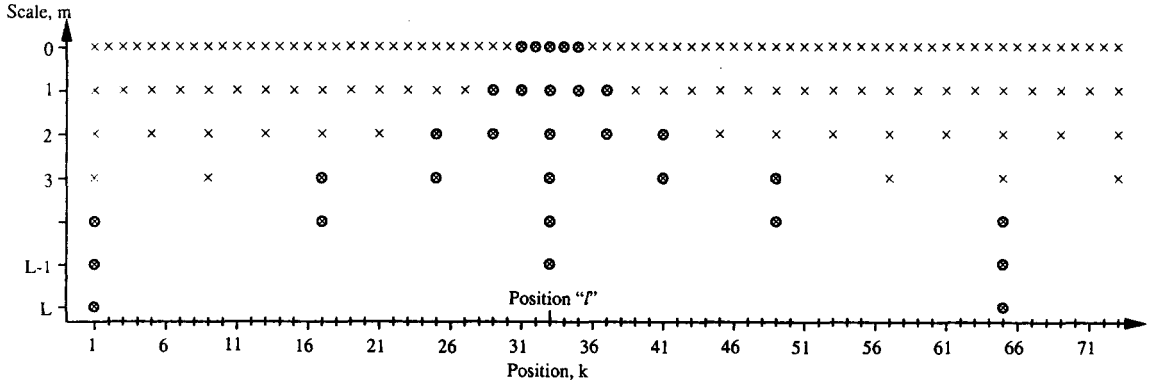

Figure 10. Extent of wavelet influence in input-space, at various scales, for a point at position "l".
Wavelet has support $2q+1=5$.

each node becomes clearer by an analysis of the number of units required for interpolation of regularly spaced training data. A similar analysis was performed for Moody's network in an earlier section. The number of points is halved at each scale of the wavelet decomposition. Thus, for M training data the first detail signal has $M/2$ points, the second $M/4$ and so on. Therefore, the total number of points arising from the decomposition at L scales is,

$$\frac{M}{2}+\frac{M}{4}+\ldots+\frac{M}{2^{L-1}}+\frac{M}{2^L}+\frac{M}{2^L}=M$$

where the last $M/2^L$ points are for the last scaled signal. Thus, a multiresolution hierarchy using orthonormal wavelets uses a maximum of M units for complete interpolation as compared to $2M[1-(1/2^{L+1})]$ required by Moody's network. Thus, the Wave-Net representation of input-output maps may provide a more efficient approximation because it requires the smallest number of units needed for interpolation of M points.

A Wave-Net can be represented by Moody's network by having only scaling functions in each node, and calculating the error by explicit subtraction. This does not guarantee independence of the output of nodes at different scales and makes the representation redundant. Thus, a Wave-Net has all the properties of Moody's network, and more.

Practical Aspects in the Design and Training of Wave-Nets

The Wave-Net training procedure for a problem with multiple inputs and outputs is outlined below:

(1) Select appropriate wavelet and scaling functions.
(2) Create the grid of coefficients at multiple resolutions for each dimension.
(3) Train ϕ-nodes at highest scale, L.
(4) Until training data are overfitted,
  (a) Determine approximation error.
  (b) Add appropriate ψ-nodes to reduce approximation error.
(5) Optimize network by crossvalidation with new data and remove ψ-nodes with small weights till performance criteria are satisfied.

Each step of this procedure is described below. We also provide theoretical measures on the computational complexity of each activity.

Selection of appropriate family of wavelet functions. The first step in the design of a Wave-Net is to select the appropriate set of wavelets and the corresponding scaling function for the given learning problem. As mentioned earlier, several different types of orthonormal wavelets are available. The nature of the hypersurface learned by the Wave-Net will depend on the nature of the wavelet activation function in each node. For example, if a continuous discriminant surface is to be learned, then the wavelet and scaling functions should allow smooth approximation of the data. The Battle-Lemarie wavelet is well-suited for such problems. In this article, we have used the cubic Battle-Lemarie wavelet, which possesses continuous derivatives up to third order, but can be extended to allow higher degrees of smoothness (Battle, 1987). Poggio and Girosi (1989) have described the relationship between the number of examples needed to achieve a given error rate, and the smoothness of the basis functions. A high degree of smoothness may be essential for achieving reasonable rates of convergence with a limited number of training examples for multidimensional problems. In such cases, the appropriate Battle-Lemarie wavelet may be chosen. Note that the support of the Battle-Lemarie wavelet increases with its smoothness. In many classification problems, the hypersurface to be approximated may be Boolean or discrete. The Haar wavelet may be used for learning such surfaces.

The shape of the receptive field of the basis functions also depends on the type of wavelet selected for the Wave-Net. For example, the Haar wavelet has rectangular receptive fields, whereas the Battle-Lemarie wavelet has elliptical receptive fields. The ratio of the major and minor axes depends on the ratio of the grid spacing along each input dimension. The wavelets shown in Figures 6 and 7 have square and circular receptive fields respectively, and correspond to equal grid spacing along both input dimensions. Various criteria may be used to decide the grid spacing, like the covariance matrix, as used by Holcomb and Morari (1991). The example in the section on fault diagnosis illustrates the use of rectangular receptive fields. Here, the grid spacing is determined by the ratio of the smallest sampling rates along each dimension.

Another decision necessary before training the Wave-Net is the dimensionality of the wavelets. This depends on the number of inputs in the problem being solved. As described earlier, orthonormal wavelets of the desired dimensions may be constructed by using Eqs. 51 and 52.

Hierarchical, multiresolution training of the complete network. The next step is to construct a grid similar to that of Figure 4, for each input dimension. This is used for locating the wavelets and scaling functions. The grid spacing at the finest resolution ($m=0$) is taken equal to the smallest, nonzero sampling rate along each dimension. The grid is developed by taking alternate points at each scale, and the highest scale, L is obtained when only two points are left. Therefore,

$$L=\inf\{\log_2(N_g)\}$$

where N_g is the number of points in the grid at the finest scale. The location of the basis functions for the ϕ- and ψ-nodes is defined by this grid. Note that training data may be available anywhere in the input space. Data points do not have to lie on the grid points.

The construction of the Wave-Net can now begin. We start with the coarsest scale, L, that is common to all inputs. A global regression to the hypersurface being learned is provided by training n_L ϕ-nodes at dilation L and translation as given by the grid. The weights are determined by applying Eq. 57. The predictions for the training data are computed from this network, and the global and local errors are determined. Construction of the Wave-Net is continued if the desired performance criteria are not satisfied. The performance of the Wave-Net at each stage of the training may be determined by using a set of testing data.

We now add ψ-nodes to the network to reduce prediction error. The location of the wavelet basis functions is chosen so as to reduce prediction error in regions where it is high. Any wavelet that has the input value within its input or scale range

will influence the local error. We use the following two criteria for selecting wavelets to add to the Wave-Net:

- Input space localization criterion: The region in the input space where the approximation error is to be minimized should lie within the support of the selected wavelet. This locates wavelets appropriately in the input space.
- Scale space localization criterion: Training or testing data should be available in the frequency range covered by the wavelet. This prevents selection of wavelets spanning regions where no data are available, and locates the wavelets appropriately in the scale space.

Wavelets satisfying these criteria are added to the network, and the weights are determined by Eq. 57. New ψ-nodes, added to the Wave-Net, are trained to minimize the error from the coarser scales. The weights of the ϕ-nodes and ψ-nodes at other scaling levels are unaffected by the addition of new ψ-nodes. This is because new ψ-nodes reduce the approximation error of the network developed so far. Since the error is the minimum least-squares error, the space in which the residual error lies is perpendicular to the space of the approximations by the nodes introduced so far. New ψ-nodes approximate this error space. The orthonormality of the wavelets ensures that the new wavelets lie entirely in the remaining error space, and do not interfere with contributions of wavelets in any other space. Therefore, independent training of each node is possible. This procedure is continued till the training data are overfitted, or the desired generalization is attained.

The Wave-Net performance is determined by testing its predictive capability on new data. This network is not guaranteed to be optimal since some of the nodes may not be contributing much to the approximation, and the theorem in the section on approximation by expansion on an orthonormal basis may not be satisfied. An optimal network may be obtained by overfitting the data, and then removing nodes that have small weights. The Wave-Net performance is checked on testing data. If the training data are overfitted, the Wave-Net prediction error on testing data goes through a minimum as wavelets of higher resolutions are selected. Thus, the optimum Wave-Net structure may be determined.

Computational complexity. The computational complexity of the training procedure is between $O(N)$ and $O(N^2)$ where N is the total number of training data. Construction of the grid takes constant time. Determination of the weights for the Wave-Net's nodes requires computing the pseudo-inverse of the matrix A as shown in Eq. 57. As described earlier, if the data are irregularly sampled in the input space, the matrix A^TA is a block diagonal matrix with less than or equal to $2q+1$ terms in each row, due to the localized nature of the activation function receptive fields. Inversion takes $O(qn^2)$ time where n is the number of nodes being trained. On the other hand, if the data are regularly sampled, then the cross terms disappear, and the matrix A^TA is an identity matrix, and computing the pseudo-inverse takes $O(qn)$ time. The number of nodes in the Wave-Net is $O(N)$, which yields complexity bounds of $O(N)$ for training with regularly sampled data, and $O(N^2)$ for training with irregularly sampled data.

These complexity measures represent a significant improvement over other types of neural net learning methods. No formal complexity results are available for BPNs and RBFNs since the training relies on trial and error. Empirical results indicate that BPNs train in $O(N^3)$ time (Hinton, 1990). This does not include the trial and error computations. For RBFNs, the clustering process of phase 1 requires $O(N^3)$ time, while the data interpolation takes $O(qN^2)$ time. But, it should be noted that in practice, the training takes much longer due to the heuristic trial and error involved. The actual training procedure for both BPNs and RBFNs requires trial and error for designing the appropriate network. BPNs may also face convergence problems. Therefore, in practice, both BPNs and RBFNs take a significant amount of time to train and adapt.

It should be noted that in the multivariable case of n independent inputs, there are (2^n-1) distinct wavelet functions at each grid point. For example, if $n=2$, we have the three wavelets, Ψ^1, Ψ^2, Ψ^3 encountered in the fourth section. Consequently, at each grid point there could be (2^n-1) nodes, requiring the computation of (2^n-1) distinct coefficients, a rather cumbersome numerical task for large n. In order to maintain a reasonable sized network, one can construct linear combinations of the distinct wavelet functions, that is,

$$\Psi^*(x_1,x_2,\ldots,x_n) = W^T\Psi(x_1,x_2,\ldots,x_n)$$

where,

$$\Psi^T(x_1,x_2,\ldots,x_n) = [\Psi^1(x_1,\ldots,x_n),\Psi^2(x_1,\ldots,x_n),\ldots,\Psi^{2^n-1}(x_1,\ldots,x_n)]$$

$\Psi^l(x_1,x_2,\ldots,x_n)$, $l=1,2,\ldots,(2^n-1)$ are the distinct wavelet functions and W is a vector of constant coefficients. The mutual orthogonality of the Ψ^l functions implies that the functions $\Psi^*(x_1,x_2,\ldots,x_n)$ at the various grid points are also orthogonal to each other. Therefore, the (2^n-1)-coefficient problem at each grid point is converted into a one-coefficient problem per grid point with complexity measures as those discussed above. The remaining problem is the *a priori* estimation of the constant coefficients in vector W, which determine the intensity of the multivariable receptive field for function $\Psi^*(x_1,x_2,\ldots,x_n)$.

The following general approaches may be used for the *a priori* estimation of vector W:

(a) *A priori* clustering and estimation of multivariable covariances of the training data within each receptive field. This is similar to the analysis used for the formulation of ellipsoidal basis functions in multivariable RBFNs (Venkatasubramaniam and Kavuri, 1991).

(b) Directional analysis of the training errors. This involves exploiting the property that each of the (2^n-1) wavelets has a different directional component.

(c) Using nonseparable wavelets. The multidimensional wavelets described above and earlier are separable, that is, they are constructed by multiplying the one-dimensional functions. Recently, nonseparable wavelets have been developed by Kovacevic and Vetterli (1992). These wavelets require only one nonseparable wavelet per grid point, instead of (2^n-1) for the separable wavelets.

In a follow-up publication we will discuss the relative advantages and disadvantages of these approaches, and give more information on their practical implementation.

Wave-Net optimization with explicit control of local and global errors. The optimal Wave-Net may be designed by removing nodes with small contribution to the approximation of the unknown functions. Using Eqs. 60 and 61, we can

compute the local and global errors of the resulting approximation by

$$e^2_{\text{local,new}} = e^2_{\text{local,old}} + \sum_m \sum_{k=l-q}^{l+q} d^2_{mk} \quad (m,k) \in S_-$$

$$e^2_{\text{global,new}} = e^2_{\text{global,old}} + \sum_m \sum_k d^2_{mk} \quad (m,k) \in S_-$$

where d_{mk} are the wavelet coefficients for the nodes removed. Thus, it is easy to keep track of global and local errors of approximation as nodes with small coefficients are removed from the Wave-Net. Nodes are removed till the performance requirements are satisfied. The benefits of using orthonormal basis functions are reaped again in this step of the training procedure. The orthonormality allows us to remove nodes from the network, and monitor its performance explicitly, without retraining. If nonorthonormal basis functions were used, the network weights would require modification at each stage. The Wave-Net obtained at the end of this stage satisfies the performance requirements, and has the smallest number of training nodes for the given training and testing data sets.

Network adaptation. The network may need adaptation to new data. It may be necessary to modify the weights and/or change the structure of the net. The presence of a new data point influences nodes at several levels of scale. This includes any ψ- and ϕ-nodes which have the training data within their own receptive fields in both the input and frequency space (see Figure 10). The new coefficients are calculated by taking the pseudo-inverse according to Eq. 57. If the new error rates fall within the desired bounds, then no further change is required in the Wave-Net. The new error rates may be easily calculated by

$$e^2_{\text{new}} = e^2_{\text{old}} - \sum_m \sum_{k \in S_u} d^2_{mk,\text{old}} + \sum_m \sum_{k \in S_u} d^2_{mk,\text{new}}$$

where, S_u is the set of nodes whose weights were modified.

If the error criteria are still not satisfied, then the network structure needs modification, and new nodes are added. The new nodes must lie on the grid and must have the new data within its input and frequency space. The new data are overfitted and the network optimized by removing nodes with the smallest coefficients without violating the error requirements.

If the new data indicates that a smaller sampling rate than that used for constructing the grid is necessary, and if nodes at the lowest level are used for the approximation, then a new level may have to be added to the grid. New levels may be easily added till the grid spacing is less than or equal to the new sampling rate. Computationally, the most expensive step during adaptation is computing a pseudo-inverse. Therefore, adaptation also has a complexity between $O(N)$ and $O(N^2)$.

Examples

The properties and performance of Wave-Nets are illustrated by two examples described below. The Wave-Net algorithms have been implemented on a Macintosh II using MATLAB.

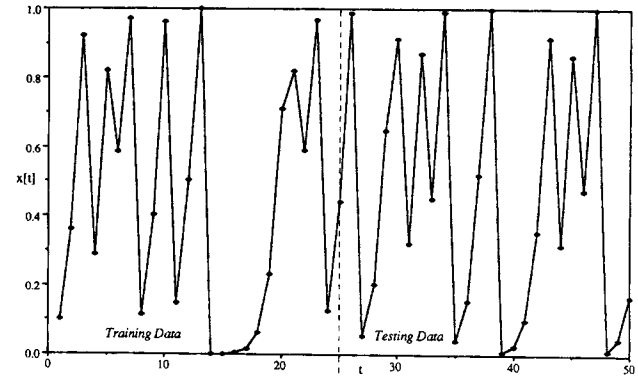

Figure 11. Chaotic time series, $x[t]$ vs. t.

Prediction of a chaotic time series

Consider the problem of predicting the evolution of a logistic map given by,

$$x[t+1] = \alpha x[t](1 - x[t])$$

where α is a parameter. This problem has been considered by Lapedes and Farber (1987) using BPNs; Moody and Darken (1989) and Stokbro et al. (1990) using RBFNs; Ydstie (1990) using BPNs, Heaviside functions (these form an orthonormal basis, but are global), and saturation functions. This system is chaotic for some values of $\alpha > 3$. We consider $\alpha = 4$, for which this system is known to be chaotic. This equation represents simple population dynamics of biological systems. The population at a future time, $x[t+1]$ is proportional to the current population $x[t]$, and the amount of food available currently, $\alpha(1 - x[t])$. Therefore, this equation is relevant to biochemical system dynamics.

The variation of $x[t]$ with t for $x[0] = 0.1$ is shown in Figure 11. We wish to predict the value of $x[t+1]$ given $x[t]$. We use the first 25 points from Figure 11 as training data, and test the network prediction capability for the next 25 points in the figure. A plot of $x[t+1]$ vs. $x[t]$ for the training data is shown in Figure 12. The Wave-Net is trained hierarchically according to the procedure given in the previous section. The number of points in the grid at level zero is 1,353, which makes the highest scaling level to be $L = \inf(\log_2 1353) = 10$. The Wave-Net construction is initiated by selecting two ϕ-nodes at $m = 10$, positioned on the appropriate points on the grid. The construction of the Wave-Net and the evolution of the global approximation

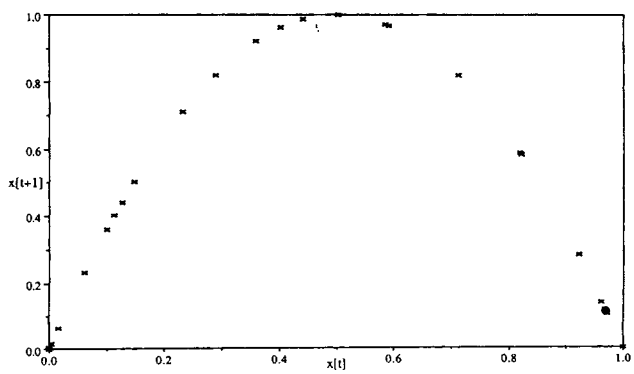

Figure 12. $x[t+1]$ vs. $x[t]$ for training data.

Table 1. Global Approximation and Prediction Errors for Modeling of Chaotic Time Series

No.	Scale	New Nodes	Approx. Error	Pred. Error
1	10	2 ϕ-nodes	1.4008	1.3101
2	10	2 ψ-nodes	0.2658	0.2742
3	9	3 ψ-nodes	0.1312	0.1792
4	8	4 ψ-nodes	0.0675	0.0856
5	7	7 ψ-nodes	0.056	0.1074
6	6	6 ψ-nodes	0.0436	0.1899
7	5	7 ψ-nodes	0.0263	0.201

and prediction errors are shown in Table 1. The corresponding evolution of the local approximation and prediction errors is shown in Figure 13. The approximation at $m = 10$ is the global least-squares fit to the training data. The network construction is continued by adding ψ-nodes that have training data present in their receptive fields in the input and frequency domains. We add 2 ψ-nodes at $m = 10$ to the Wave-Net and train them to approximate the modeling error from the ϕ-nodes. Three ψ-nodes are added at $m = 9$ to approximate the error from the network consisting of two ϕ, and two ψ-nodes. This process of adding ψ-nodes is continued at lower scales, till the training data are overfitted. The network is then optimized by cross-validation with a set of testing data consisting of the last 25 points in Figure 11. The local and global errors of approximation are monitored as ψ-nodes with small weights removed till the prediction error goes through a minimum, as shown in Table 1. Prediction of the final, optimized Wave-Net is shown in Figure 14. Error bounds on the approximation vary over the range of input values, as can be seen in Figure 15. The error bounds for other Wave-Nets are not shown since they are too close to the actual trend.

The construction of the final approximation from scaling functions and wavelets at multiple resolutions is depicted in Figure 16. The activation functions introduced at different scales, multiplied by the corresponding coefficient, are shown in column (a), with their sum in column (b). The cumulative sum of the graphs in column (b) gives the approximation at each scale, which is shown in column (c). The input values of the activation functions extend beyond the range of the training data, as shown in column (a). Portions of the activation functions in the range [0, 1] are used for the approximation as shown by the shaded region in column (a). Portions beyond the input range provide extrapolation over a limited range. The sum of the scaling functions at $m = 10$ gives the least-squares fit to the training data at the coarsest resolution. The wavelets at $m = 10$ approximate the error of approximation at this scale to give a more accurate approximation shown under column (c). Smaller wavelets are selected to further reduce the error of approximation, till the final approximation shown in column (c) for $m = 8$ is obtained. As expected, the size and extent of the wavelets decreases with the scale. Also, only three wavelets are selected at $m = 8$, instead of five, since the wavelets at positions 0.19 and 0.57 have very small weights and are eliminated during network optimization.

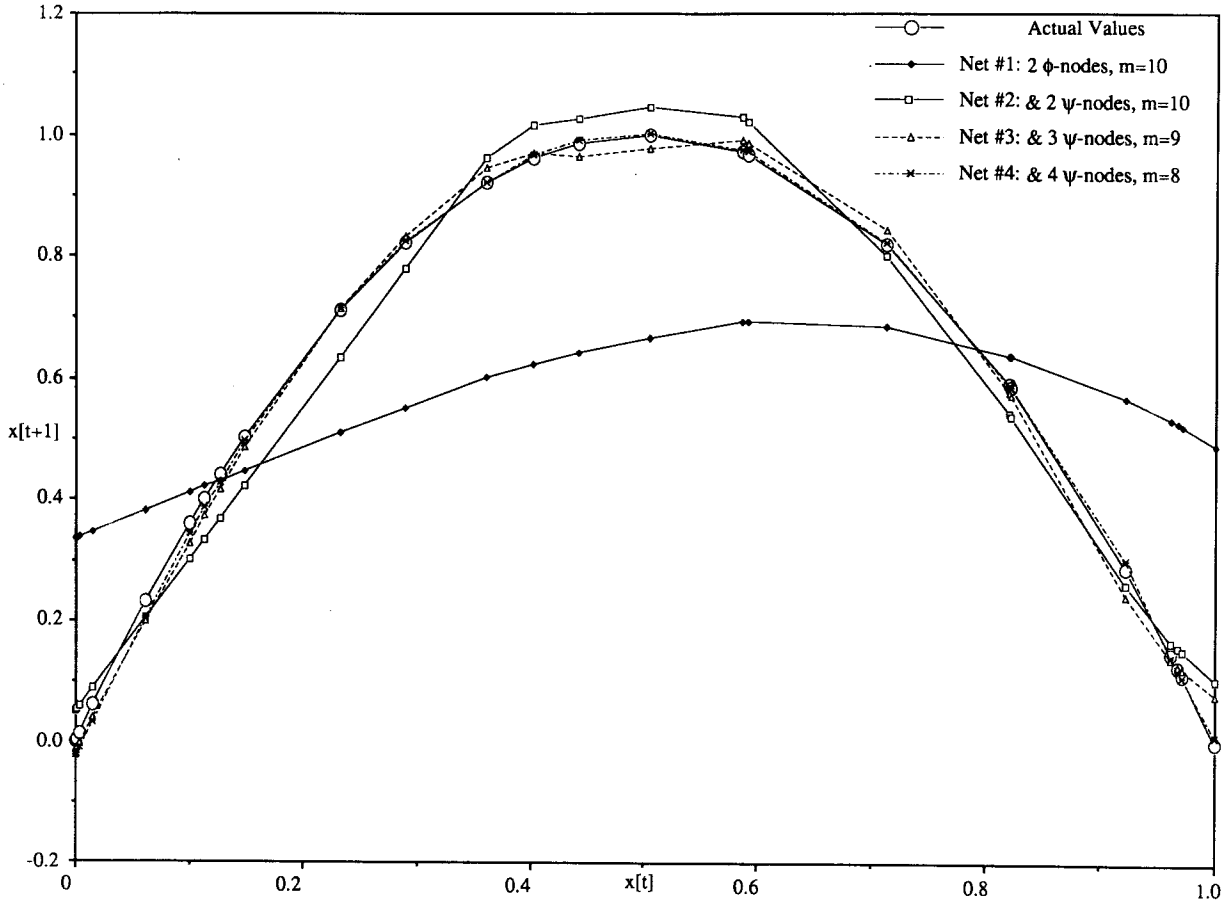

Figure 13. Evolution of Wave-Net approximation for networks in Table 1.

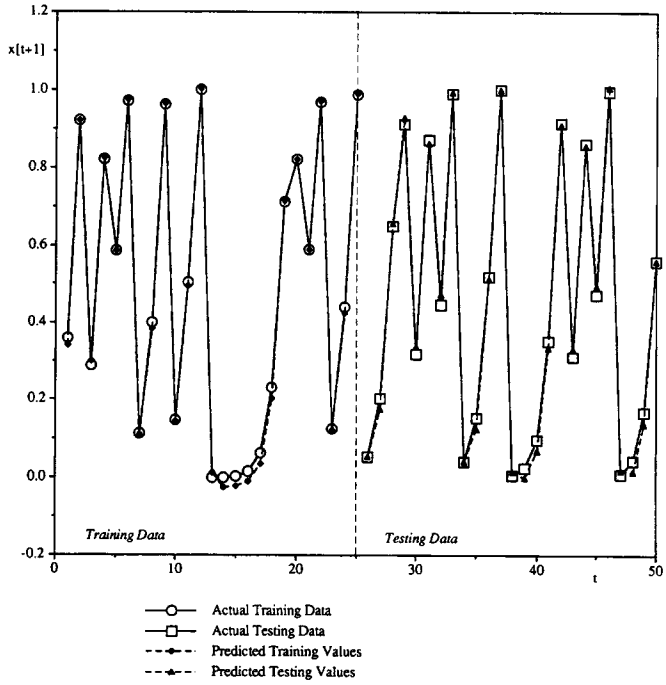

Figure 14. Predictions of optimized Wave-Net for training and testing data.

2 ϕ-nodes at $m = 10$, 2 ψ-nodes at $m = 10$, 3 ψ-nodes at $m = 9$, 4 ψ-nodes at $m = 8$.

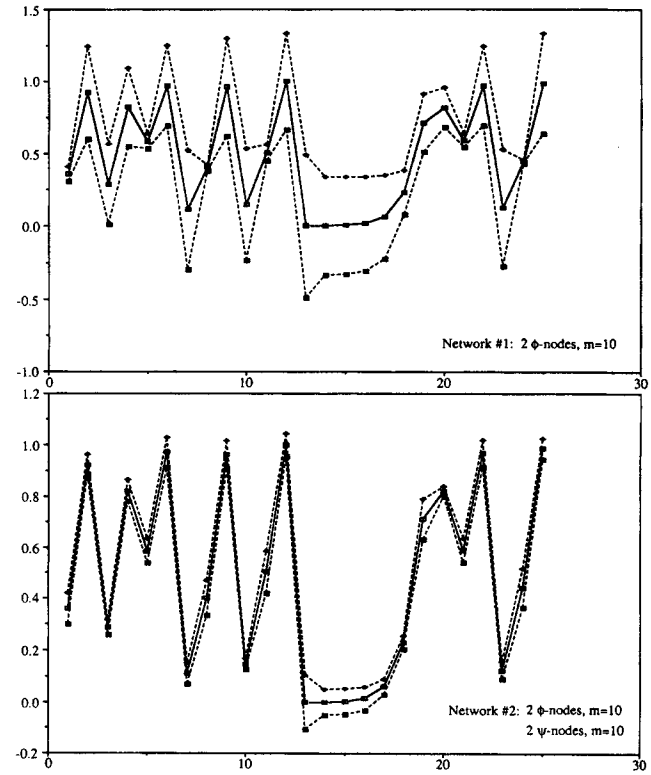

Figure 15. Error bounds on Wave-Net approximations. (See Table 1).

Fault diagnosis classification example

This example was introduced by Kramer and Leonard (1990). It is a simplified version of many static fault diagnosis problems. There are two measured variables, x_1 and x_2, and three classes of faults. The measurements are corrupted by Gaussian noise. A detailed description may be found in the original reference. This example has been used by Leonard and Kramer (1991) to demonstrate the superiority of RBFNs over BPNs, and by Holcomb and Morari (1991), Leonard et al. (1991) using improved versions of RBFNs.

We use a Wave-Net to model the set of data presented in Leonard et al. (1991). 90 data points, 30 in each class, are available. 20 points from each class were used for training, and the rest for testing. The training and testing data are shown in Figure 17. The output is considered to have a value of 1 ± 0.1, 2 ± 0.1, or 3 ± 0.1 depending on the class. The quantization of the output values into three ranges implies that the use of the Battle-Lemarie wavelets with continuity up to the third-order derivatives is an "overkill." Therefore, we chose to engage the two-dimensional Haar wavelet in order to explore its utility in mapping discriminant functions when the output takes on a small number of discrete values. The two-dimensional Haar wavelets and scaling functions are shown in Figure 6. Notice that the one-dimensional Haar scaling function is symmetric about 0.5, and the corresponding wavelet is anti-symmetric about 0.5. Both functions are compactly supported, and their receptive fields do not overlap when placed on the grid points. The two-dimensional Haar wavelet has rectangular receptive fields, therefore, the input space is approximated by several rectangles.

The smallest sampling rate for x_1 is 0.0004, and for x_2 is 0.0001. This gives a 2×5 grid at the lowest common resolution of $L = 12$. The receptive fields of the 10 nodes at $L = 12$ are shown in Figure 18a. Only nodes 2, 3, 5, 7, 9, and 10 have training data present within their receptive fields. Therefore, only these six ϕ-nodes are trained to approximate the training data. The weight of each ϕ-node is shown in Figure 18a, and the corresponding output in Figure 19a. The output values shown in Figure 19 are equivalent to the height of each rectangle, perpendicular to the input space. Nodes 2 and 10 have data belonging only to a single class in their receptive fields, and their coefficients can be adjusted to approximate this data accurately. This is indicated by the shaded regions in Figure 19a. Therefore, no new nodes at higher resolutions are required in this region, and the training is continued for the rest of the input space. The evolution of the error rates for the various Wave-Nets is shown in Table 2.

The classification error is now approximated by adding ψ-nodes at $m = 12$, located at the grid points. The receptive fields of these nodes are equal to those of the ϕ-nodes, since the scale is unchanged. Since this is a two-dimensional problem, we have three ψ-nodes at each grid point, with different orientations, as depicted in Figure 6. The 12 ψ-nodes at $m = 12$ are trained to approximate the error. The weights of the ψ-nodes and their receptive fields are shown in Figure 18b. The discriminant plane at this stage is portrayed in Figure 19b. The use of three wavelets with different orientations provides significant flexibility in the approximation of the data, since their weights can be adjusted independently, such that the four quadrants of the wavelet's receptive field have different output values.

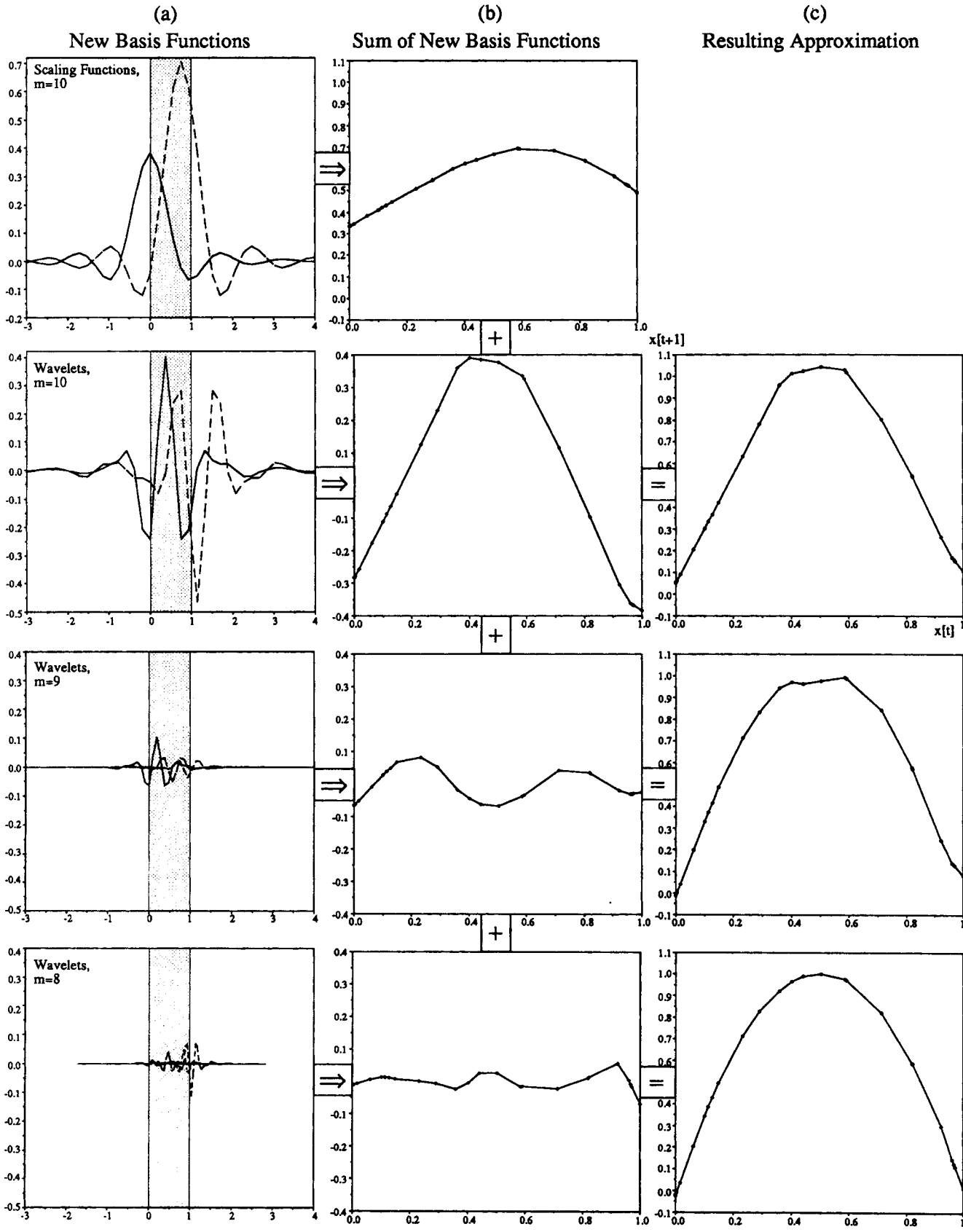

Figure 16. Construction of approximation at multiple resolutions.

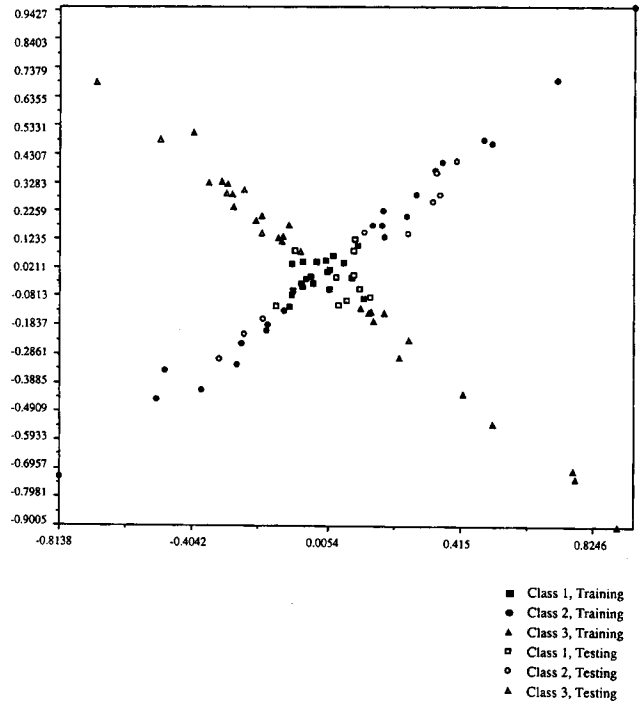

Figure 17. Training and testing data for fault diagnosis example.

Table 2. Wave-Net Performance on Training and Testing Data for Various Networks

No.	Wave-Net Structure	Approx. Error	Pred. Error
1	6 ϕ-nodes at $m=12$ 12 ψ-nodes at $m=12$ 13 ψ-nodes at $m=11$ 21 ψ-nodes at $m=10$	0/60 (0%)	4/30 (13%)
2	6 ϕ-nodes at $m=12$ 12 ψ-nodes at $m=12$ 13 ψ-nodes at $m=11$ 9 ψ-nodes at $m=10$	0/60 (0%)	4/30 (13%)
⋮	⋮	⋮	⋮
3	6 ϕ-nodes at $m=12$ 11 ψ-nodes at $m=12$ 7 ψ-nodes at $m=11$ 9 ψ-nodes at $m=10$	0/60 (0%)	4/30 (13%)
4	6 ϕ-nodes at $m=12$ 11 ψ-nodes at $m=12$ 5 ψ-nodes at $m=11$ 9 ψ-nodes at $m=10$	5/60 (8%)	7/30 (23%)

The training procedure is continued till the data are overfitted. The receptive fields of the nodes at $m=11$ and $m=10$, and their weights are shown in Figures 18c and 18d. From Table 2 we see that perfect classification is possible with 6 ϕ-nodes and 46 ψ-nodes. The Wave-Net is then optimized by crossvalidation with the testing data, by removing nodes with

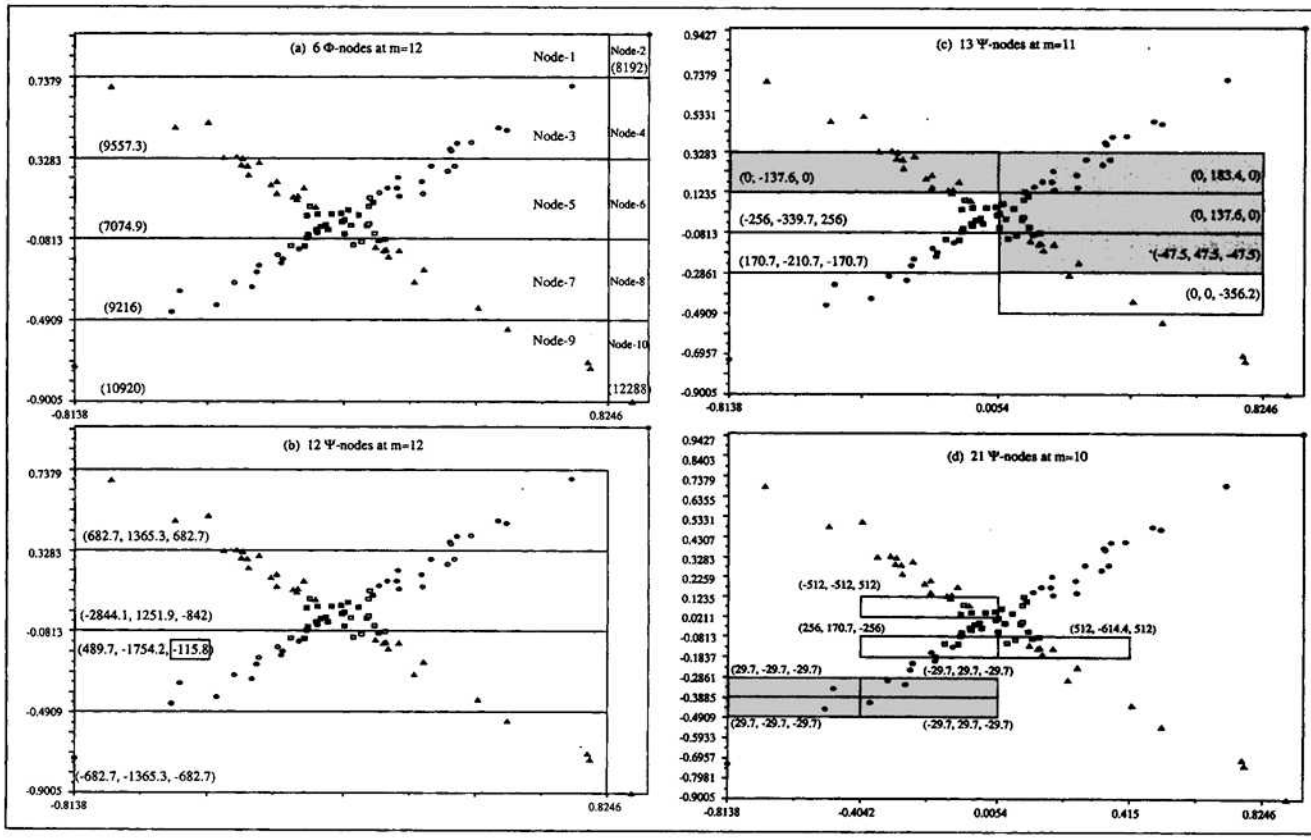

Figure 18. Receptive fields and weights at various resolutions.

Nodes eliminated during optimization are shaded. Numbers denote coefficients for Ψ^1, Ψ^2, Ψ^3.

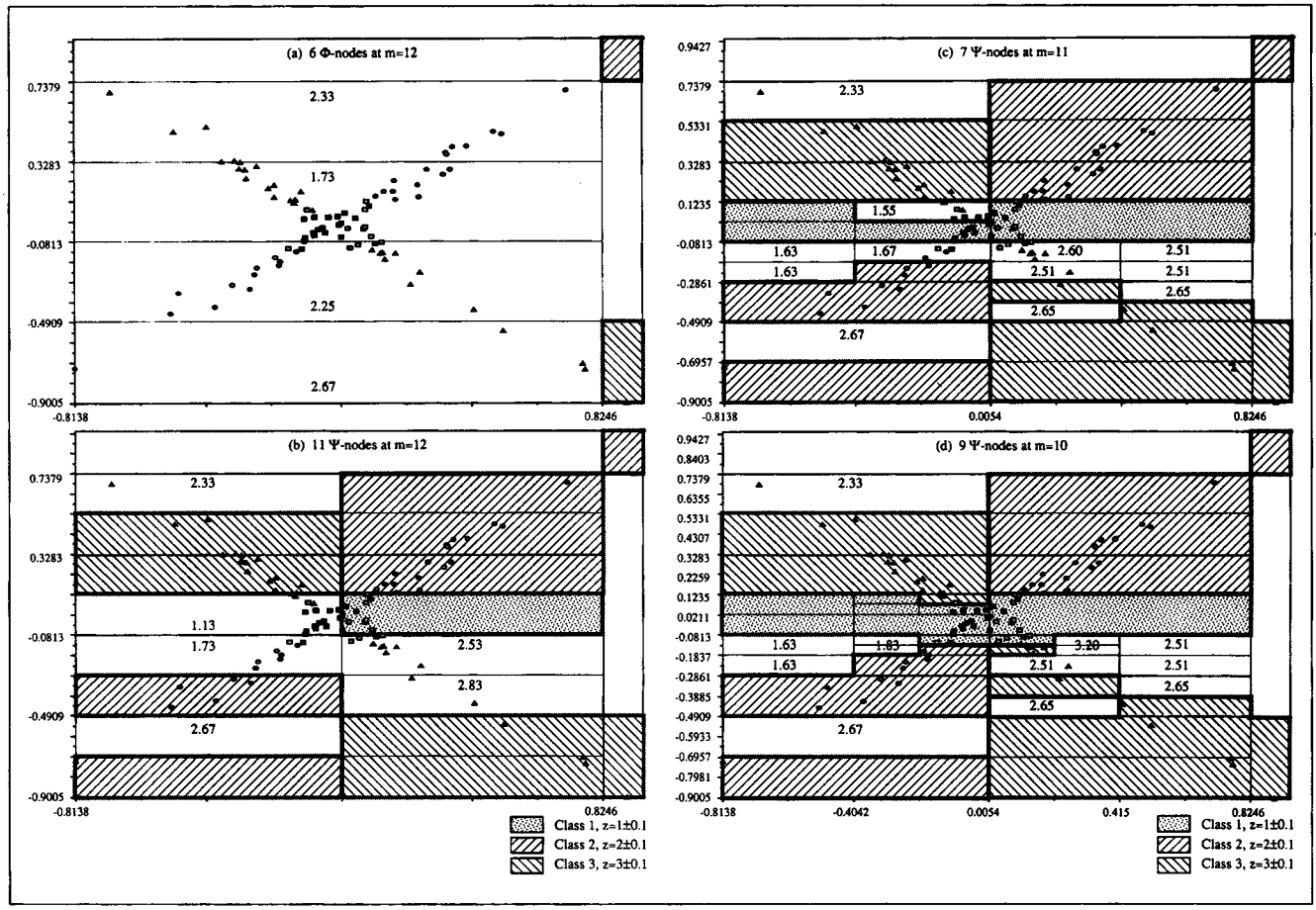

Figure 19. Evolution of discriminant surface at various resolutions.

Numbers indicate Wave-Net prediction in respective rectangular receptive fields for optimized network.

small weights and monitoring the error. The receptive fields of the nodes eliminated during Wave-Net optimization are shown shaded in Figure 18. The performance on testing data starts deteriorating for less than 33 nodes. The optimum network based on the given training data consists of 6 ϕ-nodes at $m=12$, 11 ψ-nodes at $m=12$, 7 ψ-nodes at $m=11$, and 9 ψ-nodes at $m=10$.

The evolution of the discriminant surface separating the three classes is shown in Figure 19. The Wave-Net output for only ϕ-nodes at $m=12$ is shown in Figure 19a. As mentioned above, perfect classification is possible in two regions where data belonging to only one class are present. The output for the other regions is an average of the training data within the respective receptive field. The addition of ψ-nodes divides the input space into smaller sections resulting in improved approximation as shown in Figures 19b, c and d. The discriminant surface for the final, optimized Wave-Net, classifies all the training data accurately, shown in Figure 19d. Misclassified testing data are present either in a region with inadequate training data, or on the boundary of the discriminant surfaces. From this set of figures, we see the advantages of learning at multiple resolutions. The resolution of the nodes in various regions of the input space is decided entirely by the intricacy of the function being approximated, as represented by the training data. Consequently, higher resolution wavelets are required to approximate the discriminant in regions where the classes overlap.

In this example, the dimensions of the rectangular receptive fields are chosen proportional to the smallest sampling rates along each input dimension ($=0.0004/0.0001$). Clearly this is very conservative, and the resolution of the activation functions does not need to get any finer than $m=10$. Other criteria, like the covariance matrix of the training data, may be used to select the dimensions of the receptive fields. Also, Wave-Nets with nodes having elliptical (instead of rectangular) receptive fields may be obtained by using other wavelets like Battle-Lemarie, as mentioned in the fifth section.

Conclusions

In this article we have presented a new type of artificial neural network for learning from empirical data. Wavelet Networks or Wave-Nets perform localized learning in a multiresolution, hierarchical manner, using orthonormal wavelets and scaling functions as activation functions. The network learns the mapping hierarchically by first learning a global approximation, and subsequently reducing the global and local approximation errors by adding nodes at different resolutions to capture finer details. Wave-Nets are designed based on firm

theoretical foundations derived from functional analysis and wavelet theory, and have several attractive properties.

- Wave-Net design is based on an explicit control of the local and global measures of accuracy of the approximation desired. The availability of an error estimate is very useful as a meaningful criterion to decide the network structure and parameters, and eliminates trial and error.
- The use of an orthonormal set of basis functions for approximation guarantees the minimization of the least-squares error of approximation. Also, nodes may be added or removed without retraining the network.
- The overall learning task and network adaptation are very efficient and take from $O(N)$ to $O(N^2)$ time. This is at least an order of magnitude improvement over the complexity of BPNs and RBFNs.
- Different types of wavelets and scaling functions may be used depending on the characteristics of the learning problem.

Wave-Nets solve the learning task as a functional approximation problem, but they possess all the characteristics of artificial neural networks. According to the developers of parallel distributed processing (popularly known as artificial neural networks) (Rumelhart et al., 1986), there are eight major aspects of a parallel distributed processing model:

- A set of processing units.
- A state of activation.
- An output function for each unit.
- A pattern of connectivity among units.
- A propagation rule for propagating patterns of activities through the network of connectivities.
- An activation rule for combining the inputs impinging on a unit with the current state of that unit to produce a new level of activation for the unit.
- A learning rule whereby patterns of connectivity are modified by experience.
- An environment within which the system must operate.

It is clear from the description in this article that a Wave-Net possesses all these characteristics, and is therefore an artificial neural network. It also provides insight into the link between the mystique and arbitrariness of neural network learning and the clarity and precision of functional approximation theory, thus eliminating the black box nature of neural net learning.

Several extensions and applications of Wave-Nets are possible. In this article, we have restricted ourselves only to orthonormal wavelets as activation functions, but several nonorthonormal wavelets are also available. Many functions with well known approximation properties, like cubic splines and Gaussians may be used as scaling functions, with the corresponding wavelets being their first or second-order derivatives. Even some advantages of orthonormality may be retained if certain mathematical conditions based on frame theory are satisfied (Daubechies, 1990). Many wavelets also either belong to the class of Radial Basis Functions or can easily construct these functions. The application of these wavelets for neural learning can provide the link between RBFs and wavelets. Similar connections between wavelets and BPNs have been demonstrated by Pati and Krishnaprasad (1991). A general theory of neural learning may be developed based on Wave-Nets, and is an area of active research. The examples given in this article are simple and are meant to illustrate the properties and design methodology of Wave-Nets. Future work includes the application of Wave-Nets to large, multidimensional problems, with emphasis on adaptive control and system identification.

Acknowledgments

We thank Jim Leonard for providing the data for the fault diagnosis example. Also, financial support from the Leaders for Manufacturing program at MIT is gratefully acknowledged.

Literature Cited

Adomaitis, R. A., R. M. Farber, J. L. Hudson, I. G. Kevrekidis, M. Kube, and A. S. Lapedes, "Application of Neural Nets to System Identification and Bifurcation Analysis of Real World Experimental Data," Los Alamos National Laboratory Technical Report, LA-UR-90-515 (1990).

Albus, J. S., "A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC)," *J. Dyn. Sys. Meas. Contr.*, **97**, 220 (1975).

Battle, G., "A Block Spin Construction of Ondelettes. Part 1: Lemarie Functions," *Commun. Math. Phys.*, **110**, 601 (1987).

Bhat, N. V., P. A. Minderman, T. McAvoy, and N. S. Wang, "Modeling Chemical Process Systems via Neural Computation," *IEEE Control Systems Magazine*, 24 (April 1990).

Broomhead, D. S., and D. Lowe, "Multivariable Functional Interpolation and Adaptive Networks," *Complex Systems*, **2**, 321 (1988).

Chou, K. C., "A Stochastic Modeling Approach to Multiscale Signal Processing," PhD Dissertation, *Technical Report*, LIDS-TH-2036, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA (1991).

Cybenko, G., "Approximation by Superpositions of a Sigmoidal Function," *Math. Control Signals Systems*, **2**, 303 (1989).

Daubechies, I., "Orthonormal Bases of Compactly Supported Wavelets," *Comm. Pure Applied Math.*, **XLI**, 909 (1988).

Daubechies, I., "The Wavelet Transform, Time-Frequency Localization and Signal Analysis," *IEEE Trans. Inform. Theory*, **36**, 5, 961 (1990).

Esteban, D., and C. Galand, "Applications of Quadrature Mirror Filters to Split Band Voice Coding Schemes," *Proc. Int. Conf. Acoust., Speech, Signal Processing* (May 1977).

Girosi, F., and T. Poggio, "Networks and the Best Approximation Property," *MIT AI Memo*, No. 1164 (October 1989).

Haesloop, D., and B. R. Holt, "A Neural Network Structure for System Identification," *Proc. Amer. Control Conf.*, p. 2460 (1990).

Hartman, E. J., J. D. Keeler, and J. M. Kowalski, "Layered Neural Networks with Gaussian Hidden Units as Universal Approximations," *Neural Computation*, **2**, 210 (1990).

Hernandez, E., and Y. Arkun, "Neural Network Modeling and an Extended DMC Algorithm to Control Nonlinear Systems," *Proc. Amer. Control Conf.*, p. 2454 (1990).

Hinton, G. E., "Connectionist Learning Procedures," in *Machine Learning: Paradigms and Methods*, J. Carbonell, ed., The MIT Press, Cambridge, MA (1990).

Holcomb, T., and M. Morari, "Local Training for Radial Basis Function Networks: Towards Solving the Hidden Unit Problem," *American Control Conference*, Boston, p. 2331 (1991).

Hoskins, J. C., and D. M. Himmelblau, "Artificial Neural Network Models of Knowledge Representation in Chemical Engineering," *Comp. Chem. Eng.*, **12**, 881 (1988).

Kovacevic, J., and M. Vetterli, "Non-Separable Multi-dimensional Perfect Reconstruction Filter Banks and Wavelet Bases for R^n," *IEEE Trans. Inform. Theory*, **38**, 2, 533 (1992).

Kramer, M. A., and J. A. Leonard, "Diagnosis Using Backpropagation Neural Networks—Analysis and Criticism," *Comp. Chem. Eng.*, **14**, 12, 1323 (1990).

Kreinovich, V. Ya., "Arbitrary Nonlinearity is Sufficient to Represent All Functions by Neural Networks: A Theorem," *Neural Networks*, **4**, 381 (1991).

Lapedes, A. S., and R. Farber, "Nonlinear Signal Processing Using Neural Networks: Prediction and System Modeling," *Technical Report, Los Alamos National Laboratory*, Los Alamos, New Mexico (1987).

Lee, S., and R. M. Kil, "A Gaussian Potential Function Network With Hierarchically Self-Organizing Learning," *Neural Networks*, **4**, 207 (1991).
Leonard, J. A., and M. A. Kramer, "Radial Basis Function Networks for Classifying Process Faults," *IEEE Control Sys.*, 31 (April, 1991).
Leonard, J. A., M. A. Kramer, and L. H. Ungar, "A Neural Network Architecture that Computes Its Own Reliability," *Comp. Chem. Eng.*, submitted (1991).
Levin, E., "Modeling Time Varying Systems Using a Hidden Control Neural Network Architecture," *Proc. Sixth Yale Workshop on Adaptive and Learning Systems*, p. 127 (1990).
Mallat, S. G., "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **11**, 7, p. 674 (1989).
Meyer, Y., "Principe d'Incertitude, Bases Hilbertiennes et Algebres d'Operateurs," *Bourbaki Seminar*, No. 662 (1985-86).
Moody, J., and C. J. Darken, "Fast Learning in Networks of Locally-Tuned Processing Units," *Neural Computation*, **1**, 281 (1989).
Moody, J., and C. Darken, "Learning with Localized Receptive Fields," in *Proc. Connectionist Models Summer School*, D. Touretzky, G. Hinton, and T. Sejnowski, eds., San Mateo, CA, Morgan Kaufman, p. 133 (1988).
Moody, J., "Fast Learning in Multi-Resolution Hierarchies," *Research Report*, Yale University, YALEU/DCS/RR-681 (1989).
Pati, Y. C., and P. S. Krishnaprasad, "Discrete Affine Wavelet Transforms for Analysis and Synthesis of Feedforward Neural Networks," in *Advances in Neural Information Processing Systems*, Vol. 3, R. P. Lippmann, J. C. Moody, and D. S. Touretzky, eds., Morgan Kaufman, p. 743 (1991).
Poggio, T., and F. Girosi, "A Theory of Networks for Approximation and Learning," *AI Lab Memo*, no. 1140, MIT AI Lab (July 1989).
Poggio, T., and F. Girosi, "HyperBF: A Powerful Approximation Technique for Learning," in *Artificial Intelligence at MIT*, P. H. Winston and S. A. Shellard, eds., MIT Press, Cambridge, MA (1990).
Rengaswamy, R., and V. Venkatasubramaniam, "Extraction of Qualitative Trends from Noisy Process Data Using Neural Networks," *AIChE Annual Meeting*, Los Angeles (1991).
Rice, J. R., *The Approximation of Functions*, Volume 1, Addison-Wesley Pub. Co. (1964).
Rumelhart, D. E., and J. L. McCleland, et al., *Parallel Distributed Processing*, Volume 1, The MIT Press, Cambridge, MA (1986).
Stinchcombe, M., and H. White, "Universal Approximation Using Feedforward Networks with Non-Sigmoid Hidden Layer Activation Functions," in *Proceedings of the International Joint Conference on Neural Networks*, Washington, D.C. (1989).
Stokbro, K., D. K. Umberger, and J. A. Hertz, "Exploiting Neurons with Localized Receptive Fields to Learn Chaos," *Complex Systems*, **4**, 603 (1990).
Strang, G., "Wavelets and Dilation Equations: A Brief Introduction," *SIAM Review*, **31**, 4, 614 (1989).
Ungar, L. H., B. A. Powell, and S. N. Kamens, "Adaptive Networks for Fault Diagnois and Process Control," *Comp. Chem. Eng.*, **14**, 561 (1990).
Venkatasubramaniam, V., and S. N. Kavuri, "Improving Fault Classification by Neural Networks Using Ellipsoidal Activation Functions," *AIChE Annual Meeting*, Los Angeles (1991).
Venkatasubramaniam, V., R. Vaidyanathan, and Y. Yamamoto, "Process Fault Detection and Diagnosis Using Neural Networks—I. Steady-State Processes," *Comp. Chem. Eng.*, **14**, 7, 699 (1990).
Watanabe, K., I. Matsuura, M. Abe, M. Kubota, and D. M. Himmelblau, "Incipient Fault Diagnosis of Chemical Processes via Artificial Neural Networks," *AIChE J.*, **35**, 1803 (1989).
Ydstie, B. E., "Forecasting and Control Using Adaptive Connectionist Networks," *Comp. Chem. Eng.*, **14**, 4/5, 583 (1990).

Manuscript received Feb. 25, 1992, and revision received July 23, 1992.