Wavelet Networks
Qinghua Zhang and Albert Benveniste, Fellow, IEEE

Abstract—Based on the wavelet transform theory, the new
notion of wavelet network is proposed as an alternative to feed-
forward neural networks for approximating arbitrary nonlinear
functions. An algorithm of backpropagation type is proposed for
wavelet network training and experimental results are reported.

I. INTRODUCTION

The approximation of general continuous functions by
nonlinear networks such as discussed in [1], [2] is very
useful for system modeling and identification. Such approx-
imation methods can be used, for example, in black-box
identification of nonlinear systems. Recently neural networks
have been established as a general approximation tool for
fitting nonlinear models from input/output data. The work of
G. Cybenko [3], and Caroll and Dickinson [5] established a
universal approximation property for such networks (see also
[6], [7]). On the other hand, the recently introduced wavelet
decomposition [8]–[13] emerges as a new powerful tool for
approximation. Such an approximation turns out to have a
structure very similar to the one achieved by a (1 + 1/2)-layer
neural network. In particular, recent advances have shown the
existence of orthonormal wavelet bases, from which follows the
availability of rates of convergence for approximation by
wavelet-based networks. This paper presents a new type of
network, called a wavelet network, inspired by both the
feedforward neural networks and wavelet decompositions. An
algorithm of backpropagation type is proposed and experimental
results are reported.

The paper is organized as follows. In Section I, network
structures for approximation are discussed, and the wavelet
network is introduced. A learning algorithm for wavelet net-
work training is presented in Section II. Finally, experimental
results are reported in Section III and conclusions are drawn.

The following generic notations will be used throughout this
paper: lower case symbols such as $x, y, a, b, \ldots$ refer to scalar
values, lower case boldface symbols such as $\mathbf{x}, \mathbf{y}, \mathbf{a}, \mathbf{b}, \ldots$
refer to vector valued objects, and finally capital symbols will be
used for matrices.

II. NETWORK STRUCTURES FOR APPROXIMATION

In this section some network structures for approximation
are discussed. First, feedforward neural networks as analyzed
in [3] are briefly reviewed. Then an alternative approach based
on wavelet theory is presented.

1

A. Neural Networks

Fig. 1 depicts a so-called (1 + 1/2)-layer neural network.
Recently, the ability of such neural networks to approximate
continuous functions has been widely studied [3], [5]–[7]. In
particular, the following result has been proved in [3]:

If $\sigma(\cdot)$ is a continuous discriminatory function, then finite
sums of the form

$$g(x) = \sum_{i=1}^{N} \omega_i \sigma(a_i^T x + b_i)$$  (1)

are dense in the space of continuous functions defined on
$[0, 1]^n$, where $\omega_i, b_i \in \mathbb{R}, a_i \in \mathbb{R}^n$. In other words,
given any continuous function $f$ defined on $[0, 1]^n$ and any $\varepsilon > 0$,
there is a sum $g(x)$ of the form (1), for which $|g(x) - f(x)| < \varepsilon$ for
all $x \in [0, 1]^n$.

Any bounded and measurable sigmoidal function $\sigma(\cdot)$ is
discriminatory, as shown in [3] by G. Cybenko. In particular,
you continuous sigmoidal function is discriminatory. Worth
to be mentioned is the work [4] by Barron where tight
approximation bounds for neural networks are obtained.
Finally, the work of Caroll and Dickinson [5] establishes a
link between best approximating neural networks and some
discrete approximation of the Radon transform of the function
in consideration. By the way, explicit bounds of the minimum
approximation error achieved by an optimal neural network of
a given size are obtained. The coefficients of the best
approximating neural network are not explicitly provided by
the Radon transform, however. This will be in contrast to our
approach, and this has several important consequences we shall
discuss later.

In proposing our wavelet networks as an alternative, we have
the following objectives in mind:

- Preserve the "universal approximation" property, i.e.,
  provide a class of networks exhibiting the same density
  property as the class (1).
- Have an explicit link between the network coeffi-
cients and some appropriate transform. This will be
extremely useful for gauging good initial values for the
backpropagation-like algorithm we shall propose. Fur-
thermore, this may also be used for a deeper theoretical
investigation of the properties of this learning algorithm,
but this latter point is deferred to future work.

1 As defined in [3], $\sigma(\cdot)$ is discriminatory if for a Borel measure $\mu$ on
$[0, 1]^n \times [0, 1]$, $\int_{[0, 1]^{n+1}} \sigma(a^T x + b)\mu(da, db) = 0$. For $\mu \in \mathbb{R}$
and $\forall b \in \mathbb{R}$ implies that $\mu = 0$.

2 Defined also in [3], $\sigma(\cdot)$ is sigmoidal if

$$\sigma(t) = \begin{cases} 1 & \text{as } t \to +\infty, \\ 0 & \text{as } t \to -\infty. \end{cases}$$

Manuscript received March 19, 1991; revised March 3, 1992.
Q. Zhang is with Linkping Tekniko Högskola, Institutionen for
Systemteknik, S-581 83 Linkoping, Sweden.
A. Benveniste is with IRISA-IRISA, Campus de Beaulieu, 35 042 Rennes
Cedex, France.

1045-9227/92$03.00 © 1992 IEEE
possibly achieve the same quality of approximation with a network of reduced size.

We shall discuss in our conclusion how far these objectives have been achieved. In the next subsection, the appropriate transform is introduced: the wavelet decomposition.

**B. Wavelet Decompositions as Universal Approximators**

We are interested in finding some appropriate function \( \psi: \mathbb{R}^n \rightarrow \mathbb{R} \) with the following property: there exists a denumerable family of the form

\[
\Phi = \left\{ \det(D)^{1/2} \psi[D_k x - t_k] : t_k \in \mathbb{R}^n, \quad D_k = \text{diag}(d_k), \quad d_k \in \mathbb{R}^n_+, \quad k \in \mathbb{Z} \right\}
\]

where the \( t_k \)'s are arbitrary translation vectors and the \( d_k \)'s are arbitrary dilation vectors specifying the diagonal\( k \) dilation matrices \( D_k \) satisfying the frame property: there exist two constants \( c_{\min} > 0 \) and \( c_{\max} < \infty \) such that, for all \( f \) in \( L^2(\mathbb{R}^n) \), the following inequalities hold:

\[
c_{\min} ||f||^2 \leq \sum_{k \in \mathbb{Z}} | \langle \psi, D_k f \rangle |^2 \leq c_{\max} ||f||^2
\]

In this sum, \( \langle \cdot, \cdot \rangle \) denotes the inner product in \( L^2(\mathbb{R}^n) \) and the sum ranges over all elements of the family \( \Phi \).

It is a consequence of (3) that the family \( \Phi \) is dense in \( L^2(\mathbb{R}^n) \), and the quality of the best approximation for a fixed number of terms in the sum depends on how far from 1 the coefficients \( c_{\min} \) and \( c_{\max} \) are. Hence the collection of all linear combinations of elements of the family \( \Phi \)

\[
g(x) = \sum_{k=1}^{\infty} w_k \phi_k(x), \quad \phi_k \in \Phi,
\]

is dense in \( L^2(\mathbb{R}^n) \).

From this follows also that the collection of all finite sums of the form

\[
g(x) = \sum_{k=1}^{N} w_k \det(D)^{1/2} \psi[D_k x - t_k]
\]

where the \( t_k \)'s are arbitrary translation vectors and the \( d_k \)'s are arbitrary dilation vectors specifying the diagonal dilation matrices \( D_k \) is also dense in \( L^2(\mathbb{R}^n) \), since it contains in particular all finite linear combinations of elements of the frame \( \Phi \). There are obviously more degrees of freedom in class (3) than in class (4) since translations and dilations are not constrained to belong to the denumerable families specified in (2). This additional flexibility can be used to fit these dilations and translations to a particular function, and we shall discuss this point later on.

It remains to exhibit families \( \Phi \) as above: the wavelet theory will be used for this purpose.

1) **Using the Continuous Wavelet Decomposition:** The continuous wavelet decomposition allows us to decompose any function \( f(x) \in L^2(\mathbb{R}^n) \) using a family of functions obtained by dilating and translating a single "wavelet" function \( \psi: \mathbb{R}^n \rightarrow \mathbb{R} \). To build such a wavelet we proceed as follows.

Consider first a scalar wavelet in the Morlet-Grossmann sense [13, 9], i.e., a function \( \psi_k: \mathbb{R}^n \rightarrow \mathbb{R} \) with a Fourier transform \( \hat{\psi}_k(\omega) \) satisfying the condition

\[
C_{\psi_k} = \int_{\mathbb{R}^n} |\hat{\psi}_k(\omega)|^2 d\omega < \infty.
\]

Using this scalar wavelet \( \psi_k \) we build the desired wavelet \( \psi: \mathbb{R}^n \rightarrow \mathbb{R} \) by setting

\[
\psi(x) = \psi_k(x_1) \ldots \psi_k(x_n) \text{ for } x = (x_1, \ldots, x_n)
\]

and we also introduce

\[
C_\psi = \sum_{k=1}^{\infty} C_{\psi_k}.
\]

Then the following continuous wavelet decomposition formulae hold:

\[
f(x) = \frac{1}{C_\psi} \int_{\mathbb{R}^n} \int_{\mathbb{R}^n} W(d, t)(\det D)^{1/2} \psi[D(x - t)] d\lambda d\mu
\]

\[
W(d, t) = \int_{\mathbb{R}^n} f(x)(\det D)^{1/2} \psi[D(x - t)] dx.
\]

where \( D = \text{diag}(d) \).

In these formulae, \( d \) and \( t \) are the dilation and translation vectors, respectively. This result is classical [9] for \( n = 1 \) and easily extends to the multidimensional case. For the sake of completeness, we give a proof for the case \( n > 1 \) in Appendix A. It is also shown in [9] that the denumerable family

\[
\psi_{\alpha, \beta} = \left\{ \alpha^{\beta} \psi_0(\alpha^{\beta} x - \beta) : k, l \in \mathbb{Z} \right\}
\]

constitutes a frame of \( L^2(\mathbb{R}^n) \) (i.e., satisfies (3)) for suitable choices of the parameters \( (\alpha, \beta) \). Commonly used wavelets, such as the "Morlet" wavelet for instance, yield \( c_{\min} \) and \( c_{\max} \) very close to 1 for \( \alpha = 2, \beta = 1 \). We show in Appendix A

...
that we also get a frame of $L^2(\mathbb{R}^n)$ by taking direct products of elements of $\Psi_{\lambda}(\alpha, \beta)$, i.e.,

$$\Psi(\alpha, \beta) = \{ (\det D)^{\frac{1}{2}} \psi(Dz - \beta d) : d, l \in \mathbb{Z}^n \}$$

where $d = (d_1, \ldots, d_n), L = \text{diag}(\alpha^{d_1}, \ldots, \alpha^{d_n})$ (11)

which is of the desired form (2).

Note that, if we take instead $\psi : \mathbb{R}^n \to \mathbb{R}$ such that

$$C_{\psi} = \int_0^{\infty} |\hat{\psi}(h)|^2 \frac{dh}{h}$$

exists and is independent of $\omega$ (an isotropy condition), then the wavelet decomposition formulae and frame property hold with $D$ ranging over scalar matrices of the form $D[I]$ [10], [11].

So isotropic wavelets can be used as well, but we shall not use this additional flexibility.

Finally, there is even no need for our purpose that the synthesis and decomposition formulae (8) and (9), respectively) use the same wavelet $\psi$, different wavelets in duality can be used as well. To summarize, the continuous wavelet transform theory in the Morlet-Grossmann sense provides us with considerable flexibility in designing our networks. In our experiments, we have selected $\psi_0(x) = -\pi^{-\frac{1}{2}} e^{-\frac{1}{2} x^2}$ as a scalar wavelet (it satisfies condition (6)) and in the multidimensional case we have taken direct products of such scalar wavelets. Note that the Laplacian of the Gaussian function $e^{-\frac{1}{2} |x|^2}$ satisfies the isotropic admissibility condition (12), so we might have used it directly.

2) Using Orthogonal Wavelets: On the other hand, families $\Psi(\alpha, \beta)$ satisfying the frame property (3) can be obtained directly from the theory of orthonormal wavelets [8], [12] by taking $\alpha = 2$ and $\beta = 1$ in (11) and selecting an orthonormal mother wavelet $\psi$. In this case, the family $\Psi(\alpha, \beta)$ is furthermore orthonormal.

3) Summary and Discussion: Referring to our discussion about the objectives of the paper, we may state the following at this point:

- The "universal approximation" property is guaranteed for finite sums of the form (5) if $\psi$ is chosen as an appropriate wavelet according to the discussion above.

- Explicit link between the network coefficients and the wavelet transform is provided. To make this clearer for wavelets of the Morlet-Grossmann class, rewrite decomposition (8) in the form

$$f(x) = \frac{1}{C_{\psi}} \int_{\mathbb{R}^n} \int_{\mathbb{R}^n} (\det D)^{\frac{1}{2}} \psi(D(x - t))(\mathcal{W}(d, t))ddt$$

Then an approximation of the form

$$f(x) \approx \sum_{i=1}^{N} w_i (\det D)^{\frac{1}{2}} \psi(D(x - t_i))$$

can be thought of as an approximation of measure $|W(d, t)|ddt$ by some suitable linear combination of Dirac measures.\(^5\) And it turns out that $W$ can be roughly

estimated from data using decomposition formula (9) directly.

- Referring to the two types of approximation (4) and (5), we shall call them the (truncated) wavelet decomposition and wavelet network, respectively. In the wavelet decomposition, only the weights $w_i$ will be identified, while the dilation and translations will follow the regular grid structure. In contrast, in the wavelet network, weights, dilations, and translations will jointly be fitted from data.

We shall provide in the sequel algorithms to learn the parameters of these two types of approximations using noisy input/output data.

C. Wavelet Networks and Their Parametrization

Based on the previous discussion, we propose a network structure of the form

$$g(x) = \sum_{i=1}^{N} w_i \psi[D_i(x - t_i)] + \tilde{g}$$

where the additional (and redundant) parameter $\tilde{g}$ is introduced to help dealing with nonzero mean functions on finite domains. Note that this form (13) is equivalent to the form (5) up to the constant $\tilde{g}$ since the dilations and translations are adjustable.

Furthermore, in order to compensate for the orientation selective nature of the dilations in (13), we combine a rotation with each affine transform to make the network more flexible. Our wavelet network structure is thus of the following form:

$$g(x) = \sum_{i=1}^{N} w_i \psi[D_i R_i(x - t_i)] + \tilde{g}$$

where

- the additional parameter $\tilde{g}$ is introduced in order to make it easier to approximate functions with nonzero average, since the wavelet $\psi(x)$ is zero mean;

- the dilation matrices $D_i$'s are diagonal matrices built from dilation vectors, while $R_i$'s are rotation matrices.

This network structure is illustrated in Fig. 2.

III. LEARNING ALGORITHM

In this section we propose an algorithm for adjusting the parameters of our wavelet network. The learning is based on a sample of random input/output pairs $(x, f(x))$ where $f(x)$ is the function to be approximated. A stochastic gradient type algorithm will be introduced for this purpose, which is very similar to the backpropagation algorithm for neural network learning. More precisely, in the sequel we are given a sequence of random pairs $(x_k, y_k = f(x_k) + v_k)$ where $(v_k)$ is observation noise.

A. Principle of the Stochastic Gradient Algorithm

Collect all the parameters $\tilde{g}, w_i, t_i, s, D_i$'s, and $R_i$'s in a vector $\theta$ and write $g(x)$ to refer to the network defined by (14) with the parameter vector $\theta$. The objective function to be minimized is

$$C(\theta) = \frac{1}{2} \mathbb{E}[\|y(x) - y(x)\|^2].$$

\(^5\) This is the argument used in [5] to establish the link between neural networks and the Radon transform, however this link fails to provide directly approximants in the form of neural networks.
for the sake of clarity):

\[
\begin{bmatrix}
\cos \alpha^1 & -\sin \alpha^1 \\
\sin \alpha^1 & \cos \alpha^1 \\
\end{bmatrix}
\begin{bmatrix}
1 & 0 \\
0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
\cos \alpha^2 & 0 & -\sin \alpha^2 \\
0 & 1 & 0 \\
\sin \alpha^2 & 0 & \cos \alpha^2 \\
\end{bmatrix}
\begin{bmatrix}
1 & 0 \\
0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
1 & 0 & \cos \alpha^m & -\sin \alpha^m \\
0 & 1 & \sin \alpha^m & \cos \alpha^m \\
\end{bmatrix}
\]

where \( m = \frac{1}{2} n(n - 1) \) (19)

Thus Givens rotations must be identified, which is feasible only for problems of low dimension. For problems of larger dimension \((n > 3)\), we process the rotation matrices in a different way.

For this alternative approach, at the beginning of each step of the stochastic gradient procedure, the \( R \) matrices are assumed to be rotation matrices. Then the entries of these rotation matrices are modified in the opposite of the (stochastic) gradient. After such a modification, these matrices are generally no longer rotation matrices. So we project them into the set of rotation matrices by solving the following approximation problem: given a square nonsingular matrix \( \tilde{R} \in \mathbb{R}^{n \times n} \), find

\[
R = \arg \min_{R \in \mathcal{R}} \| R - \tilde{R} \|_F \text{ with } \mathcal{R} = \{ R : R \in \mathbb{R}^{n \times n}, R^T R = I, \text{ det } R = 1 \}.
\]

For this purpose, a symmetric orthonormalization algorithm introduced in [17] is used. This is an iterative procedure which converges very rapidly when the matrix to be orthonormalized is close to an orthonormal one (this happens to be the case for us since only a small modification of the entries is performed at the stochastic gradient stage). In Appendix B, a brief description of this algorithm is provided.

C. Calculating the Stochastic Gradient

Explicit formulae for the partial derivatives of the functional (16) w.r.t. each component of the parameter vector \( \theta \) are now listed. From these formulae the calculation of the gradient follows immediately. For convenience, we shall use the following notations:

\[
\psi(x) = \frac{d\psi(x)}{dx}, \quad e_k = g_k(x_k) - y_k \quad \text{and} \quad x_i = D_i R_i (x_k - t_k).
\]
The partial derivatives of the functional $c(\theta, x_k, y_k)$ w.r.t. $g$, $w_i$, $t$, $s_i$, and $R_i$ are respectively given by

\[
\frac{\partial c}{\partial g} = e_k \\
\frac{\partial c}{\partial w_i} = e_k \psi(x_i) \\
\frac{\partial c}{\partial t_i} = -e_k u_i R_i^T D_i \psi'(x_i) \\
\frac{\partial c}{\partial s_i} = -e_k u_i D_i^T \text{diag}(R_i(x_k - t_i)) \psi'(x_i) \\
\frac{\partial c}{\partial R_i} = e_k u_k D_i \psi'(x_i)(x_k - t_i)^T.
\]

At this point we are ready to implement the stochastic gradient procedure, additional work is required, however. As is the case for the backpropagation algorithm for neural network learning [18,19], the objective function (15) is likely to be highly nonconvex, so local minima are expected. To improve the situation, careful initialization of the algorithm is performed and appropriate constraints are set on the adjusted parameters. While no theoretical investigation is available which may guarantee convergence, drastic improvement was exhibited: the original stochastic gradient usually diverged but our modified algorithm always performed successfully in our experiments. As far as we know, a similar situation is encountered in feedforward neural network training. The corresponding additional features are now presented.

D. Setting Constraints on the Adjustable Parameters

Let $f: \mathcal{D} \rightarrow \mathbb{R}$ be the function to be approximated, where $\mathcal{D} \subset \mathbb{R}^n$ is the domain where the approximation should be made. The following constraints on the parameters are introduced:

1) To keep the wavelets inside or near to the domain $\mathcal{D}$, select another domain $\mathcal{E}$ such that $\mathcal{D} \subset \mathcal{E} \subset \mathbb{R}^n$, and require

\[ t_i \in \mathcal{E}, \quad i = 1, 2, \ldots, N. \tag{20} \]

2) To avoid excessive compression of each wavelet, select $\varepsilon > 0$ and require

\[ D_i^{-\frac{1}{2}} > \varepsilon I, \quad i = 1, 2, \ldots, N. \tag{21} \]

3) To prevent the total volume of the wavelet supports from being too small, select a $V > 0$ and require

\[ \sum_{i=1}^{N} (\det D_i)^{-1} > V. \tag{22} \]

4) The rotation matrices should verify

\[ R_i^T R_i = I \quad \text{and} \quad \det R_i = 1, \quad i = 1, 2, \ldots, N. \tag{23} \]

Note that the wavelets we use are not necessarily compactly supported, but are always rapidly vanishing, so the notion of "support" used in the third constraint should be taken in an approximate sense. A formal definition for the notion of support in such a case was given by Y. C. Pati and P. S. Krishnaprasad in [20]. To satisfy these constraints, a projected stochastic gradient procedure is implemented as depicted in Fig. 3.

E. Network Initialization

This is where the link to wavelet decomposition can be used. More precisely, an initial guess for the network parameters can be derived by using the decomposition formula (9) (recall that in contrast our network structure mimics the synthesis formula (8)). The idea is the following: using noisy input/output measurements $(x, f(x))$, we can get some rough estimate of $W(h, l)$ in (9) by replacing the integration by an approximate averaging of the observations. In this paper, we only propose simple heuristic implementations of this idea.

1) The One-Dimensional Case: For the sake of simplicity, we consider first the one dimensional case. Assume that we want to approximate the function $f(x)$ over the domain $\mathcal{D} = [a, b]$ by a network of the form

\[ g(x) = \sum_{i=1}^{N} \alpha_i \psi \left( \frac{x - t_i}{s_i} \right) + g. \]

Note that no rotation parameter is needed for one dimensional case. The initialization of this wavelet network consists in the evaluation of the parameters $\beta, \alpha_i, t_i, s_i$ for $i = 1, 2, \ldots, N$. To initialize $\beta$ we need to estimate the mean of the function $f(x)$ (from its available observations) and set $\beta$ to this estimated mean. $\alpha_i$’s are simply set to zero. The rest of the problem is how to initialize $t_i$’s and $s_i$’s.

To initialize $t_1$ and $s_1$, select a point $p$ between $a$ and $b$; $a < p < b$. The choice of this point will be detailed later. Then we set

\[ t_1 = p, \quad s_1 = \xi(b - a), \]

where $\xi > 0$ is a properly selected constant (the typical value of $\xi$ is 0.5). The interval $[a, b]$ is divided into two parts by the point $p$. In each sub-interval, we recursively repeat the same procedure which will initialize $t_2, s_2, t_3, s_3$, and so on, until all the wavelets are initialized. This procedure applies in this form when a number of waveforms is used which is a power of 2. When this is not the case, the recursive procedure is applied as long as possible, then the remaining $(t_i)$ are initialized at random for the finest remaining scale.

The following method is proposed to select the point $p$ inside $[a, b]$ (and recursively points which divide all the sub-intervals). Introduce a “density” function

\[ \rho(x) = \frac{g(x)}{\int_a^b g(x) dx}, \quad \text{where} \quad g(x) = \left| \frac{df(x)}{dx} \right|. \]

must be estimated from noisy input/output observations $(x, f(x))$. A very rough estimate of $g(x)$ is used for this
IV. EXPERIMENTAL RESULTS

In this section, results on simulated as well as real data are reported. For the simulations, we use records of randomly generated \( \{x_k\} \)'s that are independent and uniformly distributed on the considered domain \( D \). Then we take \( y_k = f(x_k) + u_k \) where \( (u_k) \) is a simulated white noise independent from the \( x_k \)'s, with standard deviation being approximately 10% of the size of the range of the function \( f \). When real data records \( \{x_k, y_k\} \) are processed, they are not processed on-line, but are read from a file by selecting their index \( k \) at random with uniform distribution in the set \( \{1, \ldots, K\} \) where \( K \) is the number of observations.

To assess the approximation results, a figure of merit is needed. An obvious candidate for this purpose is the empirical estimate of the least squares objective function itself, namely \( \sum_{k=1}^{K} \frac{1}{2} |g_k(x_k) - y_k|^2 \). However, this figure of merit provides a biased assessment by favoring the areas where \( x \)'s are frequently selected; this may for instance be the case for real experiments, where the actual distribution of \( x \) is not chosen. Since we are also interested in assessing the extrapolation ability of our procedure, we preferred to use the following observation figure of merit: a set of test points \( T = \{x_j : j = 1, 2, \ldots, M\} \subset D \) is selected and we assume that \( y_j = f(x_j) + v_j \) is available for \( j = 1, 2, \ldots, M \). Then we take

\[
\delta(y) = \sqrt{\frac{\sum_{j=1}^{M} |g_j(x_j) - y_j|^2}{\sum_{j=1}^{M} (y_j - \bar{y})^2}} \quad \text{with} \quad \bar{y} = \frac{1}{M} \sum_{j=1}^{M} y_j
\]

as a figure of merit.

We have compared our wavelet network, the neural network and the wavelet decomposition (i.e., with fixed dilations and translations) on the same examples. The tested neural network follows Cybenko's structure with the sigmoidal function

\[
s(x) = \frac{1 - e^{-x}}{1 + e^{-x}}.
\]

When learning the corresponding neural network using a back propagation algorithm, we use the parametrization \( \sigma(a^T(x - b)) \) for each neuron instead of \( \sigma(a^T x + b) \), the reason for this is that the partial derivative

\[
\frac{\partial}{\partial a} \sigma(a^T x + b) = \sigma'(a^T x + b) a
\]

depends on the value of \( x \), while in contrast

\[
\frac{\partial}{\partial a} \sigma(a^T(x - b)) = \sigma'(a^T(x - b)) (x - b)
\]

depends on the value of \( (x - b) \). This latter choice makes the backpropagation algorithm more stable. After the learning procedure, we reset \( a = \hat{a} \) and \( b = -\hat{a}^T b \).

The wavelet decomposition (cf. (4)) is estimated from available data with the help of a standard least squares algorithm for minimizing the objective function

\[
\sum_k \left[ f(x_k) - \left( \sum_{i=1}^{N} w_i(x_i) \right) \right]^2
\]

with respect to the weights \( w_i \).
To facilitate the terminology, we use the common term unit to refer to neurons and wavelets in the following. Furthermore, to make comparisons as fair as possible as far as the number of adjusted parameters is concerned, we have adopted the following procedures: for each experiment, the number of units in each type of network is selected in such a way that the resulting numbers of adjusted parameters be as close as possible to each other. Moreover, in any case, the number of adjusted parameters has been always taken to be the least one for the wavelet network. The networks and their learning algorithms have been implemented in C language on a Sun4 Sparc-station.

1) Approximation of a Single Variable Function: Simulation Results: The wavelet function we have taken is the so-called “Gaussian derivative” \( \phi(x) = -ae^{-b|\xi|} \) to which the general approximation theorem described in Section II-B applies. Fig. 4 shows the results for the approximation of the piecewise defined function

\[
f(x) = \begin{cases} 
-1.8 \text{se} - 12.86 & \text{if } -10 \leq x < -2 \\
4.24 \text{se} - 10 & \text{if } -2 \leq x < 0 \\
-\sin(0.03x + 0.7x^3) & \text{if } 0 \leq x \leq 10. 
\end{cases}
\]

over the domain \( D = [-10, 10] \). For comparison we show the results with our wavelet network, with the sigmoid-based neural network, and with the wavelet decomposition. The solid lines represent the function \( f \) and the dashed lines show the approximations. We can see that the wavelet network gives the best approximations for the smooth segment as well as for the sharp segment. The figure of merit \( \delta \) for each approximation is computed on a uniformly sampled test set of 200 points. Comparable results have been found for a number of wavelets ranging from 5 to 10 for our wavelet network.

2) Approximation of Two Variable Functions: Simulations and a Real Experiment: In both cases, we have selected \( \psi(x) = x_1x_2e^{-1|x_1|+|x_2|} \) as our wavelet function.

The first example is a simulation study to approximate the function \( f(x) = (x_1^2 - x_2^2)\sin(0.5x_1) \) over the domain \( D = [-10, 10] \times [-10, 10] \). Table I shows the approximation results using the three methods. The figure of merit \( \delta \) for each method is computed on a uniformly sampled test set of 400 points. Fig. 5 illustrates the form of \( f(x) \) over \( D \) and its approximation realized by the wavelet network.

For another example, real data were taken from a gas turbine compressor. Our task is to identify the function \( \eta = f(\tau, \omega) \) where \( \eta, \tau, \) and \( \omega \) are respectively the thermodynamical efficiency, the output/input pressure ratio and the corrected rotation velocity. The form of this function is illustrated by Fig. 6 in which we can see that the observations were irregularly sampled in several slices between which no observation is available (shown by plane zones in the figure). The networks are trained with 2000 measurement points and the results are tested on 3294 points to compute \( \delta \). Table II shows the approximation results obtained with the three methods. Fig. 6 shows also the resulting approximation for the wavelet network. Note that in the white zone no observation is available, so the result in this zone is not significant. In this experiment, the domain

\[ \text{wavelet decomposition, 31 wavelets, 32 parameters, 10000 learning iterations, } \delta = 0.13286 \]

**Fig. 4.** Approximation results of function (24).

<table>
<thead>
<tr>
<th>TABLE 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPROXIMATION RESULTS OF ( f(x) = (x_1^2 - x_2^2)\sin(0.5x_1) )</td>
</tr>
<tr>
<td>method</td>
</tr>
<tr>
<td>wavelet network</td>
</tr>
<tr>
<td>neural network</td>
</tr>
<tr>
<td>wavelet decomposition</td>
</tr>
</tbody>
</table>

V. CONCLUSION AND COMMENTS

In this paper, a new method has been introduced for identifying general nonlinear static systems from input/output observations. This method was inspired by both neural networks and the wavelet decomposition as introduced by Grossmann-Morlet and further studied by Y. Meyer, I. Daubechies, and many others. The basic idea is to replace the neurons by "wavelets," i.e., computing units obtained by cascading an
1) **Guarantee the “universal approximation” property:** this has been discussed in Section II-B and is a direct byproduct of the wavelet decomposition.

2) **Have an explicit link between the network coefficients and some appropriate transform:** this is also automatically provided by the wavelet decomposition. We have used this in two different ways. First, the intuitive nature of the notions of translation, scale, and rotation, helped us to design the constraints in an appropriate way. Second, the availability of a direct and closed form formula for computing the (continuous) wavelet transform was useful for designing an initial guess of the wavelet network coefficients. Both additional features of our “backpropagation” algorithm helped drastically to improve its behavior. Using the mentioned explicit link in a more fundamental way is under progress and will be reported elsewhere.

3) **Possibly achieve the same quality of approximation with a network of reduced size.** Reported results would favor this claim, but it is wise at this point to be somewhat cautious if the complexity is measured in terms of the number of adjusted parameters, since the neural network procedure we have used for comparison might be improved as well. On the other hand, the wavelet network seems more efficient than the (fitted) wavelet decomposition, although the latter one is likely to be more robust however since it is just a linear regression algorithm. Finally, it is worth noticing that the number of units is generally smaller in wavelet networks than in neural networks, especially for problems of higher dimension. And it turns out that, for the particular wavelet we have chosen, the cost of implementing the nonlinearity (sigmoid versus Gaussian—derivative) is mainly proportional to the number of units.

Our wavelet network has been used for black-box identification of nonlinear static systems. Extension to nonlinear dynamical systems of the form

\[ y_k = g_b(y_{k-1}, y_{k-2}, \ldots, y_{k-n}, u_k) + v_k \]

or in state space form is conceptually straightforward, but several questions arise. First, the requested number of wavelets drastically increases with the model order \( d \). Second, no system theory is available for “dynamical wavelet networks” which is certainly a drawback. Further experiments must be performed in this direction before understanding the range of validity of such a black-box identification method for general nonlinear dynamical systems.
Appendix

From One- to Multidimensional Continuous Wavelets

We first derive formulae (8, 9). These formulae are known for \( n = 1 \), see [9]. They generalize immediately for functions \( f \) of the form

\[
f(x) = f_1(x_1) \times \ldots \times f_n(x_n)
\]

since each component in the integral is handled separately. By linearity, (8, 9) extend to linear combinations of such \( f \)'s, i.e., to a dense subset of \( L^2(\mathbb{R}^n) \). But the map \( f \mapsto (\det D)^{-\frac{1}{2}}W(d, t) \) defined by (9) is a bounded linear operator from \( L^2(\mathbb{R}^n) \) into \( L^2(\mathbb{R}^n \times \mathbb{R}^n, dt \times \det D \, dd) \) (the direct product of scalar affine groups with their right Haar measures). Then the composition of maps \( f \mapsto (\det D)^{-\frac{1}{2}}W(d, t) \) defined by (9) and \( (\det D)^{-\frac{1}{2}}W(d, t) \mapsto g \) defined by (8) yields \( g = f \) for \( f \) in a dense subset of \( L^2(\mathbb{R}^n) \), hence for all \( f \in L^2(\mathbb{R}^n) \).

Consequently formulae (8, 9) extend to \( L^2(\mathbb{R}^n) \).

We prove next that family (11) is a frame. Again this result is known for \( n = 1 \) [9], and we denote by \( \Phi \) the corresponding frame. Denote by \( c_{\min} \) and \( c_{\max} \) the frame constants associated with the scalar wavelet \( \psi \). Taking \( f \) of the form (25), we have on the one hand

\[
\|f\|^2 = \|f_1\|^2 \times \ldots \times \|f_n\|^2
\]

and on the other hand

\[
\sum_{\phi \in \Phi} |\phi, f|^2 = \sum_{\phi_1 \in \Phi_1} \ldots \sum_{\phi_n \in \Phi_n} \prod_{i=1}^n |\phi_i, f_i|^2
\]

\[
= \prod_{i=1}^n \left( \sum_{\phi \in \Phi_i} |\phi, f_i|^2 \right)
\]

Hence frame inequalities (3) hold for such \( f \)'s with the family \( \Phi = \Psi(\alpha, \beta) \) as defined in (10) with the same frame constants \( c_{\min} \) and \( c_{\max} \). Next consider a linear combination

\[
f = \sum_k \lambda_k f_k
\]

where the \( f_k \) take the form (25) and are mutually orthogonal; note that such linear combinations are dense in \( L^2(\mathbb{R}^n) \). For \( f \) of the form (26), we have

\[
\sum_{\phi \in \Phi} |\phi, f|^2 = \sum_k |\lambda_k|^2 \sum_{\phi \in \Phi_k} |\phi, f_k|^2
\]

\[
\|f\|^2 = \sum_k |\lambda_k|^2 \|f_k\|^2
\]

so that inequalities (3) extend to such \( f \)'s. Finally, that \( \Psi(\alpha, \beta) \) is a frame follows by density.

Symmetric Orthogonalization of Matrices

In this appendix we describe the algorithm of symmetric orthogonalization of matrices. For more details and the proofs see [17]. Our purpose is to orthogonalize the matrix

\[ A \in \mathbb{C}^{n \times p} \text{ with } \text{rank}(A) = p \leq n. \]

where \( \mathbb{C} \) denotes the complex numbers. For this we compute \( T = S^{-\frac{1}{2}} \) where \( S = A^* A \), \( A^* \) denotes the adjoint of \( A \), and \( S^{-\frac{1}{2}} \) is the symmetric inverse square root of \( S \). It is easy to see that \( R = AT \) is orthogonal. Furthermore, it is proved in [17] that the matrix \( R \) computed in this way satisfies

\[
\|A - R\| = \min_{Q \in \mathcal{U}} \|A - Q\|
\]

where \( \mathcal{U} \) is the set of all orthogonal matrices of \( \mathbb{C}^{n \times p} \). This result is true for both the Euclidean norm and the Frobenius norm. In this sense we say that \( R \) is the symmetric orthogonalization of \( A \).

Let \( \rho(S) \) be the spectral radius of \( S \); it is shown in [17] that if \( \mu < \sqrt{\rho(S)} \), then the scheme

\[
T_0 = \mu I
\]

\[
T_{m+1} = T_m + \frac{1}{2} T_m(I - T_mST_m)
\]

quadratically converges to \( S^{-\frac{1}{2}} \). Moreover, if \( K(S) < 9 \) where \( K(S) \) is the ratio of the extremal eigenvalues of \( S \), then this scheme is stable w.r.t. rounding errors. This condition can be weakened into \( K(S) < (17 + 6\sqrt{8}) \) if a symmetrization is performed at every step on \( T_m \).

In order to define an initial guess, the spectral radius \( \rho(S) \) of the matrix \( S \) has to be estimated. In fact, the \( \infty \)-norm is used instead of this spectral radius. It is shown that:

\[
\mu = \sqrt{\frac{3}{\|S\|_\infty}} \leq \sqrt{\frac{3}{\rho(S)}}
\]

so the initialization \( T_0 = \mu I \) will ensure the convergence of the algorithm.

The first iteration can be skipped since it is easy to compute the following:

\[
T_1 = \frac{3}{2} I - \frac{1}{2} \rho(S)
\]

However, if the matrix \( \Delta = S - I \) is small (i.e., \( \|\Delta\| < 1 \)), the initial guess can be much improved by taking for \( T_0 \) the Taylor expansion of order \( k \) of \( (I + \Delta)^{-\frac{1}{2}} \)

\[
T_0 = I + \sum_{i=1}^n (-1)^i \binom{-\frac{1}{2}}{i} \Delta^i
\]

After \( m \) iterations the magnitude of the error is given by

\[
T_m - S^{-\frac{1}{2}} = o(\|\Delta\|^m)
\]

Symmetrization of \( T_m \) is performed at every stage to improve the stability when needed. The algorithm is now given:

begin

\( S := A^* A; \)

\( \Delta := S - I; \)

\( \delta := ||\Delta||_{\infty}; \)

if \( (\delta < \epsilon) \) then

nothing to do;

elseif \( (\delta < 1) \) then

end
else
    \( \mu := \sqrt{3/5}; \)
    \( T := (3/2)\mu I - (1/2)\mu^2 S; \)
end;

\( \text{sym} := \text{true}; \)

end.

\( \text{iter} := 0; \)

loop
    \( \delta_0 := \delta; \)
    \( Z := 1 - T \times S \times T; \)
    \( \delta := ||Z||_{\infty}; \)
    if \( \delta < \varepsilon \) then exit of the loop end;
    if \( \delta > \delta_0 \) then divergence end;
    \( \text{iter} := \text{iter} + 1; \)
    \( T := (1/2)I \times (2I + Z); \)
    if (sym) then \( T := (1/2)(T' + T) \) end;
endloop.

\( R = A \times T; \)

end.

REFERENCES


Qinghua Zhang was born in Shenzhen, China, in 1963. He graduated from the University of Sciences and Techniques of China in 1986, and received the diploma of D.E.A. from Université de Lille I, France in 1988. From 1988 to 1991 he studied industrial process monitoring at INRIA, Rennes, and received the degree of "docteur de l'Université de Rennes 1" in 1991. Since 1992 he has been with the Department of Electrical Engineering at Linköping University, Sweden, as a guest researcher. His research interests are failure detection and neural networks.

Albert Benveniste (M’81, SM’89, F’92) was born on May 8, 1940 in Paris, France. In 1971 he graduated from Ecole des Mines de Paris. He performed the Thèse d’Etat in Mathematics, probability theory, in 1975. From 1976 to 1979 he was Assticute Professor of mathematics at the Université de Rennes I. From 1979 to present he has been Directeur de Recherche at INRIA. For his thesis, he worked on probability theory, stochastic processes, and ergodic theory. In parallel, he pursued work on automatic control and signal processing in the area of adaptive systems for time-varying systems, with applications to data communications. In 1980 he began joint work with Michel Bassoville on the subject of change detection in signals and dynamical systems with application to vibration mechanics and fault detection in process control. Since 1987 he has been involved in developing jointly with A. S. Willsbky from MIT, a statistical theory of multiresolution signal and image processing, based on dynamical systems and Gaussian random fields on homogeneous trees. Since 1989 he has been interested in computer science, in the area of real-time languages and systems. Cooperating with P. Lécuyer, he participated to the definition of and theoretical studies on the Signal language.