MULTIWAVELET NEURAL NETWORK PREPROCESSING OF IRREGULARLY SAMPLED DATA

AKIRA MORIMOTO, RYUICHI ASHINO, AND RÉMI VAILLANCOURT

Received August 3, 2005

Abstract. Multiwavelets are briefly reviewed and preprocessing and postprocessing for such wavelets are introduced. Least squares curve fitting of irregularly sampled data is achieved by means of unshifted and shifted multiscaling functions. This preprocessing procedure combined with multiwavelet neural networks for data-adaptive curve fitting is shown to perform well in the case of high resolution. In the case of low resolution it is more accurate than numerical integration and cheaper than matrix inversion.

1. Introduction

Wavelets and multiwavelets are used by scientists and engineers to represent and process empirical data in order to extract important characteristics from the data, to denoise or compress the data, and so on.

Scalar wavelets, also called *uniwavelets*, are functions which localize a given function both in space and frequency. A family of scalar wavelets can be constructed from a scalar function called *wavelet function*. Wavelets pick up the details at various scales. A second scalar function, called *scaling function*, is used to pick up approximations at various scales. The simplest scalar wavelet in $L^2(\mathbb{R})$ is the Haar system, see Meyer [1, Section 3.2], with the indicator function of the interval $[0, 1]$ as a scaling function.

On the other hand, *multiwavelets* consist of several scaling functions and wavelets. It is believed that multiwavelets are ideally suited to multichannel signals like color images which are two-dimensional three-channel signals and stereo audio signals which are one-dimensional two-channel signals. For instance, for a two-channel signal, which consists of a two-vector sequence of bits, $\{x_k\}$, the lowpass and highpass filters are 2×2 matrix functions corresponding to two scaling functions and two wavelets, respectively. Multiscaling functions and multiwavelets can simultaneously have orthogonality, linear phase, symmetry and compact support. Such a situation cannot occur in the scalar case with real scaling functions and real wavelets.

A natural question for the discrete multiwavelet transform is how to construct a vector input from a given scalar sampled data (see [2] or [3]). One of the problems here is the difference in data structures. Converting sampled data to input is called *preprocessing* or *prefiltering*. This question is not so serious for uniwavelets because, in practice, the given sampled data can be used as an input for the discrete uniwavelet transform (for detail, see [4, pp. 232–233] and [5, pp. 43–45].

In this paper, a wavelet neural network is used in the interactive construction of vector input data from irregularly sampled scalar data. The desired vector input data is expanded in terms of the multiscaling functions of a chosen multiwavelet and fit in the least squares

2000 *Mathematics Subject Classification.* Primary: 42C40; Secondary: 94A12, 68U10.
Key words and phrases. preprocessing multiwavelet; neural network; curve fitting; irregular sampling.

sense to the given scalar data by means of the wavelet neural network with back propagation. It is assumed that linear combinations of shifted and dilated multiscaling functions can approximate any function in $L^2(\mathbb{R})$. The more accurate is this step, the better will be the discrete multiwavelet transform. It will be seen that adding a constant shift parameter, θ, to the arguments of the multiscaling functions can avoid structural problems with certain multiwavelets. For this reason, the curve fitting procedure is essential for preprocessing. Neural networks allow data-adaptive curve fitting and adaptability greatly reduces the computation cost for certain problems.

In [6], a preprocessing design is proposed for multiwavelet filtering using neural networks for regularly sampled data. In the present paper, a curve fitting method for irregularly sampled data is proposed and applied to a preprocessing design with multiwavelet neural network for the discrete multiwavelet transform. Other preprocessing and postprocessing designs for multiwavelets can be found in [5, pp. 157–161].

The paper is organized as follows. In section 2, multiwavelets are briefly reviewed. Preprocessing and postprocessing are defined in section 3. Section 4 deals with least squares curve fitting to irregularly sampled data. Section 5 introduces multiwavelet neural networks and present numerical results. Section 6 summarizes our preprocessing and postprocessing designs based on curve fitting. Section 7 is a conclusion.

2. Multiwavelets

Definitions and properties of multiwavelets, filters and filter banks can be found, for instance, in Ashino, Nagase, and Vaillancourt [7], Zheng [8], and in the monograph by Keinert [5]. The following standard wavelet and multiwavelet notation will be used.

Notation 1. The notation is as follows.
(i) Given a function $f \in L^2(\mathbb{R})$, we let $f_{jk}(x)$ denote the scaled and shifted functions

$$f_{jk}(x) = 2^{j/2} f(2^j x - k), \qquad j \in \mathbb{Z},\ k \in \mathbb{Z}. \tag{1}$$

(ii) Given a vector-valued function $F = [f^1, \dots, f^d]^T \in L^2(\mathbb{R})^d$, we let F_{jk} denote the scaled and shifted vector functions

$$F_{jk} = [f^1_{jk}, \dots, f^d_{jk}]^T, \qquad j \in \mathbb{Z},\ k \in \mathbb{Z}. \tag{2}$$

(iii) $D = \{1, \dots, d\}$ for a positive integer d.
(iv) $\mathbb{Z}_+ = \{0, 1, 2, \dots\}$ is the set of natural numbers including zero.
(v) $\langle f, g \rangle = \int_{\mathbb{R}} f(x)\overline{g(x)}\,dx$ is the $L^2(\mathbb{R})$ inner product of f and g.

Definition 1. A vector-valued function $\Psi := [\psi^1, \dots, \psi^d]^T \in L^2(\mathbb{R})^d$ is called a *multiwavelet function* if the system

$$\{\psi^\delta_{jk}\}_{\delta \in D,\, j \in \mathbb{Z},\, k \in \mathbb{Z}} \tag{3}$$

forms an orthonormal basis for $L^2(\mathbb{R})$. In this case, the functions ψ^δ_{jk} are called *multiwavelets* and (3) is called an *orthonormal multiwavelet basis*. The *multiwavelet expansion* of $f \in L^2(\mathbb{R})$ with respect to (3) is

$$f(x) = \sum_{\delta \in D,\, j,k \in \mathbb{Z}} \left\langle f, \psi^\delta_{jk} \right\rangle \psi^\delta_{jk}(x). \tag{4}$$

To construct a multiwavelet function, Ψ, from a multiscaling function, Φ, we generalize to multiwavelets the notion of multiresolution analysis given in Mallat [9] and Meyer [1] for scalar wavelets.

Definition 2. An increasing sequence of closed subspaces $\{V_j\}_{j\in\mathbb{Z}}$ of $L^2(\mathbb{R})$,

$$\cdots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \cdots,$$

is called a *multiwavelet multiresolution analysis* if it satisfies the following four properties:

(i) $\cap_{j\in\mathbb{Z}} V_j = \{0\}$ and $\cup_{j\in\mathbb{Z}} V_j$ is dense in $L^2(\mathbb{R})$.

(ii) $f(x) \in V_j$ if and only if $f(2x) \in V_{j+1}$.

(iii) $f(x) \in V_0$ if and only if $f(x-k) \in V_0$ for every $k \in \mathbb{Z}$.

(iv) There exists a multiscaling function $\Phi := [\varphi^1, \ldots, \varphi^d]^T \in V_0^d$ such that $\{\varphi^\delta(x-k)\}_{\delta\in D, k\in\mathbb{Z}}$ form an orthonormal basis of V_0.

When multiwavelets are constructed from a multiresolution analysis, there exist functions φ^δ, $\delta \in D$, called *scaling functions*, such that the set of functions

$$\{\varphi^\delta_{0,k}\}_{\delta\in D,\, k\in\mathbb{Z}} \bigcup \{\psi^\delta_{jk}\}_{\delta\in D,\, j\in\mathbb{Z}_+,\, k\in\mathbb{Z}} \tag{5}$$

is an orthonormal basis of $L^2(\mathbb{R})$. The multiwavelet expansion of $f \in L^2(\mathbb{R})$ with respect to (5) is

$$f(x) = \sum_{\delta\in D,\, k\in\mathbb{Z}} \left\langle f, \varphi^\delta_{0,k} \right\rangle \varphi^\delta_{0,k}(x) + \sum_{\delta\in D,\, j\in\mathbb{Z}_+,\, k\in\mathbb{Z}} \left\langle f, \psi^\delta_{jk} \right\rangle \psi^\delta_{jk}(x). \tag{6}$$

The coefficients $\left\langle f, \varphi^\delta_{0,k} \right\rangle$ and $\left\langle f, \psi^\delta_{jk} \right\rangle$ are called *multiscaling coefficients* and *multiwavelet coefficients* of f, respectively.

Remark 1. In the n-dimensional case, a multiresolution analysis $\{V_j\}_{j\in\mathbb{Z}}$ of $L^2(\mathbb{R}^n)$ for multiwavelets is defined the same way as in the one-dimensional case, but there are $2^n - 1$ multiwavelet functions which can be parametrized by the set $E := \{0,1\}^n \backslash \{(0,\ldots,0)\}$ as

$$\Psi_\varepsilon := [\psi^1_\varepsilon, \ldots, \psi^d_\varepsilon]^T \in V_1^d, \qquad \varepsilon \in E.$$

A multiresolution analysis $\{V_j\}_{j\in\mathbb{Z}}$ of $L^2(\mathbb{R}^n)$ can be constructed from a given one-dimensional multiresolution analysis by means of the tensor product of multiresolution analyses.

Assume that we have a multiwavelet multiresolution analysis $\{V_j\}_{j\in\mathbb{Z}}$ of $L^2(\mathbb{R})$. Using (ii) of Notation 1, we define the lowpass matrix coefficients

$$H_k := \left\langle \Phi_{0,0}, \Phi^T_{1,k} \right\rangle = \left(\left\langle \varphi^\delta_{0,0}, \varphi^\eta_{1,k} \right\rangle \right)_{(\delta,\eta)\in D\times D} \in \mathbb{C}^{d\times d},$$

and the matrix frequency response, or matrix symbol,

$$M_0(\xi) := \frac{1}{\sqrt{2}} \sum_{k\in\mathbb{Z}} H_k e^{-ik\xi} \in L^2([0,2\pi])^{d\times d}.$$

Then, the *dilation equation* and its Fourier transform are

$$\Phi(x) = 2^{1/2} \sum_{k\in\mathbb{Z}} H_k \Phi(2x-k), \qquad \widehat{\Phi}(\xi) = M_0(\xi/2)\widehat{\Phi}(\xi/2),$$

where $\widehat{\Phi}(\xi) := [\widehat{\varphi}^1(\xi), \ldots, \widehat{\varphi}^d(\xi)]^T \in L^2(\mathbb{R})^d$. It is known that if we choose $M_1(\xi)$ such that

$$M(\xi) := \begin{bmatrix} M_0(\xi) & M_0(\xi+\pi) \\ M_1(\xi) & M_1(\xi+\pi) \end{bmatrix}$$

is a unitary matrix for almost all $\xi \in [0, 2\pi]$, then the multiwavelet function Ψ is given by the *wavelet dilation equation* or its Fourier transform,

$$\Psi(x) = 2^{1/2} \sum_{k\in\mathbb{Z}} G_k \Phi(2x-k), \qquad \widehat{\Psi}(\xi) = M_1(\xi/2)\widehat{\Phi}(\xi/2),$$

where G_k, $k \in \mathbb{Z}$, are the Fourier coefficients of $M_1(\xi)$, that is,

$$M_1(\xi) = \frac{1}{\sqrt{2}} \sum_{k\in\mathbb{Z}} G_k e^{-ik\xi} \in L^2([0,2\pi])^{d\times d}.$$

Thus, the highpass matrix coefficients, G_k, $k \in \mathbb{Z}$, are given by the scalar products

$$G_k := \left\langle \Psi_{0,0}, \Phi_{1,k}^T \right\rangle = \left(\left\langle \psi_{0,0}^\delta, \varphi_{1,k}^\eta \right\rangle \right)_{(\delta,\eta)\in D\times D} \in \mathbb{C}^{d\times d}.$$

The orthogonal projection P_j on V_j can be represented as

$$P_j f = \sum_{k\in\mathbb{Z}} C_{j,k}^T \, \Phi_{j,k}, \qquad C_{j,k} := \int_{\mathbb{R}} f(x) \, \overline{\Phi_{j,k}(x)} \, dx \ \in \mathbb{C}^d,$$

where, explicitly, $C_{j,k} = \left[C_{j,k}^1, \cdots, C_{j,k}^d\right]^T$ and

$$C_{j,k}^\delta := \int_{\mathbb{R}} f(x) \, \overline{\varphi_{j,k}^\delta(x)} \, dx, \qquad \delta \in D.$$

Similarly, the orthogonal projection Q_j on W_j, where W_j denotes the orthogonal complement of V_j in V_{j+1}, can be represented as

$$Q_j f = \sum_{k\in\mathbb{Z}} D_{j,k}^T \, \Psi_{j,k}, \qquad D_{j,k} := \int_{\mathbb{R}} f(x) \, \overline{\Psi_{j,k}(x)} \, dx \ \in \mathbb{C}^d.$$

The terms of the sequences $\left\{C_{j,k} \in \ell^2(\mathbb{Z})^d\right\}_{k\in\mathbb{Z}}$ and $\left\{D_{j,k} \in \ell^2(\mathbb{Z})^d\right\}_{k\in\mathbb{Z}}$ are called the *approximation coefficients* and *detail coefficients*, respectively, at resolution j.

Definition 3 (Discrete multiwavelet transform)**.** The *discrete multiwavelet transform* of f is the mapping

$$\left\{C_{j,k} \in \ell^2(\mathbb{Z})^d\right\}_{k\in\mathbb{Z}} \to \left(\left\{C_{j-1,k} \in \ell^2(\mathbb{Z})^d\right\}_{k\in\mathbb{Z}}, \left\{D_{j-1,k} \in \ell^2(\mathbb{Z})^d\right\}_{k\in\mathbb{Z}} \right), \tag{7}$$

defined by

$$C_{j-1,k} = \sum_{n\in\mathbb{Z}} \overline{H_{n-2k}} \, C_{j,n} = \sum_{n\in\mathbb{Z}} \overline{H_n} \, C_{j,n+2k},$$
$$D_{j-1,k} = \sum_{n\in\mathbb{Z}} \overline{G_{n-2k}} \, C_{j,n} = \sum_{n\in\mathbb{Z}} \overline{G_n} \, C_{j,n+2k}.$$

Definition 4 (Inverse discrete multiwavelet transform)**.** The *inverse discrete multiwavelet transform* is the mapping

$$\left(\left\{C_{j,k} \in \ell^2(\mathbb{Z})^{2d}\right\}_{k\in\mathbb{Z}}, \left\{D_{j,k} \in \ell^2(\mathbb{Z})^{2d}\right\}_{k\in\mathbb{Z}} \right) \to \left\{C_{j+1,k} \in \ell^2(\mathbb{Z})^{2d}\right\}_{k\in\mathbb{Z}}, \tag{8}$$

defined by

$$C_{j+1,n} = \sum_{k\in\mathbb{Z}} H_{n-2k}^T \, C_{j,k} + \sum_{k\in\mathbb{Z}} G_{n-2k}^T \, D_{j,k}.$$

The discrete transforms (7) and (8) are sometimes referred to as *decomposition* and *reconstruction*, respectively.

3. Preprocessing and postprocessing

Let $\{x_n\}_{n\in\mathbb{Z}}$ be a sequence of increasing *sampling points* in $\mathbb{R}$. Let $f(t)$ be a function of a continuous variable t and $f[x_n]$ denote a discretized version of f defined on the sequence $\{x_n\}$. The sequence $\{f[x_n]\}_{n\in\mathbb{Z}}$ is called the *sampled data* or the *sampling data*. For each pair of successive sampling points, x_n and x_{n-1}, define the *sampling width* by

$$\Delta_n := x_{n+1} - x_n.$$

If all the sampling widths are equal, the sampling points, $\{x_n\}_{n\in\mathbb{Z}}$, are said to be *regularly sampled*, otherwise they are *irregularly sampled*.

Let $f \in V_0 \subset L^2(\mathbb{R})$ be a signal to be analyzed. The following Procedure 1, as illustrated in Figure 1, is typical of discrete multiwavelet analysis.

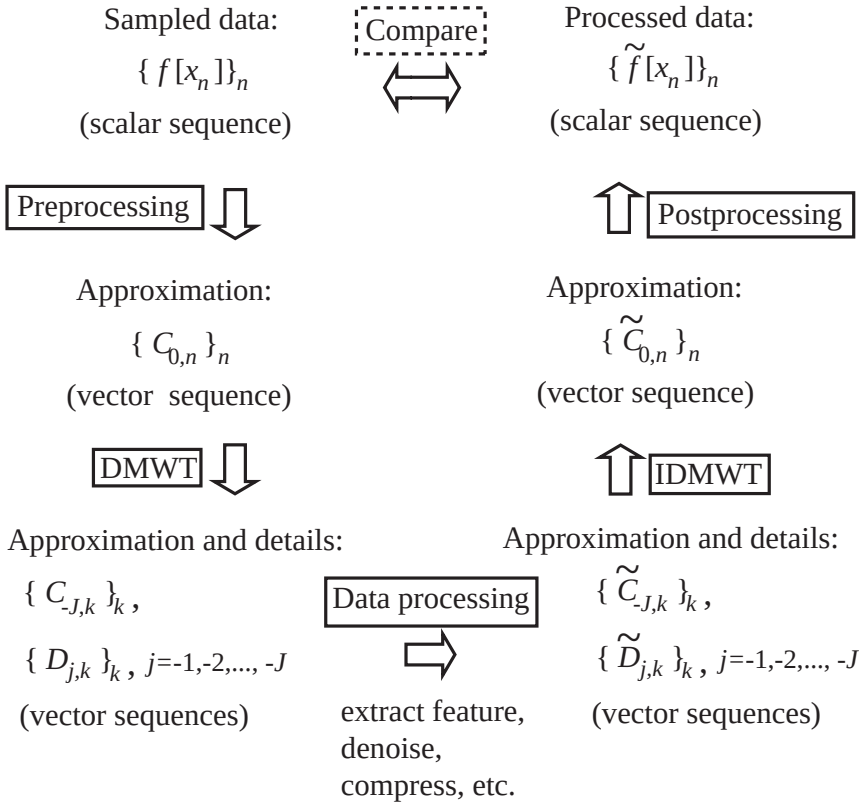

Figure 1. Typical discrete multiwavelet analysis procedure.

Procedure 1 (Discrete multiwavelet analysis)**.**

(i) *Given $f(t)$, construct a sampled data $\{f[x_n]\}_{n\in\mathbb{Z}}$.*

(ii) *Use $\{f[x_n]\}_{n\in\mathbb{Z}}$ to construct appropriate approximation vector coefficients $C_{0,n} \in \ell^2(\mathbb{Z})^d$ for $n \in \mathbb{Z}$. This procedure is called preprocessing.*

(iii) *Do J iterations of the discrete multiwavelet transform, starting with $C_{0,n} \in \ell^2(\mathbb{Z})^d$ for $n \in \mathbb{Z}$, to get the coarsest approximation coefficients $\{C_{-J,k} \in \ell^2(\mathbb{Z})^d\}_{k\in\mathbb{Z}}$ and the coarser detail coefficients $\{D_{j,k} \in \ell^2(\mathbb{Z})^d\}_{k\in\mathbb{Z}}$ for $j = -1, -2, \ldots, -J$.*

(iv) *Process those coefficients to get processed approximation coefficients* $\{\widetilde{C}_{-J,k} \in \ell^2(\mathbb{Z})^d\}_{k\in\mathbb{Z}}$ *and processed detail coefficients* $\{\widetilde{D}_{j,k} \in \ell^2(\mathbb{Z})^d\}_{k\in\mathbb{Z}}$, *for* $j = -1, -2, \ldots, -J$.

(v) *Do* J *iterations of the inverse discrete multiwavelet transform starting with* $\{\widetilde{C}_{-J,k} \in \ell^2(\mathbb{Z})^d\}_{k\in\mathbb{Z}}$ *and* $\{\widetilde{D}_{j,k} \in \ell^2(\mathbb{Z})^d\}_{k\in\mathbb{Z}}$, *for* $j = -1, -2, \ldots, -J$ *to get the processed approximation coefficients* $\{\widetilde{C}_{0,n} \in \ell^2(\mathbb{Z})^d\}_{n\in\mathbb{Z}}$.

(vi) *Construct an appropriate sequence* $\{\widetilde{f}_n\}_{n\in\mathbb{Z}}$ *from* $\{\widetilde{C}_{0,n} \in \ell^2(\mathbb{Z})^d\}_{n\in\mathbb{Z}}$ *which should have the same structure as the original sequence* $\{f[x_n]\}_{n\in\mathbb{Z}}$. *This procedure is called postprocessing.*

In this paper, we deal only with the preprocessing and postprocessing steps, leaving data processing for future work.

3.1. **The simplest preprocessing and postprocessing for uniwavelets.** Let ψ be a uniwavelet function with scaling function φ. Assume that $f \in \overline{\mathrm{Span}}\,\{\varphi_{j_0,k}\}_{k\in\mathbb{Z}}$ for fixed $j_0 \in \mathbb{Z}$ and its sampled data $\{f_n := f[x_n]\}_{n\in\mathbb{Z}}$ is given. Then, f can be represented as

$$f(x) = \sum_{k\in\mathbb{Z}} \langle f, \varphi_{j_0,k}\rangle\, \varphi_{j_0,k}(x).$$

The preprocessing procedure for uniwavelets is to get the coefficients $\{\langle f, \varphi_{j_0,k}\rangle\}_{k\in\mathbb{Z}}$ from the sampled data $\{f_n\}_{n\in\mathbb{Z}}$. Recall that $2^j\,\varphi(2^j x - k)$ tends to Dirac's δ function as $j \to \infty$, because scaling functions for uniwavelets are usually required to be continuous, well-localized and to satisfy

$$\int_{\mathbb{R}} \varphi(x)\,dx = 1.$$

We may assume that j_0 is large enough and that x_0 satisfies $k \sim 2^{j_0}x_0$. Then we have the following correspondence between coefficients and sampled data:

$$2^{j_0/2}\,\langle f, \varphi_{j_0,k}\rangle = \int_{\mathbb{R}} f(x)\,2^{j_0}\,\overline{\varphi(2^{j_0}x - k)}\,dx \;\sim\; f(x_0).$$

Thus, the simplest preprocessing and postprocessing for uniwavelets are the following:

Procedure 2 (Preprocessing for uniwavelets)**.** *For given sampled data* $\{f_n\}_{n\in\mathbb{Z}}$, *assume* $j_0 = 0$ *and obtain the approximation coefficients by* $\langle f, \varphi_{0,n}\rangle := f_n$.

Procedure 3 (Postprocessing for uniwavelets)**.** *For given coefficients* $\{\langle f, \varphi_{0,n}\rangle\}_{n\in\mathbb{Z}}$, *obtain the sampled data by* $\widetilde{f}_n := \langle f, \varphi_{0,n}\rangle$.

3.2. **Simple preprocessing for multiwavelets.** Many approaches to preprocessing for multiwavelets have been proposed. A short summary is given in [5]. Here, as an example, we describe a simple preprocessing for multiwavelets. It is named *odd-even preprocessing* and based on the simplest preprocessing for uniwavelets explained above. We also give an example to show that a "good" preprocessing is needed.

Procedure 4 (Odd-even preprocessing)**.** *For given sampled data* $\{f_n\}_{n\in\mathbb{Z}}$, *obtain approximation coefficients by* $C_{0,n} := [f_{2n-1}, f_{2n}]^T$.

In our examples, we use the MATLAB sample data `leleccum.dat` which involves a real-world electrical consumption signal measured over the course of three days. This signal is particularly interesting because of noise introduced when a defect developed in the monitoring equipment as measurements were being made. We use the Chui–Lian CL2 multiwavelet given in [10]. Figure 2 illustrates the multiscaling and multiwavelet functions of CL2.

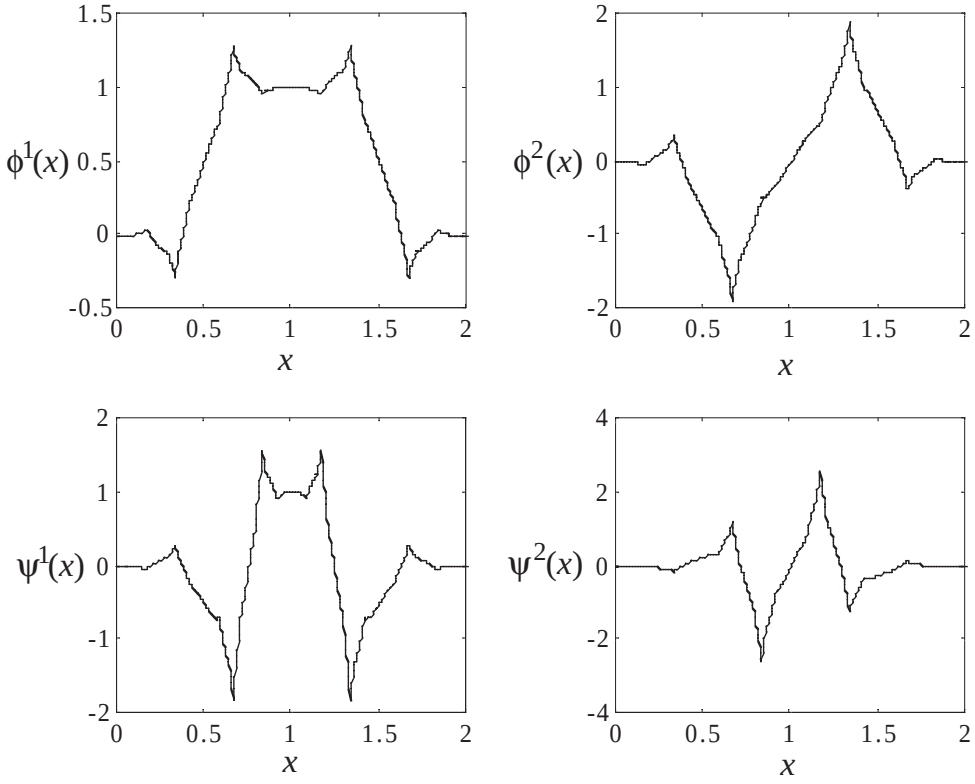

FIGURE 2. CL2 multiscaling functions, $\phi^1(x)$ and $\phi^2(x)$, and multiwavelet functions, $\psi^1(x)$ and $\psi^2(x)$.

Example 1 (Odd-even preprocessing for the CL2 multiwavelet). The data `leleccum.dat` is an electrical consumption measured (say, in kilowatts) every minute over the course of 3 days, that is, 4320 minutes. We use the first 1024 points of this data.

(i) Apply the odd-even preprocessing to get the approximation $C_0 = [C_0^1, C_0^2]^T$, which is shown in the top two plots of Figure 3.

(ii) Apply the CL2 discrete multiwavelet transform twice to get $C_2 = [C_2^1, C_2^2]^T$, $D_2 = [D_2^1, D_2^2]^T$, and $D_1 = [D_1^1, D_1^2]^T$. The result is in the bottom parts of Figure 3, where C_2^1, D_2^1 and D_1^1 are in the left-hand side and C_2^2, D_2^2 and D_1^2 are in the right-hand side.

Following standard notation, C_j^i denotes the set of coefficients of the approximation part by the scaling function ϕ^i at level j. Similarly, D_j^i denotes the set of coefficients of the detail part by the wavelet function ψ^i at level j.

Obsevation 1. The odd-even preprocessing is not well suited for the CL2 multiwavelet because D_1^2 and $-C_0^2$ have a similar shape and the maximum norm of D_1^2 is big. This means that D_1^2 contains low frequency information, which should be contained in the approximation coefficients, that is, C_2 in this case.

4. Least squares curve fitting to irregularly sampled data

Generally, given a finite set of irregularly sampled points $\{(x_n, y_n)\}$ in the plane, one tries to fit a curve $f(x; a_1, a_2, \ldots, a_r)$ to these points in the weighted least squares sense by

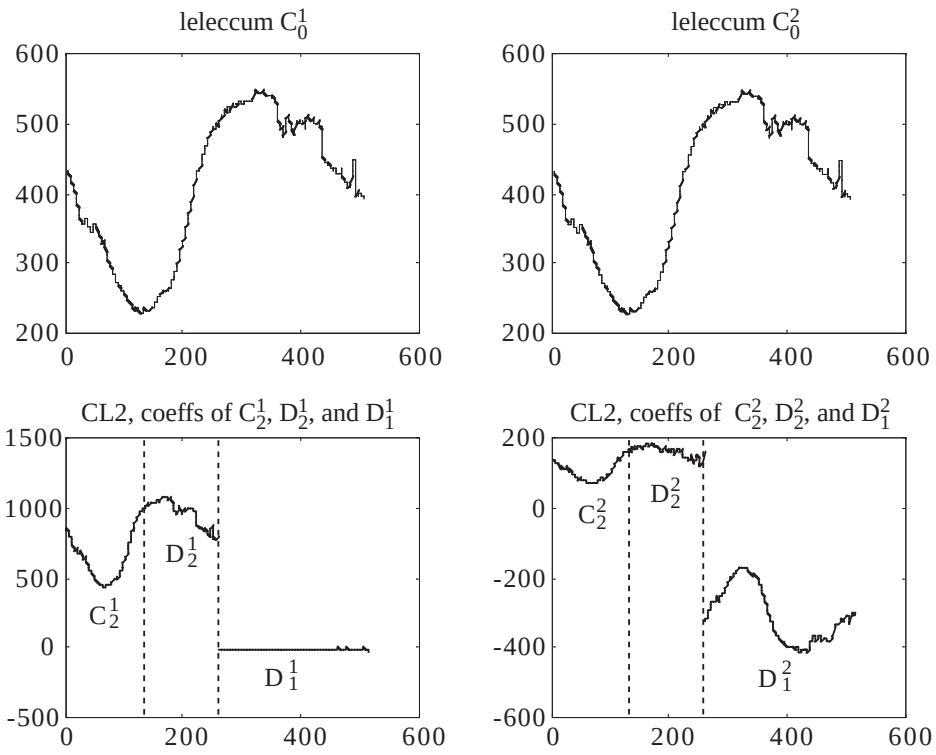

FIGURE 3. Plot of electricity consumption (say, in kilowatts) against time in minutes of the odd-even preprocessed `leleccum.dat` and its discrete multiwavelet transforms for Example 1.

minimizing the sum of squares

$$\sum_n |f(x_n; a_1, a_2, \dots, a_r) - y_n|^2 \left(\frac{\Delta_n + \Delta_{n-1}}{2}\right)$$

over the parameters $\{a_j\}$. This model will be adapted to the problem in hand in the following subsections.

4.1. **Curve fitting with multiscaling functions.** Hereafter, we only deal with the real-valued case and assume that the number of multiscaling functions is two, that is, $d = 2$.

First, let us find the least squares approximation to a continuous function $f \in L^2(\mathbb{R}) \cap C(\mathbb{R})$ in V_j. As each element $s_j \in V_j$ is represented by the two sums

$$s_j(x) = \sum_k c_{j,k}^1 \varphi^1(2^j x - k) + \sum_k c_{j,k}^2 \varphi^2(2^j x - k), \tag{9}$$

our problem is to find coefficients $c_{j,k}^1$ and $c_{j,k}^2$ that minimize the integral

$$\int_{\mathbb{R}} |f(x) - s_j(x)|^2 \, dx. \tag{10}$$

When $\Phi = [\varphi^1, \varphi^2]^T$ is an orthonormal multiscaling function, the best approximation is given by the integrals

$$c^1_{j,k} = 2^j \int f(x)\varphi^1(2^j x - k)\,dx, \qquad c^2_{j,k} = 2^j \int f(x)\varphi^2(2^j x - k)\,dx. \tag{11}$$

which can be calculated by numerical integration. In general, given an irregularly sampled data $\{f[x_n]\}$, integral (10) can be approximated by the formula

$$E(c^1_{j,k}, c^2_{j,k}) = \sum_n |f[x_n] - s_j[x_n]|^2 \left(\frac{\Delta_n + \Delta_{n-1}}{2}\right). \tag{12}$$

We expect the least squares solution to (12) to be an accurate approximation to $f[x_n]$ at the points x_n and the function $s_j(x)$ defined by the least squares solution to give "the best-fitting" curve to the given irregularly sampled data.

4.2. **Curve fitting with shifted multiscaling functions.** For certain types of multiscaling functions, Φ, and sampling points, $\{x_n\}$, it can happen that a given data $\{f[x_n]\}$ cannot be well approximated by (9). For example, the multiscaling functions $CL2$ and $CL3$, to be discussed in section 5, present a structural problem when solving a finite dimensional version of the equation

$$\sum_{\ell=1}^{2} \sum_{k\in\mathbb{Z}} c^\ell_{j,k}\varphi^\ell_{j.k}(x_n) = f_{j,n}, \tag{13}$$

where $f_{j,n}$ are determined from j and $f[x_n]$. More precisely, when the left-hand side of (13) is represented in matrix form:

$$A\,\left[\ldots, c^1_{j,k}, \ldots, c^2_{j,k}, \ldots\right]^T, \tag{14}$$

where the components of A are $\varphi^\ell_{j.k}(x_n)$, a finite dimensional approximation of A is a singular matrix. In such a case, we propose to use a shifted function $s_j(x+\theta)$ instead of $s_j(x)$, where the *shift parameter* θ will be chosen so as to avoid such a structural problem. We call this procedure a *shifted scaling fitting* and its algorithm is as follows.

Algorithm 1 (Shifted scaling fitting)**.** *Minimize*

$$E_\theta(c^1_{j,k}, c^2_{j,k}) := \sum_n |f[x_n] - s_j[x_n + \theta]|^2 \left(\frac{\Delta_n + \Delta_{n-1}}{2}\right) \tag{15}$$

over $c^1_{j,k}$ *and* $c^2_{j,k}$ *for fixed* θ*.*

5. Multiwavelet neural networks

The field of neural networks started some fifty years ago but has found solid applications only in the past twenty years and it is developing rapidly. Neural networks described in Rumelhart and McClelland [11] are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the network function is determined largely by the connections between elements.

Assume that each summation of $\varphi^1(2^j x - k)$ and $\varphi^2(2^j x - k)$ in (9) contains L terms and consider a three-layer neural network with input x and output $s_j(x)$ as shown in Figure 4. Then, the backpropagation learning method gives the least squares solution to (15). We call such a neural network a *multiwavelet neural network*.

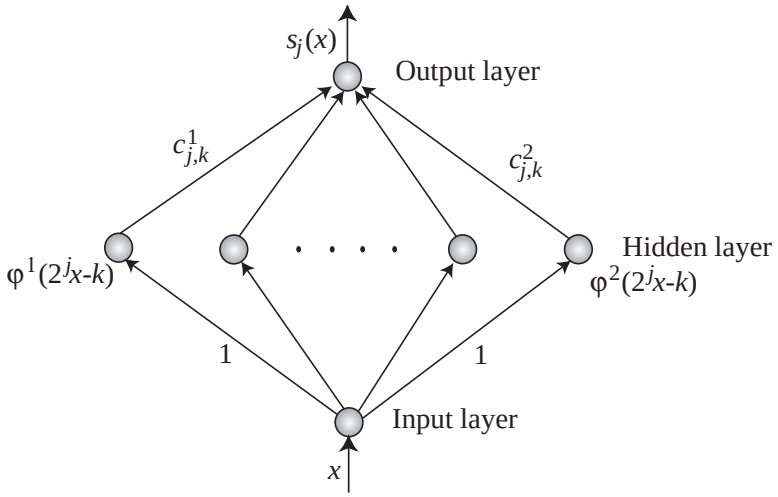

FIGURE 4. A three-layer multiwavelet neural network.

5.1. **Training algorithm.** Neural networks described [12] and [13] can be trained to perform a particular function by properly choosing the values of the connections (weights) between elements. Commonly, neural networks are adjusted, or trained, so that a particular input leads to a specific target output. The network is adjusted by comparing the output and the target until the network output matches the target. Typically, in supervised learning, many input/target pairs are used to train a network.

The training algorithm for our multiwavelet neural networks consists in the four steps listed in Algorithm 2.

Algorithm 2 (Training algorithm)**.** *Let input x be given.*
(i) Fix the resolution j and set $m = 0$ (the number of trainings). Choose appropriate initial coefficients $c^{\ell,[m]}_{j,k}$, $\ell = 1, 2$. Set the conjugate gradients $dc^{\ell,[m]}_{j,k} = 0$. Calculate the initial square error $E^{[m]} = E_\theta\left(c^{1,[m]}_{j,k}, c^{2,[m]}_{j,k}\right)$.
(ii) Choose a constant $0 < \lambda^{[m]} < 1$ and calculate the conjugate gradients as follows:

$$dc^{\ell,[m+1]}_{j,k} = \frac{\partial E_\theta\left(c^{1,[m]}_{j,k}, c^{2,[m]}_{j,k}\right)}{\partial c^{\ell,[m]}_{j,k}} + \lambda^{[m]} dc^{\ell,[m]}_{j,k}.$$

(iii) Choose a constant $\eta^{[m]} > 0$ and calculate the new coefficients

$$c^{\ell,[m+1]}_{j,k} = c^{\ell,[m]}_{j,k} - \eta^{[m]} dc^{\ell,[m+1]}_{j,k}.$$

(iv) Calculate the square error

$$E^{[m+1]} = E_\theta\left(c^{1,[m+1]}_{j,k}, c^{2,[m+1]}_{j,k}\right).$$

If $E^{[m+1]}$ is small enough, then the training is good and the algorithm is stopped. Else if the relative error,

$$\frac{E^{[m]} - E^{[m+1]}}{E^{[m]}},$$

is small, then the algorithm is aborted and we conclude that more training is useless and a coarser resolution, $j - 1$, is needed for this experiment. Otherwise, set $m = m + 1$ and go to (ii).

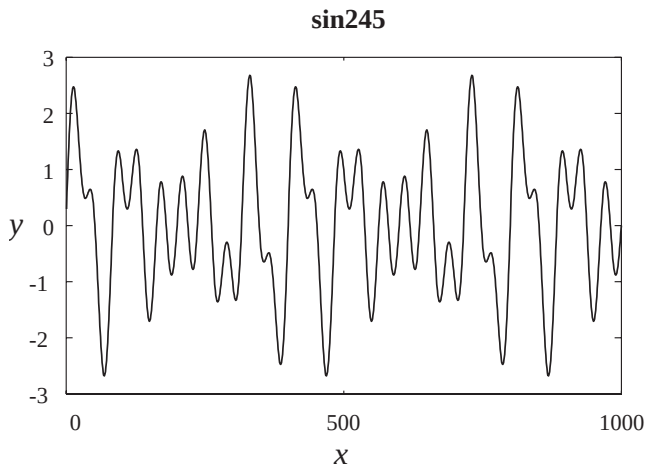

FIGURE 5. The graph of the function `sin245`: $y = \text{sin245}\,(x)$.

5.2. **Numerical results on curve fitting.** In our numerical experiments, we have applied shifted scaling fitting to various kinds of data. We state here only one of them. Computation was done with MATLAB version 6.5.1 on a PC running under Windows 2000 with 512MB of RAM. The CPU is AMD Athlon 1.13GHz.

In multiwavelet neural networks, the pairs of input and output $\big(x, s_j(x+\theta)\big)$ are known and the coefficients $\left[c^1_{j,k}, c^2_{j,k}\right]$ are unknown. We will deal with overdetermined systems. In this case, the number of input and output pairs $\big(x, s_j(x+\theta)\big)$ exceeds the number of unknown coefficients $\left[c^1_{j,k}, c^2_{j,k}\right]$.

5.2.1. *Multiscaling functions used in numerical experiments.* The following three types of multiscaling functions have been used in our numerical experiments.

- `ANV2-3` and `ANVb4.4`: The multiscaling functions, with supports $[0,3]$, $[0,3]$ and $[-3,3]$, $[-3,4]$, respectively, of Ashino, Nagase, and Vaillancourt [7] are generated by Daubechies' compactly supported scalar wavelets with $N = 2$ and the biorthogonal wavelet 9/7, respectively.
- `CL2` and `CL3`: The multiscaling functions of Chui and Lian [10] with $N = 2$ and $N = 3$, respectively, with support $[0, N]$.
- `GHM`: The multiscaling function of Geronimo, Hardin, and Massopust [14] and its multiwavelet function given in Donavan, Geronimo, Hardin, and Massopust [15]. The supports of the two components of the multiscaling function are $[0,1]$ and $[0,2]$, respectively.

5.2.2. *Irregularly sampled data used in numerical experiments.* To generate an irregularly sampled data, we use the following function `sin245`:

$$\text{sin245}\,(x) = \sin\frac{\pi}{20}x + \sin\frac{\pi}{40}x + \sin\frac{\pi}{50}x, \qquad 0 \le x \le 1000,$$

which is illustrated in Figure 5. Define a function named `noised-sin245` by adding to the function `sin245` a white noise with values in the interval $[-0.2, 0.2]$.

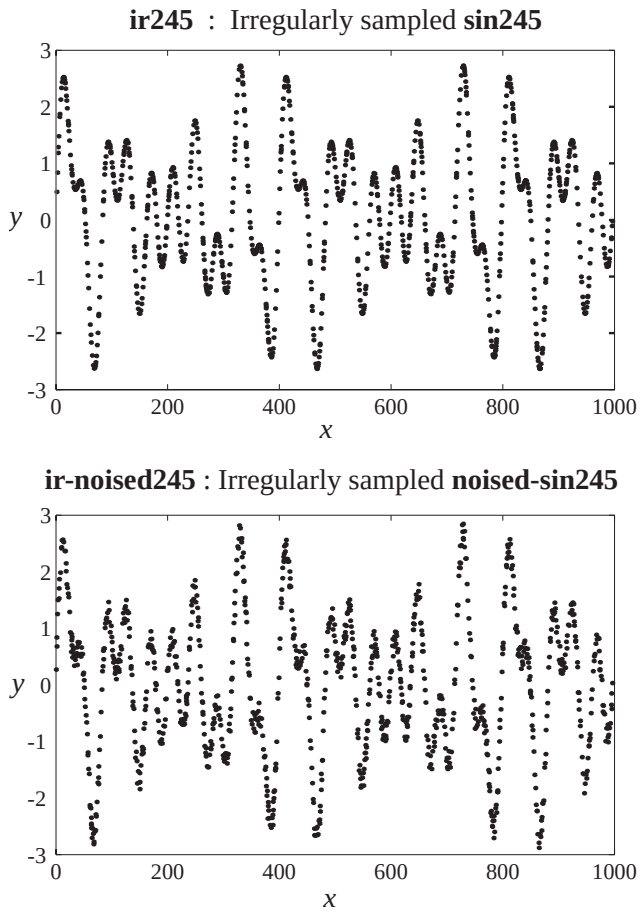

FIGURE 6. Irregular sampling, y vs. x, of `ir245` (top) and `ir-noised245` (bottom).

Irregular sampling widths $\{\Delta_n\}_{n\in\mathbb{Z}_+}$ are generated by uniform random numbers in $[0.1, 1.9]$ and the sampling points $\{x_n\}_{n\in\mathbb{Z}_+}$ are defined recursively by

$$x_0 := 0, \qquad x_{n+1} := x_n + \Delta_n, \ \ n \in \mathbb{Z}_+.$$

We take the sampling points $\{x_n\}$ in the open interval $I = (0, 1000)$. In our experiment, there were 1004 sampling points in I, the first being 1.5044 and the last being 999.5026. We denote the sampling points in I by $\{x_n\}$ without referring to the index set.

An irregularly sampled data `ir245` is given by sampling the function `sin245` at the sampling points $\{x_n\}$. An irregularly sampled noised data `ir-noised245` is given by sampling the function `noised-sin245` at the sampling points $\{x_n\}$. Figure 6 illustrates these irregularly sampled data, where the dots "•" denote the irregular sampling points.

Example 2 (Curve fitting to `ir245` and `ir-noised245` by V_{-3}). For irregularly sampled data, it is necessary that the chosen resolution, $j \leq -2$, gives rise to overdetermined systems. Hence we choose $j = -3$.

The following notation will be used:

TABLE 1. Accuracy of curve fitting to `ir245` by V_{-3}.

Wavelet	N_c	θ	N_ℓ	E_ℓ	E_M
`GHM`	252	0	10	3.73E-01	5.23E-02
	254	0.1	10	4.00E-01	5.89E-02
`CL2`	252	0	12	1.05E+00	8.88E-02
	254	0.1	10	1.08E+00	9.19E-02
`CL3`	254	0	12	4.76E-02	1.92E-02
	256	0.1	11	4.60E-02	2.31E-02
`ANV2-3`	254	0	10	9.15E-01	1.06E-01
	256	0.1	10	9.07E-01	1.04E-01
	256	0.2	10	9.85E-01	1.03E-01
`ANVb4.4`	266	0	10	1.70E-02	2.18E-02
	268	0.1	10	1.95E-02	4.34E-02

TABLE 2. Accuracy of curve fitting to `ir-noised245` by V_{-3}.

Wavelet	N_c	θ	N_ℓ	E_ℓ	E_M
`GHM`	252	0	10	9.80E+00	2.77E-01
	254	0.1	10	9.89E+00	2.83E-01
`CL2`	252	0	12	1.03E+01	3.05E-01
	254	0.1	10	1.01E+01	3.05E-01
`CL3`	254	0	11	9.50E+00	2.82E-01
	256	0.1	11	9.32E+00	2.77E-01
`ANV2-3`	254	0	10	1.06E+01	2.79E-01
	256	0.1	10	1.01E+01	2.94E-01
	256	0.2	10	1.04E+01	3.13E-01
`ANVb4.4`	266	0	10	9.33E+00	2.79E-01
	268	0.1	10	9.69E+00	2.80E-01

j	Resolution level	N_l	Number of learnings
N_c	Number of coefficients	E_l	Square error after learning
θ	Shift parameter	E_M	Maximum error after learning

The accuracy of curve fitting to `ir245` and `ir-noised245` by V_{-3} are given in Tables 1 and 2, respectively. The graphs of `sin245` (top) and the fitting curve to `ir245` by V_{-3} of GHM (bottom) are given in Figure 7 and the graphs of `noised-sin245` (top) and the fitting curve to `ir-noised245` by V_{-3} of GHM (bottom) are given in Figure 8. Corresponding graphs for other multiscaling functions are similar.

Obsevation 2. Table 2 shows that the curve fitting accuracy is stable under the changes of θ and multiscaling functions. The square errors after learning, E_l, are around 10 and the maximum error after learning, E_M, is around 0.3, which is close to the maximum value of the energy of the noise. This means that we can have stable curve fitting under noise. Figures 7 and 8 show that our curve fitting performs well with the GHM multiscaling function and this is also true for other multiscaling functions. Comparing `noised-sin245` (top) with the fitting curve to `ir-noised245` (bottom) in Figure 8, we can see the good denoising property of our curve fitting using the GHM multiscaling function and this is also true for other multiscaling functions.

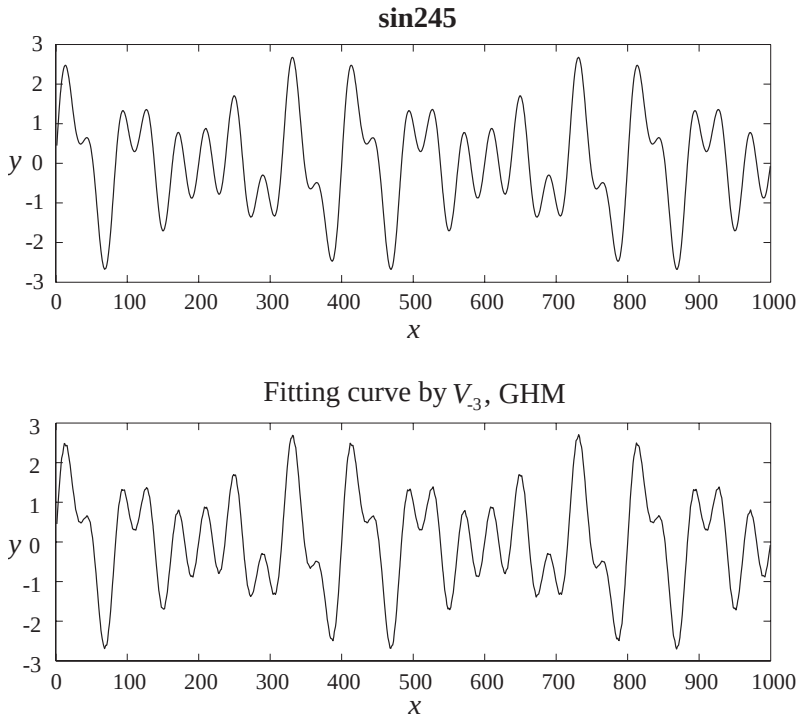

FIGURE 7. The graphs, y vs. x, of `sin245` (top) and the fitting curve to `ir245` by V_{-3} of GHM (bottom).

Example 3 (Neural networks preprocessing for CL2). We use the data `leleccum.dat` of Example 1.

(i) Apply the multiwavelet neural networks preprocessing to get an approximation $C_0 = [C_0^1, C_0^2]^T$. The result is given in the top row of Figure 9.
(ii) Apply the CL2 discrete multiwavelet transforms twice to get $C_2 = [C_2^1, C_2^2]^T$, $D_2 = [D_2^1, D_2^2]^T$, and $D_1 = [D_1^1, D_1^2]^T$. The result is shown in the bottom row of Figure 9, where C_2^1, D_2^1, and D_1^1 are in the left-hand side and C_2^2, D_2^2, and D_1^2 are in the right-hand side.

Obsevation 3. The difficulty with the odd-even preprocessing stated in Obsevation 1 is that D_1^2 contains low frequency information. For our multiwavelet neural network preprocessing, it seems that D_1^2 does not contain low frequency information, because the coefficients D_1^2 highly oscillates and the maximum norm of D_1^2 is small. Hence, our multiwavelet neural networks preprocessing is well adapted to the CL2 multiwavelet.

6. Preprocessing and postprocessing based on curve fitting

Our preprocessing design is the following.

Procedure 5 (Multiwavelet neural network preprocessing)**.**

(i) *Try to find a minimizing solution of* (15) *for* $\theta = 0$ *by multiwavelet neural networks.*
(ii) *If step (i) is hard, perturb* θ *to find a minimizing solution.*
(iii) *Denote the minimizing solution by* $C_{j,k}^* := [c_{j,k}^{1*}, c_{j,k}^{2*}]^T$.
(iv) *Obtain approximate coefficients* $\{C_{j,k}\}_{k\in\mathbb{Z}}$ *by* $C_{j,k} := 2^{-j/2} C_{j,k}^*$.

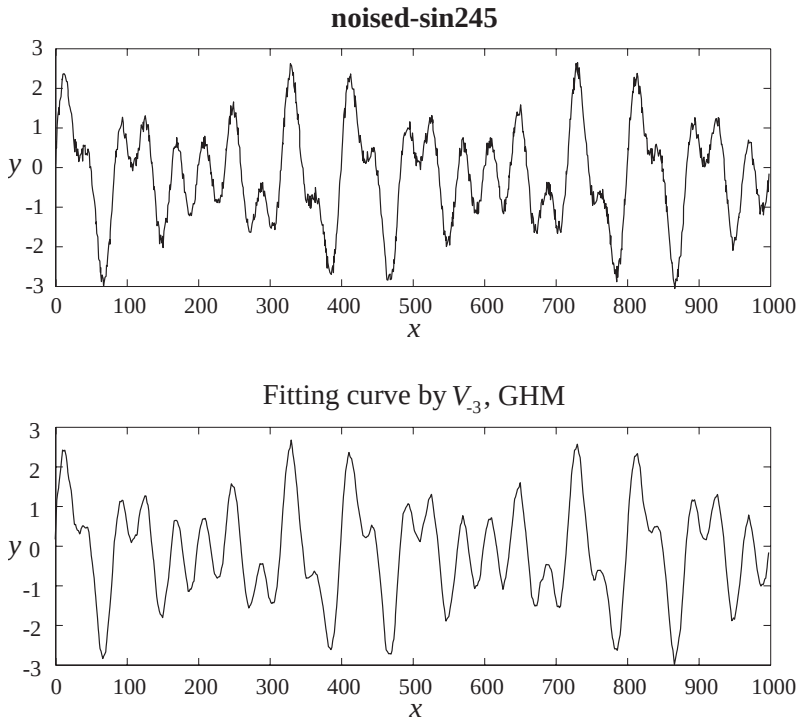

FIGURE 8. The graphs, y vs. x, of `noised-sin245` (top) and the fitting curve to `ir-noised245` by V_{-3} of GHM (bottom).

Our postprocessing design is the following.

Procedure 6 (Multiwavelet neural network postprocessing)**.** *Obtain processed data* $\{\widetilde{f}_n\}_{n\in\mathbb{Z}}$ *by*

$$\widetilde{f}_n := \sum_k c^{1*}_{j,k}\varphi^1\big(2^j(x_n+\theta)-k\big) + \sum_k c^{2*}_{j,k}\varphi^2\big(2^j(x_n+\theta)-k\big).$$

7. Conclusion

Various numerical experiments have led to the following conclusions.

(i) Our curve fitting method is stable under noise for various multiscaling functions.
(ii) Our multiwavelet neural network preprocessing performs well for irregularly sampled data.

This proposed preprocessing procedure combined with multiwavelet neural networks for data-adaptive curve fitting is shown to perform well in the case of high resolution. In the case of low resolution it is more accurate than numerical integration and cheaper than matrix inversion.

Acknowledgement

This research was partially supported by the Japanese Ministry of Education, Culture, Sports, Science and Technology, Grant-in-Aid for Scientific Research (C) 15540170 (2003-2004), (C) 17540158 (2005-2006) and the Japan Society for the Promotion of Science, Japan-Canada Scientist Exchange Program (2004), and the Natural Sciences and Engineering Research Council of Canada.

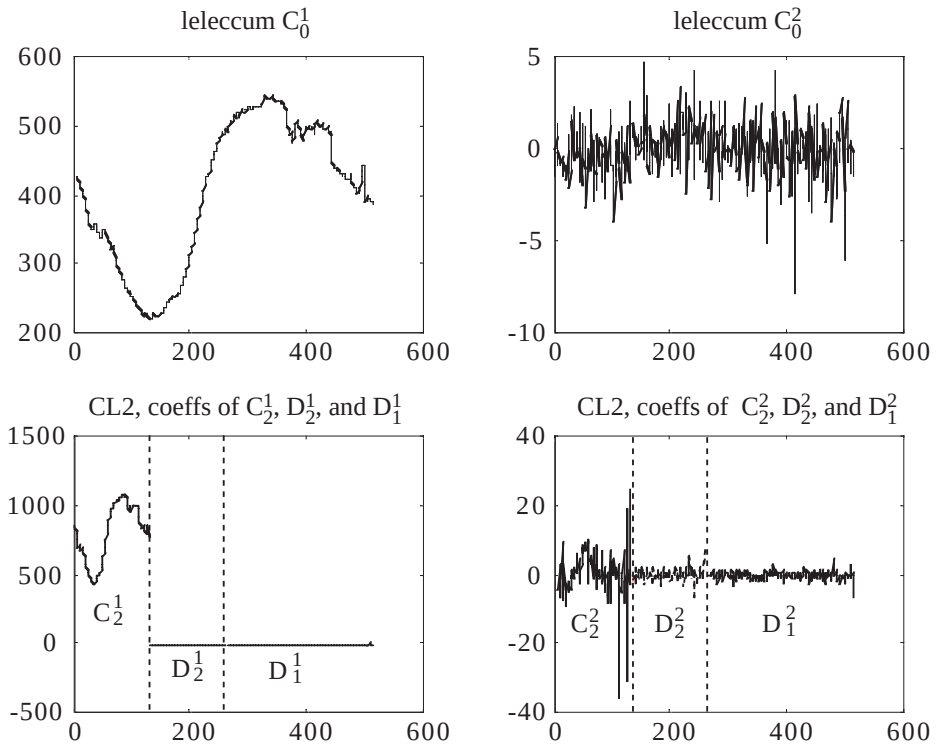

FIGURE 9. Plot of electricity consumption (say, in kilowatts) against time in minutes of the neural network least squares preprocessed leleccum data, `leleccum.dat`, and its discrete multiwavelet transforms for Example 3.

REFERENCES

[1] Y. Meyer, *Wavelets and operators*, Cambridge Studies in Advanced Mathematics **37**, Cambridge University Press, 1992.

[2] X.-G. Xia, J. S. Geronimo, D. P. Hardin, and B. W. Suter, *Design of prefilters for discrete multiwavelet transforms*, IEEE Trans. on Signal Processing **44**, No. 1 (1996) 25–35.

[3] X.-G. Xia, J. S. Geronimo, D. P. Hardin, and B. W. Suter, *Why and how prefiltering for discrete multiwavelet transforms*, Proceedings of IEEE ICASSP'96, Atlanta (1996) 1578–1581.

[4] G. Strang and T. Nguyen, *Wavelets and filter banks*, Wellesly-Cambridge Press, Box 812060, Wellesly MA 02181, 1997.

[5] F. Keinert, *Wavelets and multiwavelets*, Chapman & Hall/CRC, Boca Raton FL, 2004.

[6] A. Morimoto, R. Ashino, and R. Vaillancourt, *Pre-processing design for multiwavelet filters using neural networks*, International J. of Wavelets, Multiresolution and Information Processing **2**, No. 2 (2004) 133–148.

[7] R. Ashino, M. Nagase, and R. Vaillancourt, *A construction of multiwavelets*, Computer Math. Applic. **32** (1996), 23–37.

[8] E. Y. Zheng, *A comparative study of wavelets and multiwavelets*, Ottawa-Carleton Institute of Mathematics and Statistics, M.Sc. thesis, (1996).

[9] S. Mallat, *Multiresolution approximations and wavelet orthonormal bases of* $L^2(\mathbb{R})$, Trans. Amer. Math. Soc. **315** (1989), 69–87.

[10] C. K. Chui and J. Lian, *A study of orthonormal multi-wavelets*, Appl. Numer. Math. **20** (1996), 273–298.

[11] D. E. Rumelhart and J. L. McClelland, *Parallel distributed processing: Explorations in the microstructure of cognition: Foundations (Parallel distributed processing)*, MIT Press, Cambridge MA, 1986.

[12] H. B. Demuth and M. H. Beale, *Neural network toolbox for use with Matlab*, Users' Guide, Version 3, The Mathworks, Natick MA, 1998.
[13] M. T. Hagan, H. B. Demuth and M. H. Beale, *Neural network design*, PWS, Boston MA, 1998.
[14] J. S. Geronimo, D. P. Hardin, and P. R. Massopust, *Fractal functions and wavelet expansions based on several scaling functions*, J. Approx. Theory **78** (1994), 373–401.
[15] G. C. Donavan, J. S. Geronimo, D. P. Hardin, and P. R. Massopust, *Construction of orthogonal wavelet using fractal interpolation functions*, SIAM J. Math. Anal. **27** (1996), 1158–1192.

Division of Information Science, Osaka Kyoiku University, Kashiwara, Osaka 582-8582, Japan
E-mail address: morimoto@cc.osaka-kyoiku.ac.jp

Division of Mathematical Sciences, Osaka Kyoiku University, Kashiwara, Osaka 582-8582, Japan
E-mail address: ashino@cc.osaka-kyoiku.ac.jp

Department of Mathematics and Statistics, University of Ottawa, 585 King Edward Ave., Ottawa ON Canada K1N 6N5
E-mail address: remi@uottawa,ca