Signal Decomposition with Multiscale Learning Algorithms

J. W. Wang, C. H. Chen, and *J. C. Luo*
Dept. of Electrical Engineering, National Cheng Kung University
1 University Road, Tainan 70101, Taiwan, R. O. C.

Abstract-In this paper, neural networks based on orthonormal wavelets are constructed to decompose signals into full scale space. Two algorithms, the *global multiscale learning* (GML) and the *pyramid multiscale learning* (PML) were proposed and comparatively studied for their convergence speed, accuracy, and multiresolution representation property in functional approximation. It is shown that the GML algorithm produces larger approximation errors, fails to approximate a chaotic signal, and does not posses multiresolution representation property. On the other hand, the PML algorithm produces smaller approximation errors, can approximate a chaotic signal, and possesses multiresolution representation property.

1. Introduction

Recent works have proposed the use of wavelet functions as activation functions and shown their powers in functional approximation and localization in time and frequency of the signals [1-3]. However, the training of wavelet neural networks using the back-propagation learning can be slow and inefficient due to the global interactions over every node of each scale. Moreover, most works published so far stressed on approximation accuracy and rarely discussed the multiresolution representation property. The study in this paper focuses on the question of how to decompose the input signals into their multiresolution representation with fast convergence speed and high accuracy. In what follows, two types of network learning algorithms will be studied and compared. One is the *global multiscale learning* (GML) which is stemmed from the back-propagation algorithm in the artificial neural network [4] and the other is the *pyramid multiscale learning* (PML) proposed in this paper which is based on the fact that functions can be represented as a weighted sum of orthogonal basis functions. Two example 1-D signals decomposed by the GML algorithm and the PML algorithm into full scale space will be illustrated.

This paper is organized as follows. In section 2, network structures and signal approximations are presented. Experimental results about 1-D signals are reported and discussed in Section 3. Finally, Section 4 gives the conclusion.

2. Network Structures and Signal Approximations

Feedforward neural networks have been widely studied and demonstrated for their ability for fitting complicated maps very well [5-7]. On the other hand, the recently introduced wavelet decomposition [8-11] has emerged as a powerful tool for functional approximation and turns out to have a structure very similar to the feedforward neural network. Two types of network learning methods, inspired by both the feedforward neural networks and wavelet decomposition, are introduced in this section.

2.1 Learning Algorithms

Let Θ be a set containing pairs of sampled inputs and $f(x) \in L^2(R)$ be the mapping to be approximated, i.e.

$$\Theta = \{(x_k, y_k) \mid y_k = f(x_k),\ x_k, y_k \in R,\ k = 0,1,\ldots,K-1,\ K < \infty\} \qquad (1)$$

Θ is called the training set and K is the number of training patterns. Based on Eq. (1) and [8], a single-input-single-output (SISO) feedforward neural network with only one hidden layer consisting of $K-1$ wavelets can be constructed as Fig. 1. The output of the network is given by

$$\hat{f}(x_k) = \bar{f} + \sum_{m=0}^{N_m-1} \sum_{n=0}^{N_n(m)-1} w_{mn}\, \psi_{mn}(x_k)\ ,\ \ w_{mn} \in R,\ m \text{ and } n \in Z \qquad (2)$$

where $\psi_{mn}(\cdot)$ is the wavelet family, w_{mn} are linear weights, and $\bar{f}$ is introduced to deal with nonzero mean functions on finite domain since the wavelet function $\psi_{mn}(\cdot)$ is zero mean. $K-1 = \sum_{m=0}^{N_m-1} N_n(m)$, m is the dilation parameter, n is the translation parameter, $N_m = \log_2 K$ is the number of the total scales, and $N_n(m) = 2^{N_m-m-1}$ is the number of the translations at scale m.

GML Algorithm: At each training iteration the GML algorithm updates the weights after the presentation of the complete set of training data and incorporates one sweep through all orthonormal wavelets. Furthermore, it uses a gradient search technique [4] to find the network weights that minimize an sum-of-squared-error function. All the weights of the GML algorithm are determined iteratively according to the following rules.

$$\Delta w_{mn} = \sum_{k=0}^{K-1} [\hat{f}(x_k) - y_k]\, 2^{-m/2}\, \psi(2^{-m} x_k - n) \qquad (3)$$

where $m = 0,\ 1,\ldots,\ N_m - 1$.

$$w_{mn}(z+1) = w_{mn}(z) - \eta\, \Delta w_{mn} \qquad (4)$$

where z is the iteration index and η is the learning rate.

PML Algorithm: If we rewrite the Eq. (2) as below

$$\hat{f}(x_k) = \bar{f} + w_{N_m-1,0}\Psi_{N_m-1,0}(x_k) + \cdots + \sum_{n=0}^{N_n(0)-1} w_{0,n}\Psi_{0,n}(x_k) \qquad (5)$$

We can construct another type of network structure as shown in Fig. 2 and find a distinction between these two networks. The GML updates all the weights simultaneously at each iteration, while the PML updates the weights by scale at each iteration. Hence, the algorithm of GML is an overall-scales learning and the algorithm of PML is a scale-by-scale learning. The approximation of the function $f(x)$ at the $(N_m - 1)$th scale is then given by

$$\hat{f}_{N_m-1}(x_k) = \bar{f} + w_{N_m-1,0}\psi_{N_m-1,0}(x_k) \tag{6}$$

The resulting network has only one node whose activation function is the wavelet function at scale $N_m - 1$. Suppose we want a more refined approximation of $f(x)$, that is, the approximation of $f(x)$ at scale $N_m - 2$. We proceed to calculate

$$\hat{f}_{N_m-2}(x_k) = \hat{f}_{N_m-1}(x_k) + \sum_{n=0}^{N_n(N_m-2)-1} w_{N_m-2,n}\psi_{N_m-2,n}(x_k) \tag{7}$$

Continuing with the addition of nodes at the scales $N_m - 3,\ N_m - 4, \ldots,\ 0$, we obtain a complete network as shown in Fig. 2. The approximation using the errors from the previous approximations as the training data approaches the unknown function $f(x)$ with progressively smaller global error. Finally, Eq. (5) describes what is known as the wavelet decomposition of a function $f(x) \in L^2(R)$.

The GML updates all the weights simultaneously at each iteration, while the PML updates the weights by scale at each iteration. The weights of each scale of the PML algorithm are adjusted scale-by-scale iteratively according to Eqs. (4) and (8) given below.

$$\Delta w_{mn} = \sum_{k=0}^{K-1}[\hat{f}_m(x_k) - \hat{f}_{m+1}(x_k)]\ 2^{-m/2}\ \psi\ (\ 2^{-m}x_k - n\ ) \tag{8}$$

where $m = 0,\ 1, .., \ N_m - 2$, when $m = N_m - 1$, $\hat{f}_{m+1}(x_k)$ is replaced with y_k.

2.2 Approximation Errors

When approximating a function with the GML algorithm, the global error is given by

$$\varepsilon^2 = \sum_{k=0}^{K-1} \frac{1}{2}\ [\hat{f}(x_k) - y_k]^2 \tag{9}$$

, while the local error with the PML algorithm at resolution 2^m is

$$\varepsilon_m^2 = \sum_{k=0}^{K-1}[\hat{f}_m(x_k) - \hat{f}_{m+1}(x_k)]^2 \tag{10}$$

and satisfies $\varepsilon_m^2 \to 0$ as $m \to -\infty$ by the properties of the multiresolution decomposition. Using Eq. (10), we can keep track the local errors of approximation as nodes of each scale are added to the network. The error of approximation from the GML algorithm considers overall scales (global) error as given by Eq. (9) while the error of approximation from the PML algorithm considers each scale (local) error as given by Eq. (10). This difference is the basis for the selection of appropriate learning methods for the wavelet network. Such a measure can provide an estimate of the reliability of the network prediction.

3. Experimental Results and Discussions

For simplicity of comparison, we confine our experiments on two different aspects based on the same learning parsimony (fixed number of nodes). The first is the learning accuracy and the second is the learning speed which is the amount of CPU time used for training.

Signals generated by chaotic systems represent a potentially rich class of signals both for detecting and characterizing physical phenomena and in synthesizing new classes of signals for communications, remote sensing, and a variety of other signal processing applications. From the point of view of signal processing, the approximation of this class of signals present a significant challenge and an opportunity to explore and develop completely a new type of algorithm to match the special characteristics of chaotic signals [12].The experiments conducted in this paper not only aim to obtain satisfactory approximation results but also gain insight into the behaviors about the resolutions (scales) and strengths (wavelet coefficients) of the wavelet projection. The latter can provide us with far more valuable information about the signals than the former.

In this section, two example signals are used to test the two learning algorithms. The equations of these two signals are given as below.

$$f_1(x) = \begin{cases} 1 & \text{if} \quad 0 \le x < 64 \\ -1 & \text{if} \quad 64 \le x < 128 \\ 1 & \text{if} \quad 128 \le x < 192 \\ -1 & \text{if} \quad 192 \le x < 256 \end{cases} \tag{11}$$

$$f_2(x) = \{\, x(t) \mid x[t] = 4x[t-1][1 - x(t-1)],\ \ 0 \le x < 256 \,\} \tag{12}$$

The first signal $f_1(x)$ is simply a periodic square wave and the second signal $f_2(x)$ is a chaotic signal. The wavelet bases adopted are the orthogonal family of Daubechies D_n, where n is 2, 4, and 20 [8].

3.1. Decomposition of 1-D signals using the GML Algorithm

Experiment 1: Figs. 3(a), (b), and (c) show the original, the approximation, and the error signals of $f_1(x)$ using D_2 wavelet. Fig. 4 reports the wavelet projections over each scale. From Fig. 3 and Fig. 4, we see that the network can be trained to approximate $f_1(x)$ with convergence. However, the network can not obtain the proper wavelet projections of the original signal.

Experiment 2: Figs. 5(a), (b), and (c) show the original, the approximation, and the error signals of $f_1(x)$ using D_2 wavelet. In this example, we find that the network can not converge to the original signal. Thus, the GML algorithm fails to approximate the chaotic signal.

3.2. Decomposition of 1-D signals using the PML Algorithm

Experiment 3: Figs. 6 (a), (b), and (c) show the original, the approximation, and the error signal of $f_1(x)$ using D_2 wavelet. Fig. 7 reports the wavelet projections over each scale. In this example, we find that the wavelet basis at scale 6 resembles $f_1(x)$ very well. Thus the projection at scale 6 generated from the network gives a good approximation to the original signal $f_1(x)$, while the other components are almost equal to zero. By retaining only projection of scale 6 of $f_1(x)$, one can achieve a rate-reduction.

Experiment 4: Figs. 8 (a), (b), and (c) show the original, the approximation, and the error signal of $f_2(x)$ using D_2 wavelet. Fig. 9 reports the wavelet projections over each scale. It is shown that $f_2(x)$ can be well approximated to the accuracy desired and does give the multiresolution representations of the signal.

Experiment 5: To get more insights into the PML algorithm, the corresponding experimental results of the approximation of $f_1(x)$ and $f_2(x)$ using D_4 and D_{20} wavelets were also obtained respectively. These decompositions are as good as expected to give the multiresolution representations of the signal [13].

3.3. Comparison of the GML algorithm and the PML algorithm

From the results obtained from *Experiments 1-5*, observations about the two learning algorithms are made as follows. The GML algorithm fails to approximate the chaotic signal and takes much longer time to converge for the signals it can approximate. This is because that the network converges to a local minimum and thus fails to obtain the exact mapping. This drawback is also found in the other networks that use the back-propagation algorithm [14]. On the other hand, the PML algorithm can overcome these drawbacks. The PML algorithm is easily constructed and guaranteed to converge to the desired accuracy in a duration shorter than of the GML algorithm. In addition, multiresolution representations of a signal can be obtained from the wavelet neural network using the PML algorithm. Comparable to the neural network trainings [1,2,15], the two algorithms like back-propagation does rely on stochastic gradient search but have different adaptation efficiencies of wavelet neural network. One way to explain this sluggishness is to characterize the error surface which is being searched. In the case of the PML algorithm the error surface is a relatively agreeable surface to search (with a single global minimum), however in the case of the GML algorithm the error surface is quite harsh (with local minima) [14].

4. Conclusion

In this paper, a pyramid wavelet neural network inspired by the feedforward neural network and discrete wavelet transform was constructed to decompose 1-D signals. The concept of multiresolution analysis is utilized to train the network parameters. The training algorithm based on this idea is named the *pyramid multiscale learning* (PML) algorithm. The PML algorithm can improve the drawbacks of the *global multiscale learning* (GML) algorithm, offer a multiresolution representation of the original signal, and easily be implemented.

References

[1] Q. Zhang, "Wavelet networks," *IEEE Trans. Neural Network*, vol. 3, no. 6, pp. 889-898, Nov. 1992

[2] T. I. Boubez and R. L. Peskin, "Wavelet neural networks and receptive field partitioning," *IEEE Int. Conf. Neural Networks*, vol. 3, pp. 1544-1549, Mar. 1993

[3] Y. C. Pati and P. S. Krishnaprasad, "Analysis and synthesis of feedforward neural networks using discrete affine wavelet transformations," *IEEE Trans. Neural Network*, vol. 4, no. 1, pp. 73-85, Jan. 1993

[4] R. Hecht-Nielsen, "Theory of the backpropagation neural network, " *Proc. IJCNN*, Washington D. C., I-593, Jun. 18-22, 1989

[5] G. Cybenko, "Approximation by superposition of a sigmoidal function, " *Mathematics of Control, Signal and Systems*, vol. 2, pp. 303-314, 1989

[6] S. M. Carrol and B. W. Dickinson, "Construction of neural nets using the radon transform," *Proc. IJCNN*, 1989

[7] A. R. Webb, "Functional approximation by feedforward networks: a least squares approach to generalization," *IEEE Trans. Neural Networks*, vol. 5, no. 3, pp. 363-371, May 1994

[8] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Comm. on Pure and Applied Math.*, vol. 91, pp. 909-996, 1988

[9] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Trans. PAMI*, vol. 11, no. 7, pp. 674-693, Jul. 1989

[10] —, "Multiresolution approximation and wavelets: orthonormal bases of $L^2(R)$," *Trans. Amer. Math. Soc.*, 315, no. 1, pp. 69-88, 1989

[11] S. G. Mallat and S. Zhong, "Characterization of signals from multiscale edges," *IEEE Trans. PAMI*, vol. 14, no. 7, pp. 710-732, Jul. 1992

[12] H. D. I. Abarbanel, "Chaotic signals and physical systems, " *Proc. IEEE ICASSP*, 1992

[13] J. C. Luo and C. H. Chen, "A study of signal decomposition using pyramidal wavelet networks," Master Thesis, National Cheng Kung Univ., Taiwan, R. O. C., Jun. 1995

[14] D. Hush, J. M. Salas, and B. Horne, "Error surfaces for multilayer perceptrons, " *IEEE Trans. SMC*, vol. 22, no. 5, 1992

[15] B. Delyon, A. Juditsky, and A. Benveniste, "Accuracy analysis for wavelet approximations," *IEEE Trans. Neural Networks*, vol. 6, no. 2, pp. 332-348, Mar. 1995

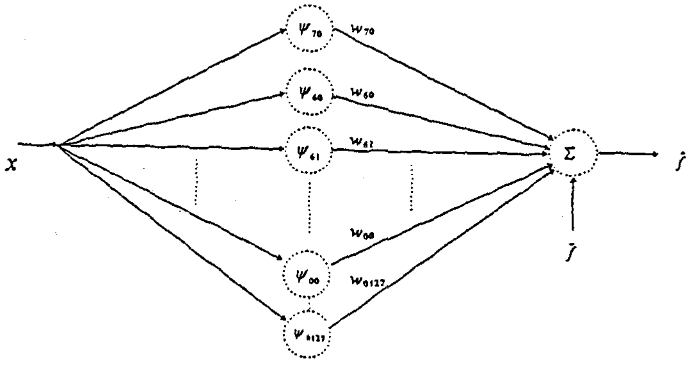

Fig. 1 Wavelet neural network with GML algorithm for K=256.

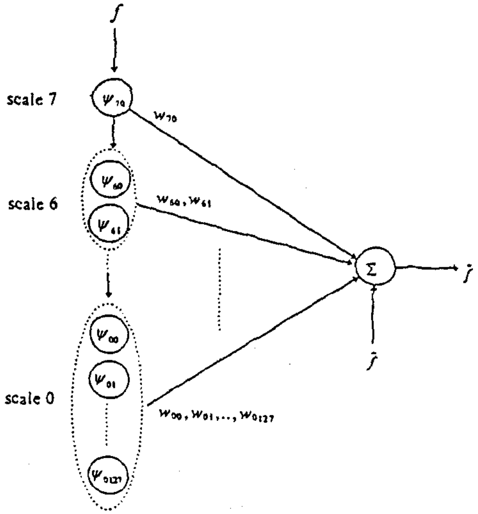

Fig. 2 Wavelet neural network with PML algorithm for K=256.

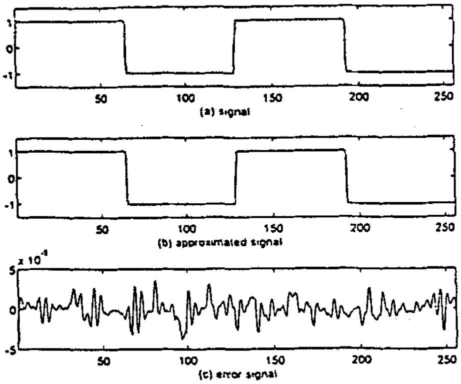

Fig. 3

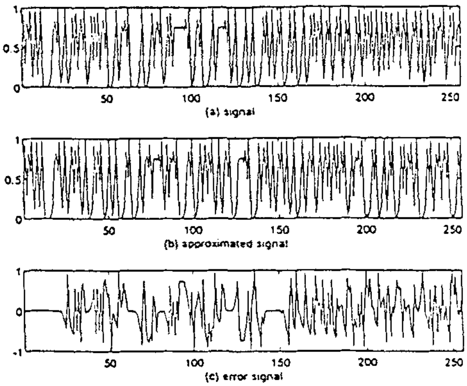

Fig. 5

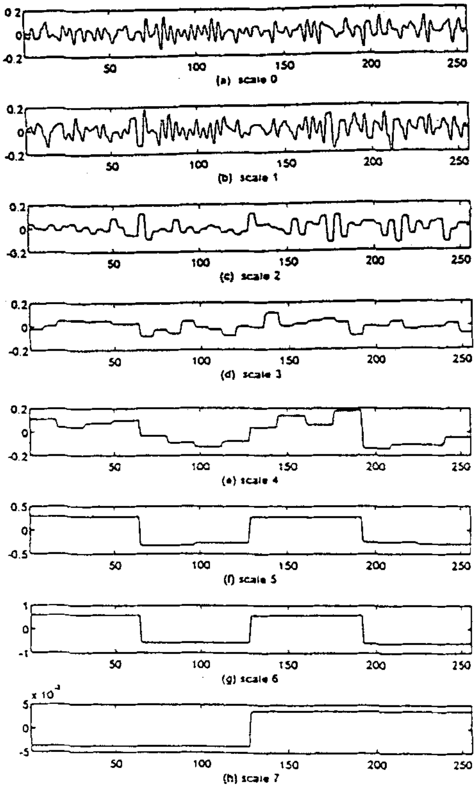

Fig. 4

Fig. 3 Experimental results using the GML algorithm for $f_1(x)$, (a) the original $f_1(x)$, (b) the approximated signal, (c) the error signal (D_2 wavelet).

Fig. 4 The wavelet projections of the approximated signal of $f_1(x)$ using the GML algorithm (D_2 wavelet).

Fig. 5 Experimental results using the GML algorithm for $f_2(x)$, (a) the original $f_2(x)$, (b) the approximated signal, (c) the error signal (D_2 wavelet).

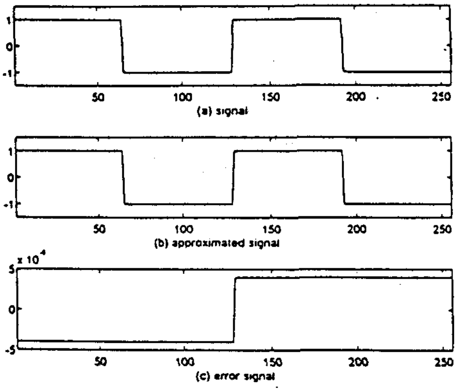

Fig. 6

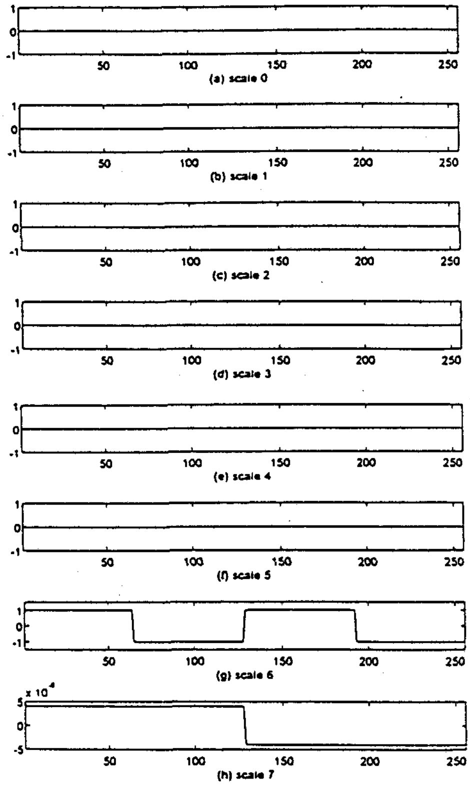

Fig. 7

Fig. 6 Experimental results using the PML algorithm for $f_1(x)$ (a) the original $f_1(x)$, (b) the approximated signal, (c) the error signal (D_2 wavelet).

Fig. 7 The wavelet projections of the approximated signal of $f_1(x)$ using the PML algorithm (D_2 wavelet).

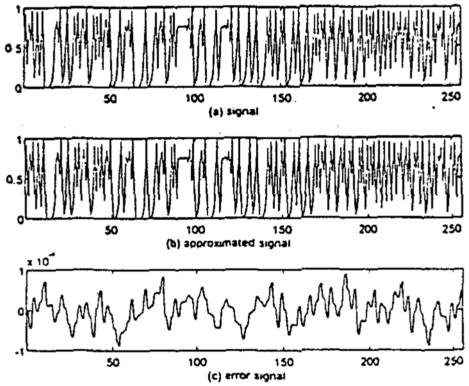

Fig. 8

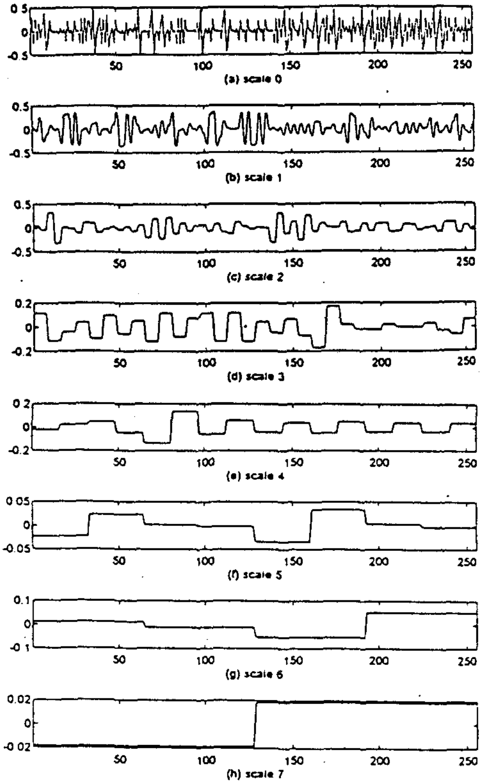

Fig. 9

Fig. 8 Experimental results using the PML algorithm for $f_2(x)$ (a) the original $f_2(x)$, (b) the approximated signal, (c) the error signal (D_2 wavelet).

Fig. 9 The wavelet projections of the approximated signal of $f_2(x)$ using the PML algorithm (D_2 wavelet).