Electrocardiogram (ECG) Signal Modeling and Noise Reduction Using Wavelet Neural Networks[*]

Suranai Poungponsri, Xiao-Hua Yu

Department of Electrical Engineering
California Polytechnic State University
San Luis Obispo, CA 93407, USA

Abstract - **Electrocardiogram (ECG) signal has been widely used in cardiac pathology to detect heart disease. In this paper, wavelet neural network (WNN) is studied for ECG signal modeling and noise reduction. WNN combines the multi-resolution nature of wavelets and the adaptive learning ability of artificial neural networks, and is trained by a hybrid algorithm that includes the Adaptive Diversity Learning Particle Swarm Optimization (ADLPSO) and the gradient descent optimization. Computer simulation results demonstrate this proposed approach can successfully model the ECG signal and remove high-frequency noise.**

Index Terms – Wavelet neural networks, ECG signal, particle swarm optimization.

I. INTRODUCTION

The electrocardiogram (ECG) signal is generated by the rhythmic contractions of the heart. It represents the electrical activity of the heart muscles, and is usually measured by the electrodes placed on body surface. Clinical data show that ECG signal is very effective to detect heart disease. A typical one cardiac cycle ECG waveform which consists of a P-wave, a QRS-complex, and a T-wave, is shown in Fig. 1.
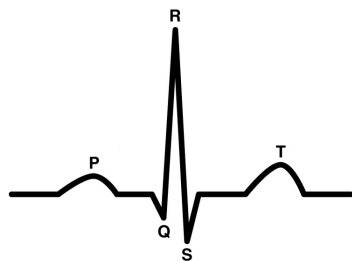

Fig. 1. A typical ECG signal

It should be noticed that even though the ECG signals from different patients have the similar forms, the ECG of each individual patient is different. To make appropriate medical diagnosis, doctors often need to compare the ECG signal with the patient's own individual record. Therefore, modeling the ECG signal for each patient becomes very important in clinic practice. In addition, ECG signal is often corrupted with noise, which makes an accurate diagnosis very difficult.

ECG noise removal is complicated due to the time-varying nature of ECG signals. The traditional approach to remove high frequency noise from ECG signal is to employ a low-pass filter [9]. However, the cut-off frequency is difficult to determine and it may introduce some additional artifacts to the signal, especially on the QRS wave. Other filtering techniques that have been proposed concentrate on the design of hybrid filters and filter banks, such as [10].

The concept of Wavelet Neural Network (WNN) was inspired by both the technologies of wavelet decomposition and neural networks. In standard neural networks, the input-output mapping is approximated by the superposition of sigmoid functions; while in WNN, this relationship is approximated by the superposition of a series of wavelet functions. WNN combines the multi-resolution nature of wavelets and the adaptive learning ability of neural networks, and has found many applications in time series prediction, function approximation and fault diagnosis.

In this research, WNN is studied for ECG signal modeling and noise reduction. The network is trained by a hybrid algorithm with ADLPSO (Adaptive Diversity Learning Particle Swarm Optimization) and gradient descent optimization. Computer simulation results demonstrate this proposed approach can successfully model the ECG signal and remove high-frequency noise.

II. THE WAVELET NEURAL NETWORK AND LEARNING ALGORITHM

In wavelet transform, various wavelets are generated from a single basic wavelet $\psi(t)$ known as the mother wavelet. This is done by introducing the scale (or dilation) factor, s; and translation (or shift) factor, τ. The shifted and dilated versions of the mother wavelet can be expressed as:

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t-\tau}{s}\right) \tag{1}$$

The wavelet transform of a signal $x(t)$ with mother wavelet function $\psi(t)$ can be written as:

$$T(s,\tau) = \frac{1}{\sqrt{s}} \int_{-\infty}^{\infty} x(t)\psi^*\left(\frac{t-\tau}{s}\right) dt \tag{2}$$

The asterisk indicates that the complex conjugate of the wavelet function is used in the transform. In fact, wavelet transform can be thought of as the cross-correlation of a signal with a set of wavelets of various "widths."

A typical Wavelet Neural Network (WNN) contains an input layer, an output layer, and a hidden layer. It employs a set of wavelets as the activation functions of the hidden neurons. Fig. 2 shows the structure of a WNN with multiple inputs x_1, x_2, …, x_n, where Ψ_i ($i = 1, 2, …, N$) represents the wavelet function of each hidden neuron; and $\bar{g}$ is the bias of the output neuron.
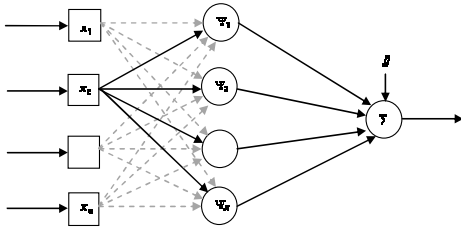

Fig. 2. The Wavelet Neural Network

For this application, we choose the mother wavelet function to be the following:

$$\psi(x) = -x\, e^{-\frac{1}{2}x^2} \qquad (3)$$

The plot of the mother wavelet function can be found in Fig. 3, which yields a similar shape as the ECG signal shown in Fig. 1.
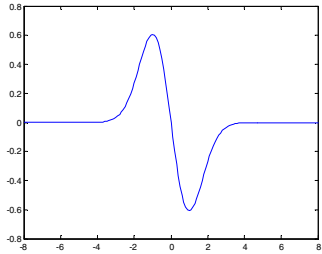

Fig. 3. The Mother Wavelet Function

The Particle Swarm Optimization (PSO) is a population based optimization method first proposed by Kennedy and Eberhart ([1]) in 1995. PSO is a multi-agent parallel search technique, inspired by the social behavior of bird flocking or fish schooling. It shares many similarities with evolutionary computation techniques such as Genetic Algorithm (GA). Both of them are initialized with a population of random solutions; then both algorithms search for optima, and update their solutions in each generation. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solution is the optimum particle of the current generation that flies through the multi-dimensional search space. Each particle keeps track of its coordinates (represented by a position vector z_i) in the space. If a best solution (among its neighborhood) can be found based on a preset fitness criterion, it is recorded as p_i. When a particle takes all the population as its topological neighbors, the best neighbor solution becomes a global best and is called p_g.

In the particle swarm optimization algorithm, the velocity v of particle i moving toward its local p_i and global optimal locations is updated at each training step. It is weighted by the two terms (ϕ_1 and ϕ_2) that are randomly distributed in [0, 1] with two separate constant coefficients c_1 and c_2 (in general, the range of c_1 and c_2 is [0, 4]). That is,

$$v_i(k+1) =$$
$$\chi\left[\alpha v_i(k) + c_1\phi_1\big(p_i(k) - z_i(k)\big) + c_2\phi_2\big(p_g(k) - z_i(k)\big)\right] \qquad (4)$$

where $i = 1, 2, …, N_T$ and N_T is the total number of particles (i.e., population size); χ and α are two real constant coefficients; and k is the index of generation. The parameter χ controls the magnitude of v, whereas the inertial weight α weights the magnitude of the old velocity $v_i(k)$ in the calculation of the new velocity $v_i(k+1)$. Based on the updated velocity, each particle changes its position according to the following equation:

$$z_i(k+1) = z_i(k) + v_i(k+1) \qquad (5)$$

The idea of Adaptive Diversity Learning PSO (ADLPSO) was introduced ([2]) to avoid the tendency of "localized cluster" in the PSO algorithm. In PSO algorithm, when a local optimum is found by one particle, the other particles will be drawn toward it. In other words, all particles may tend to stay at the same local optimum without much chance to escape. The basic idea of Adaptive Diversity Learning (ADL) is to allow some of the particles explore other areas in the solution space for possible better solutions.

ADL is actually integrated as an additional part in PSO. In each generation, the performances of all particles are evaluated and ranked by a fitness rule. The particles are then divided into two groups. The first group contains most of the particles (usually about 70% of the total population) with "good" performance while the second group contains a relatively smaller portion of particles (usually about 30% of the total population) with the "not-so-good" performance. All the particles in group-one are updated by the basic PSO algorithm (using Eq. (4) and (5)); the particles in group-two, however, are enhanced by a diversity model (ADL).

Let the total number of particles of the second group (i.e., the one updated by ADL) be denoted by N_s; and note that $N_s \le N_T$. According to [2], the diversity model can be specified using a specific probability density function (PDF):

$$p(x) = \begin{cases} (1-q)\beta e^{\beta x} & x \le 0 \\ q\beta e^{-\beta x} & x > 0 \end{cases} \qquad (6)$$

Thus, a random diversity vector can be written as:

$$\tilde{d} = [d_1, d_2, \cdots, d_{N_s}] \tag{7}$$

Each element in the vector can be found as:

$$d_i = \begin{cases} \dfrac{1}{\beta} ln\left(\dfrac{r_i}{1-q_i}\right), & 0 < r_i \leq (1-q_i) \\[3mm] -\dfrac{1}{\beta}\left(\dfrac{1-r_i}{q_i}\right), & (1-q_i) \leq r_i < 1 \end{cases} \tag{8}$$

where $i = 1, 2, \ldots, N_s$; r_i is a real random number uniformly distributed between [0, 1]. Note that we search around the local area in both positive and negative direction. The adjustable parameter $q_i \in [0, 1]$ determines the probability of search direction. The larger the q_i, the higher search probability in positive direction. Another control parameter β determines the range of the diversification search. The larger the β, the smaller range of local search. If this local search (in the kth generation) is performed more than once, then β can be updated in each iteration:

$$\beta(k+1) = \frac{[\beta(k) - \beta(0)]}{T}(T-t) + \beta(0) \tag{9}$$

where t is the index of iteration and T is the maximum number of iterations (in the kth generation).

Similar to the basic PSO, ADL updates the particle positions using the following rule:

$$\tilde{z}(k+1) = \tilde{z}(k) + \tilde{d}(k+1) \tag{10}$$

where $\tilde{d}(k+1)$ can be calculated by Eq. (8).

ADLPSO is a very effective global search algorithm; however, it may not be able to find an accurate solution. On the other hand, gradient descent algorithm is capable of finding a precise solution within certain rage, but it may be easily trapped in local minimum. In this research, a hybrid algorithm ([2]) is employed to train WNN. A region in search space is located by ADLPSO first; then the gradient descent algorithm is used to fine-tune the optimum solution in this region. Note that in Fig. 2, for each wavelet function of a hidden neuron, two more parameters (i.e., the scale factor and the translation) must be considered (in addition to the weights of NN). Let the NN output (single output) be:

$$g(x) = \sum_{i=1}^{N} (w_i \Psi_i(x)) + \bar{g} \tag{11}$$

where w_i is the weight (connection) from the ith neuron in the hidden layer to the output; and N is the total number of neurons in the hidden layer. For a multivariate wavelet basis function, it can be calculated by the tensor product of single wavelet basis function as follows:

$$\Psi(\tilde{x}) = \prod_{i=1}^{n} \psi(x_i) \tag{12}$$

where n is the total number of inputs of the neural network. For a real mother wavelet function, substituting Eq. (1) into Eq. (10), we obtain:

$$g(\tilde{x}) = \sum_{i=1}^{N} (w_i \Psi_i(\tilde{x})) + \bar{g}$$
$$= \sum_{i=1}^{N} \left(w_i |\tilde{s}_i|^{-\frac{1}{2}} \psi\left(\frac{\tilde{x} - \tilde{\tau}_i}{\tilde{s}_i}\right) \right) + \bar{g} \tag{13}$$

where

$$\tilde{x} = (x_1, x_2, \cdots, x_n) \tag{14}$$
$$\tilde{\tau}_i = (\tau_{i,1}, \tau_{i,2}, \cdots, \tau_{i,n}) \tag{15}$$
$$\tilde{s}_i = (s_{i,1}, s_{i,2}, \cdots, s_{i,n}) \tag{16}$$

For simplicity, let

$$\psi'(\tilde{x}) = \frac{d}{d\tilde{x}} \psi(\tilde{x}) \tag{17}$$

$$\tilde{z}_i = \frac{\tilde{x}_k - \tilde{\tau}_i}{\tilde{s}_i} \tag{18}$$

Also, let J_k, the objective function of the kth training iteration, be the square error:

$$J_k = \frac{1}{2}(e_k)^2 \tag{19}$$

where

$$e_k = g_\theta(\tilde{x}) - y_k \tag{20}$$

Then by using chain rule, the gradient of each parameter can be calculated as:

$$\frac{dJ_k}{d\bar{g}} = e_k \tag{21}$$

$$\frac{dJ_k}{dw_i} = e_k \Psi'(\tilde{z}_i) \quad \text{(for multiple inputs)}$$

or

$$\frac{dJ_k}{dw_i} = e_k \psi'(\tilde{z}_i) \quad \text{(for single input)} \tag{22}$$

$$\frac{dJ_k}{d\tilde{\tau}} = -e_k w_i \left(\frac{1}{\tilde{s}_i}\right) \psi'(\tilde{z}_i) \tag{23}$$

$$\frac{dJ_k}{d\tilde{s}} = -e_k w_i \left(\frac{1}{\tilde{s}_i^2}\right)(\tilde{x}_k - \tilde{\tau}_i)\psi'(\tilde{z}_i) \tag{24}$$

Finally, each parameter is updated as:

$$\bar{g}(k+1) = \bar{g}(k) - \eta \frac{dJ_k}{d\bar{g}} \tag{25}$$

$$w_i(k+1) = w_i(k) - \eta \frac{dJ_k}{dw_i} \tag{26}$$

$$\tilde{\tau}(k+1) = \tilde{\tau}(k) - \eta \frac{dJ_k}{d\tilde{\tau}} \tag{27}$$

$$\widetilde{s}(k+1) = \widetilde{s}(k) - \eta \frac{dJ_k}{d\widetilde{s}} \qquad (28)$$

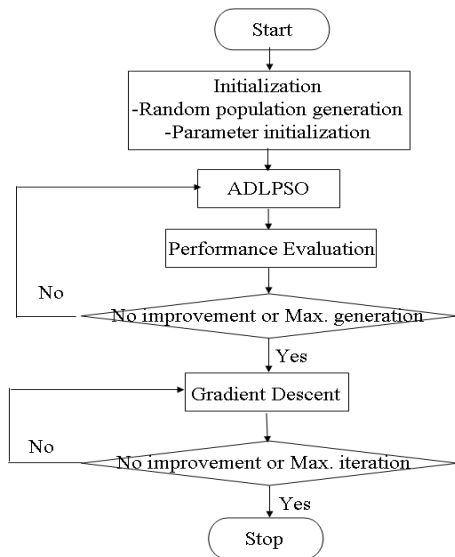where η is the learning rate. The flow chart of the overall training algorithm is shown in Fig. 4.

Fig. 4. The ADLPSO-Gradient Training Algorithm

III. SIMULATION RESULTS

In this section, the proposed wavelet neural network (trained by the adaptive diversity learning particle swarm optimization and the gradient descent algorithm) is applied to ECG signal modeling and noise reduction.

Both training and testing data are taken from [5]. White noise is added to the original (clean) ECG signal and the SNR (signal to noise ratio) of the noisy (unfiltered) signal is about 17.7dB. The neural network is trained using noisy ECG data as its input, with clean (noise-free) signal as the desired target. In the hybrid algorithm, the neural network is trained by ADLPSO with 400 iterations first, then trained by the gradient descent algorithm by 1600 iterations. The number of population is 20.

The simulation results are shown in Fig. 5. The ECG signals are shown in (a), with the clean (noise-free) signal on the top and the noisy signal on the bottom. The output of NN (in (b)) is smoother than the input, indicating that a well-trained WNN can effectively filter out high frequency noise. The SNR of the filtered signal (i.e., the output of NN) is approximately 21.1dB (or 4.1dB of improvement).

In general, for diagnostic purpose, the range of interest of ECG signal is about 0.05 Hz at the lower end of spectrum which allows for ST segments to be recorded and 40, 100, or 150 Hz in the higher end spectrum [8]. The choice for the frequency at the higher end depends on the patient's health and profile conditions as well as what type of diagnostic a doctor wants to capture. To compare the result of WNN with the traditional approach, a 501-tapped FIR low-pass filter with cut-off frequency at 90Hz is employed. The filtered signal is shown in (c). Obviously, the traditional filter can eliminate some, but not all the noise in the signal.
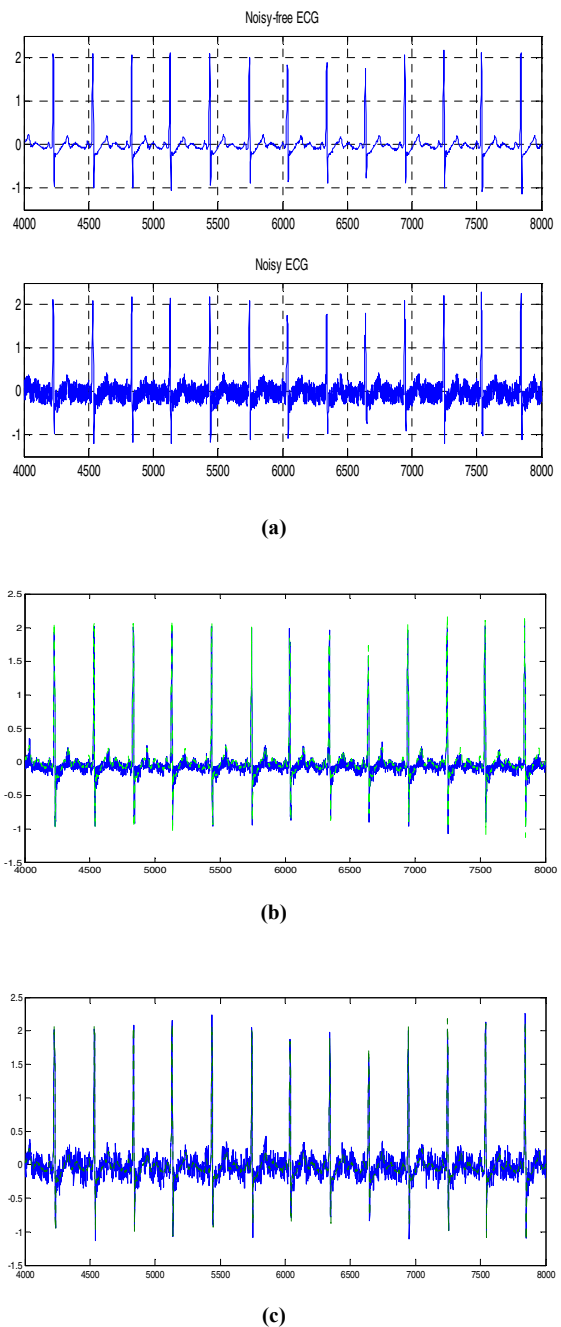
Fig. 5. Neural Network Testing Results
(a) The clean (Noise-free) signal (top) and the unfiltered noisy (bottom)
(b) The filtered ECG signal (by NN, solid line) and the clean signal (dotted line)
(c) The filtered ECG signal (by low-pass filter, solid line) and the clean signal (dotted line)

397

IV. Conclusions

An approach to study ECG signal based on wavelet neural network and adaptive diversity learning particle swarm optimization algorithm for network training is investigated in this paper. Computer simulation results show this approach is promising for ECG signal modeling and noise reduction. More tests will be conducted to further investigate its performance in the future.

References

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," Proc. IEEE Int. Conf. Neural Networks, pp.1942-1948, 1995.

[2] Y. Chen, B. Yang, and J. Dong, "Time-series prediction using a local linear wavelet neural network," *Neurocomputing*, vol. 69, pp.449-465, 2006.

[3] Q. Zhang, and A. Benveniste, "Wavelet networks", *IEEE Trans. Neural Network*, Vol. 3, No. 6, p.889-898, 1992

[4] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed., Prentice Hall, 1999.

[5] PhysioBank, *The MIT-BIH noise stress test database*, http://www.physionet.org/physiobank/database/nstdb/.

[6] PhysioBank, *MIT-BIH arrhythmia database*, http://www.physionet.org/physiobank/database/mitdb/.

[7] G. Moody, *WFDB applications guide*, http://www.physionet.org/physiotools/wag/nst-1.htm#sect4.

[8] Wikipedia, "Electrocardiogram", http://en.wikipedia.org/wiki/ECG.

[9] T. Slonim, M. Slonim, E. Ovsyscher, "The use of simple FIR filters for filtering of ECG signals and a new method for post-filter signal reconstruction", Proc. Computers in Cardiology, 1993.

[10]V. Afonso, W. Tompkins, et al., "Filter bank-based processing of the stress ECG," IEEE 17th Annual Conference on Engineering in Medicine and Biology Society, 1995.