Neural-Network-Based Adaptive Matched Filtering for *QRS* Detection

Qiuzhen Xue, *Member, IEEE,* Yu Hen Hu, *Senior Member, IEEE,* and Willis J. Tompkins, *Fellow, IEEE*

Abstract—**We have developed an adaptive matched filtering algorithm based upon an artificial neural network (ANN) for *QRS* detection. We use an ANN adaptive whitening filter to model the lower frequencies of the ECG which are inherently nonlinear and nonstationary. The residual signal which contains mostly higher frequency *QRS* complex energy is then passed through a linear matched filter to detect the location of the QRS complex. We developed an algorithm to adaptively update the matched filter template from the detected *QRS* complex in the ECG signal itself so that the template can be customized to an individual subject. This ANN whitening filter is very effective at removing the time-varying, nonlinear noise characteristic of ECG signals. Using this novel approach, the detection rate for a very noisy patient record in the MIT/BIH arrhythmia database is 99.5%, which compares favorably to the 97.5% obtained using a linear adaptive whitening filter and the 96.5% achieved with a bandpass filtering method.**

Introduction

QRS detection is the most important task in ECG signal analysis systems. After the *QRS* complex has been identified, the heart rate may be calculated, the ST-segment may be examined for evidence of ischemia, or the waveform may be classified as normal or abnormal. A number of *QRS* detectors have been designed which work well in the presence of moderate noise [1]–[3]. These detectors generally include a bandpass filter with a center frequency in the range of 10–17 Hz. After passing through the bandpass filter the signal may be squared and averaged over a number of samples to give an estimate of the local energy in the passband which is then used as a detection statistic. These techniques mainly suffer from two problems: 1) the signal passband of the *QRS* complex is different for different subjects and even for different beats of the same subject, 2) the noise and *QRS* complex passbands overlap. A matched filter can maximize the signal-to-noise ratio for detection of a known signal in noise. However, the design of an optimal matched filter requires knowledge of both the signal and the correlation statistics of the noise. The nonstationary nature of the signal and noise in an ECG represents an obstacle in application of matched filtering to *QRS* detection. We previously developed a linear adaptive matched filter for detection of *QRS* complexes in extreme noise [4]. This filter attempts to adjust itself to compensate for changing signal shapes and noise conditions. In the presence of motion artifact, its performance is better than our previously developed *QRS* detection algorithm based on bandpass filtering and local energy estimation [2].

The ECG, however, is a nonlinear signal generated from a nonlinear system—the human body. It is difficult to adapt to a nonlinear signal using a linear model. In other words, the ECG signal cannot be whitened much by a linear adaptive filter, hence the performance of the matched filter was not optimal. Since artificial neural networks are inherently nonlinear models, ANN-based filtering methods are potentially useful for signals with inherent nonlinearity. Unlike our approach, previous applications of ANN models in biomedical signal processing are mostly pattern recognition oriented [5]–[9].

In this study, we added a hidden layer with nonlinear units to our previously designed linear combiner, which is another name for a FIR linear adaptive filter. By introducing nonlinear hidden units, we are able to model much more complicated nonlinear signals. An ANN model with a nonlinear hidden layer is theoretically able to approximate any function that arises in the real world [10]. In practical use, this means that an ANN model can adapt to a nonlinear signal much better than can linear models, although this approximation may be limited because of restrictions on the number of units, as well as the use of different learning strategies. In our ANN-based filter there is a single output unit, while pattern recognition tasks usually have more output units. For our signal modeling application, the number of input units and hidden units are flexible, unlike other applications where the number of input units is fixed by the nature of the input signal, portion of signal, or image. In our case, the number of input units corresponds to the filter order as in the example of the linear adaptive filter. If the order is too small, we cannot obtain good modeling results, while if the order is too large, the filter will not have good transient properties (i.e., we cannot model the nonstationary part closely). However in *QRS* detection, we hope that the residual noise signal has more *QRS* information, which means that the model should predict non-*QRS* portions of the signal better than the *QRS* complex itself. Thus, we use a relatively large number of input units. Choice of the

Manuscript received February 6, 1991; revised September 17, 1991.

Q. Z. Xue was with the Department of Electrical and Computer Engineering, University of Wisconsin, Madison, WI 53706. He is now with Miles, Inc., Elkhart, IN 46515.

Y. H. Hu and W. J. Tompkins are with the Department of Electrical and Computer Engineering, University of Wisconsin, Madison, WI 53706.

IEEE Log Number 9106400.

number of hidden units is a common problem for both recognition and filtering tasks. For a point-by-point modeling task, the learning speed is important, which suggests a small number of hidden units. Use of too few hidden units, however, restricts adaptation to complicated nonlinearities.

Theory

In the signal detection problem, the objective is to detect the presence of a signal $s(t)$ in the received signal $x(t)$ which is contaminated with additive noise $n(t)$:

$$x(t) = s(t) + n(t). \tag{1}$$

For ECG beat detection, the signal $s(t)$ represents the *QRS* complex, and the noise $n(t)$ represents all other components of the ECG signal including the P and T waves, additive instrumentation noise, and time-varying electromyographic noise.

Matched Filter and Whitening Filter

In general, we have some *a priori* knowledge of the signal of interest such as a typical template of the *QRS* complex. However, the actual shape of such a template is different for different patients and changes with time in the same patient. Given such prior knowledge, our approach for detecting its presence in an ongoing ECG is based upon a matched filter. A matched filter can be realized using a finite impulse response (FIR) digital filter which has an impulse response $h(t) = s(t - t_0)$ such that the output signal-to-noise ratio when $x(t)$ is applied to the input is maximized.

$$r_0 = \frac{|y_s(t_0)|}{\sqrt{E\{y_n^2\}}} \tag{2}$$

In this equation, the signal output $y_s(t_0) = s(t) * h(t)$ is the result of convolution of the signal with the filter response. Similarly, $y_n(t) = n(t) * h(t)$ is the noise output.

The matched filter is an optimal detector if $n(t)$ is a white noise random process. Unfortunately this is hardly the case in the *QRS* detection problem. In particular, the background noise includes portions of the ECG signal such as the P and T waves, instrument noise, and electromyographic noise. These noise components are often correlated (i.e., they are colored noise) and time varying. To apply the matched filter detection scheme, a preprocessing procedure called a whitening filter is performed to remove the correlation components in the noise. Among many alternative whitening filter strategies, a linear adaptive autoregressive (AR) modeling technique is very suitable for real-time processing of ECG signals. In this model, we assume that the data sampled from the colored noise process arising in *QRS* detection at time t can be predicted by a linear combination of previous q data samples $\{y_{t-i} | i = 1, \cdots, q\}$. In other words,

$$n_t = \sum_{i=1}^{q} u_i n_{t-1} + \epsilon_t \tag{3}$$

where $\{u_i | i = 1, \cdots, q\}$ is called the set of AR parameters, and ϵ_t is the modeling error, which gradually approximates white noise if the model is correct. Also we observe that the *QRS* complex waveform, which exists only in a very short time duration within each period of the ECG signal, consists of mainly higher frequency components. Hence, by selecting the proper length q, the whitening filter can be adjusted so that it does *not* predict the *QRS* complex $s(t)$. Therefore when the received signal $x(t)$ is applied to the whitening filter, a large prediction error signals the presence of a desired *QRS* complex. In other words, after we apply the signal $x(t)$ to the whitening filter, its output is

$$\begin{aligned} y(t) &= x(t) - \sum_{i=1}^{q} u_i x_{t-1} \\ &= s(t) - \sum_{i=1}^{q} u_i s_{t-1} + \epsilon_t = s_I(t) + \epsilon_t \end{aligned} \tag{4}$$

where $s_I(t)$ is the distorted signal after passing through the whitening filter. As long as the observation window length q is kept small enough so that the higher frequency components in $s(t)$ are not modeled by the set of AR parameters $\{u_i | i = 1, \cdots, q\}$, the detection of $s_I(t)$ is equivalent to the detection of $s(t)$.

Adaptive Linear Whitening Filter

In order to cope with the time varying nature of the background noise in *QRS* detection, we chose to use an adaptive least mean square (LMS) algorithm [12] to compute the set of AR coefficients. The LMS algorithm uses the gradient search method which updates the filter coefficients for each incoming datum $\boldsymbol{x}_t$ as

$$\boldsymbol{u}_{t+1} = \boldsymbol{u}_t + 2\mu e_t \boldsymbol{x}_t \tag{5}$$

where μ is the step size which controls the search step, and

$$e_t = \boldsymbol{x}_t^H \boldsymbol{u}_t \tag{6}$$

is the instantaneous error of the whitening filter. To reduce the magnitude of random fluctuation, we also add a *momentum* term which leads to the momentum-LMS algorithm

$$\boldsymbol{u}_{t+1} = \boldsymbol{u}_t + 2\mu e_t \boldsymbol{x}_t + \alpha[\boldsymbol{u}_t - \boldsymbol{u}_{t-1}]. \tag{7}$$

Fig. 1 is a diagram of this adaptive whitening filter.

We reported earlier the use of a linear adaptive whitening filter to remove colored noise [6]. Due to the nonlinearity inherent in the background noise processes the effectiveness of such a linear model to perform *QRS* complex detection may still be very limited. What is needed is a method to accurately model the nonlinear relationship that exists among the samples of the background noise processes. Motivated by this observation, we investigated a neural-network-based nonlinear whitening filter in this study.

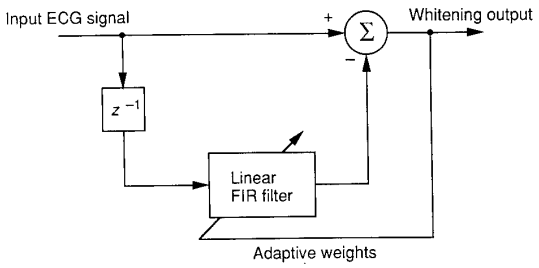

Fig. 1. Block diagram of linear adaptive noise removal filter.

Neural-Network-Based Nonlinear Whitening Filter

Our approach for dealing with the inherent nonlinearity of the ECG signal is to replace the linear adaptive whitening filter having the structure shown in Fig. 2(a) with a neural-network-based nonlinear adaptive filter illustrated in Fig. 2(b). This structure can be derived from the linear adaptive whitening filter by adding a nonlinear *hidden layer* which consists of a number of nonlinear processing units. We call these hidden units as they produce only intermediate results which then are combined to form the actual output. Each of the hidden units produces a nonlinear intermediate result

$$z_i = f\left(\sum_{j=1}^{M} w_{ij} y_{t-j} + b_j\right) \tag{8}$$

where $f(\ )$ is a sigmoid function defined as

$$f(x) = \frac{1}{1 + e^{-x/T}}. \tag{9}$$

The value of T, called *temperature*, controls the nonlinearity of the function. The smaller the value of T, the more nonlinear the function. The w_{ij}'s are the weights which connect the input units to the hidden units, and the b_j's are bias terms. The output of the nonlinear whitening filter is a linear combination of the outputs of the hidden units:

$$y_0 = y_t - \hat{y} \tag{10}$$

$$y_t - \sum_{i=1}^{q} u_i z_{t-i} = y_t - \sum_{i=1}^{q} u_i f\left(\sum_{j=1}^{M} w_{ij} y_{t-j} + b_j\right). \tag{11}$$

The most widely-used algorithm to update the weights in the hidden layer is the generalized delta rule, and the neural network so formed is called a back-propagation network. It is an extension of the LMS algorithm for nonlinear units, since both techniques use a gradient search based algorithm. The least-mean-square error of this filter is

$$e = E\left\{\left(y_t - \sum_{i=1}^{q} u_i f\left(\sum_{j=1}^{M} w_{ij} y_{t-j} + b_j\right)\right)^2\right\}. \tag{12}$$

The hidden layer weights are updated as

$$w_{t+1}^j = w_t^j + 2\mu\delta_j x_t + \alpha(w_t^j - w_{t-1}^j)$$
$$j = 1, \cdots, q \tag{13}$$

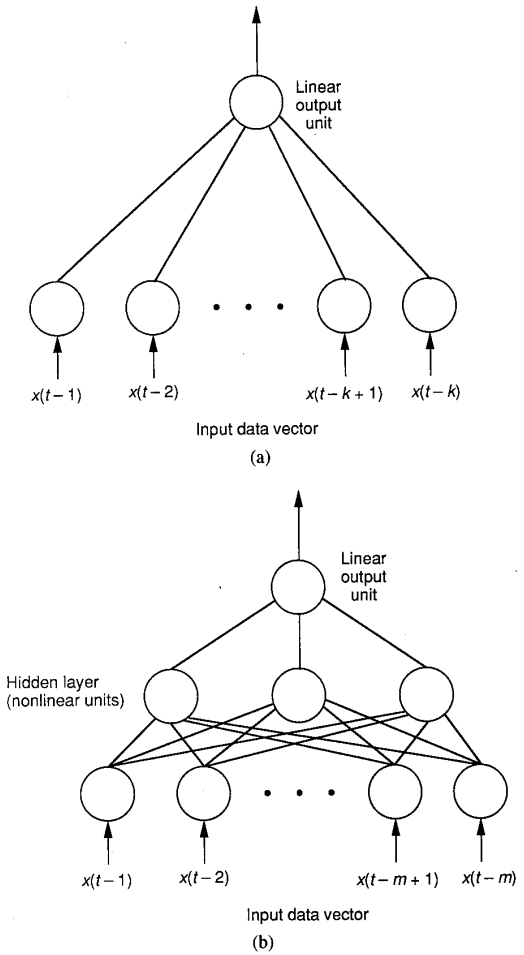

Fig. 2. Adaptive filter structure. (a) Single-layer linear. (b) Multilayer nonlinear.

where $w_t^j = (w_{1j}, w_{2j}, \cdots, w_{mj})^t$ is the weight vector connected from the input units to the jth hidden unit; $\delta_j = yh_t^j(1 - yh_t^j)n_t a_j$ is the error term passed back from the upper layer; α is the step size of a momentum term. Since the output is a linear combination of the hidden layer outputs, its weights $\boldsymbol{u}_t$ are updated according to (7) in the same way as the linear whitening filter. Fig. 3 is a block diagram of the adaptive nonlinear whitening filter.

Our nonlinear adaptive filter structure uses a single linear output unit. This is because our goal is to model the nonlinearity of the underlying ECG signal which has a dynamic range not limited to the range between 0 and 1 and which cannot be dynamically scaled easily. In the Appendix, we provide convergence analyses of the ANN-based adaptive filter.

Template Selection and Filtering

A template is needed to do matched filtering. The first step is to select a template, and the second step is to filter it according to (11) where a whitening filter is applied to both the input signal and the template since the back-

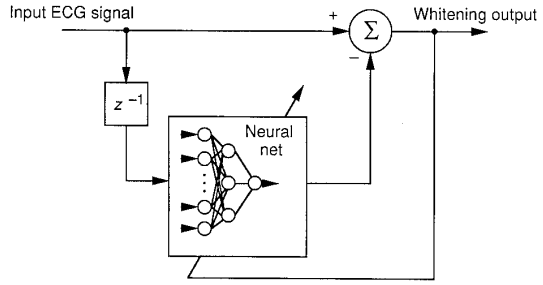

Fig. 3. Block diagram of adaptive noise removal filter based on artificial neural network.

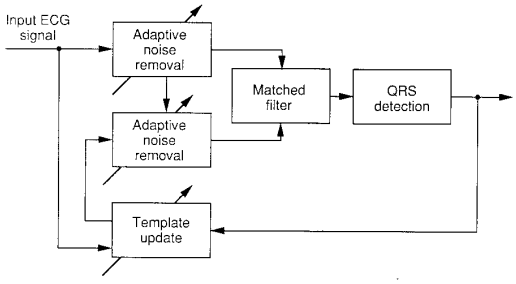

Fig. 4. Block diagram of ANN-based adaptive matched filter for detection of *QRS* complexes in an ECG signal.

ground noise is not white noise. In selecting the template, our previous method used a fixed template [4]. However generally the ongoing *QRS* complexes of the subject change with time. Since a fixed template cannot respond to changing information, this nonadaptability influences the performance of the matched filter. Thus, we developed a neural-network-based recognition method to update the template dynamically. In a preprocessing period, we train a small neural net model using 20–30 *QRS* complexes from the ongoing signal. The neural network model used is a three-layer feedforward back-propagation model, which has 70 input units to cover the *QRS* complex, ten hidden units, and ten output units. After the learning process is finished, the neural network coefficients are stored for determining a new template during the processing. At the beginning of the processing, four recognized normal *QRS* complexes are kept in the template bank, and the newest template is the average of these four complexes. The template vector is

$$\boldsymbol{QRS} = \frac{1}{4} \sum_{i=1}^{4} QRS_i(t). \tag{14}$$

As a result of using a matched filter to indicate the presence of the *QRS* complexes, the corresponding most-recently-recognized *QRS* complex of the original signal (not the output of the matched filter since this output is a distorted version of the signal just used for detecting the presence of the *QRS* complex) is sent to the neural network recognition model. The model determines if the new *QRS* complex is one which should be put into the template bank. In this way, we obtain new information about the ongoing signal while at the same time preventing abnormal waveforms from entering the template bank.

The template is filtered by the adaptive whitening filter synchronously with the filtering of the input signal. The whitened *QRS* template is

$$WQRS(t)_k = QRS(t)_k - \sum_{i=1}^{q} u_i f\left(\sum_{j=1}^{M} w_{ij} QRS(t)_{k-j} + b_j \right) \tag{15}$$

where $k = 1, \cdots, L - M$; L is the length of the *QRS* template vector; and M is the number of input units in the model.

Neural-Network-Based Adaptive Matched Filtering

After both the whitened signal and the whitened template have been obtained, matched filtering is performed. The output of the matched filter is

$$y_m(t) = \sum_{i=1}^{L} WQRS_i y_w(t - i) \tag{16}$$

where *WQRS* is the whitened template and $y_w(t)$ is the output of the whitening filter of the original signal.

Fig. 4 is the complete diagram of the neural-network-based adaptive matched filter. The input ECG signals first go through an adaptive whitening filter having as a key part the neural-net-based nonlinear adaptive noise removal filter. The input layer of the neural net model gets the data vector from the ECG signal. In this case, the input pattern is the data vector

$$y(t - 1) = \{y_{t-q}, y_{t-q+1}, \cdots, y_{t-1}\}^T$$

and the target pattern is y_t. The output of the model is the estimation of y_t. In each epoch, the input and target patterns are updated by shifting the time window one step forward, and the weights of the model are updated by the generalized delta rule with the error propagated backwards. If the original signal is approximately stationary, the weights will not change much after they have converged to certain values, which approximate the local minima on an error surface. If the original signal is nonstationary, the weights will be updated throughout a relatively large range through the data processing period. The template bank is updated only when a new *QRS* complex is identified and this new *QRS* complex is passed to the recognition part. Otherwise this template bank remains unchanged. The averaged template is filtered by the same adaptive noise removal filter from the neural net model synchronously with the signal filtering process. The output of the template filter forms a matched filter which filters the output of the data whitening filter. Finally, the data output from the matched filter is sent to the *QRS* detection circuit, which includes squaring, moving averaging, and threshold checking as described in [2].

Implementation and Results

Filtering Response to Linear and Nonlinear System Signals

First, we investigated the behavior of neural-network-based adaptive filtering applied to signals from both linear

and nonlinear systems. Then we compared these results with those of linear adaptive filtering. We generated the linear signals by filtering a set of normal distributed random signals $\{x(n)\}$ using an FIR filter $H(z)$. The output of this linear system is

$$y(n) = \sum_{i=0}^{N} h(i)x(n-i) \qquad n = 1, 2, \cdots \tag{17}$$

Our first nonlinear system signals were formed with

$$z_1(n) = [(y(n)]^3 \qquad n = 1, 2, \cdots \tag{18}$$

and our second nonlinear system signals were formed with

$$z_2(n) = k_1 e^b y(n) \qquad n = 1, 2, \cdots \tag{19}$$

We applied all three sets of signals to both linear and nonlinear adaptive filters and obtained the residual noise of each filter in order to study the learning ability and the performance of the filters. The smaller the residual noise, the better is the tracking ability of the filter. We also computed the autocorrelation function of the residual noise to see the orthogonality of the residual noise; that is, to see how "white" it was. The residual noise should approximate white noise if the filtering was performed appropriately.

Fig. 5 shows the square of the residual noise $e = n(t)^2$ of both filters when linear signals $y(n)$ were presented. You can see that the square errors of the nonlinear adaptive filter (solid line) and the linear adaptive filter (dashed line) are comparable. Fig. 6 shows the autocorrelation functions (ACF) of the residual noise from both approaches. They are both damped rapidly which means that the residual noise has good orthogonality (both cases approximate white noise). Although the performance of the linear and nonlinear adaptive filters are similar for this linear signal case, we would choose the linear filter for its simplicity, since the nonlinear filter does not provide an advantage in this case.

Figs. 7 and 8 show the same comparison but for nonlinear signals $z_2(n)$. We can see from Fig. 7 that the residual error of the ANN-based filter (solid line) is much smaller than that of the linear adaptive filter (dashed line), which means that the ANN filter has better learning ability than the linear filter for these nonlinear signals. The ACF plot of Fig. 8 shows that the ACF of residual noise from the ANN filter damped faster than that of the linear filter, which means that the residual noise of the ANN filter more closely approximates white noise than does the linear filter. We conclude that the ANN filter has better performance for nonlinear signals since the nonlinearity of the filter permits it to rapidly track these signals with a better approximation to the noise.

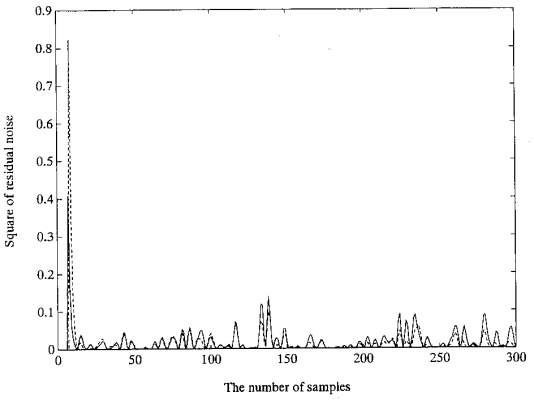

Fig. 5. Residual noise of ANN-based and linear adaptive filters when the input signal is from a linear system. Solid line: ANN-based filter. Dashed line: linear filter.

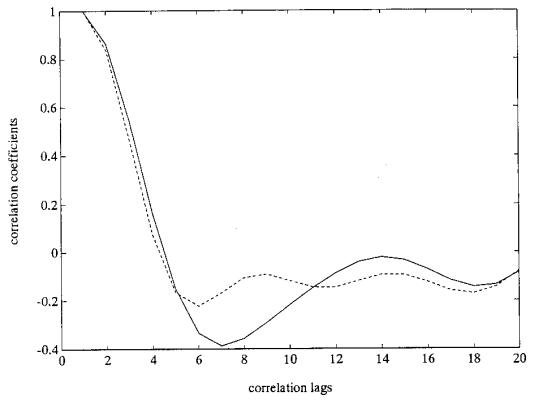

Fig. 6. Autocorrelation function (ACF) of the residual noise in Fig. 5.

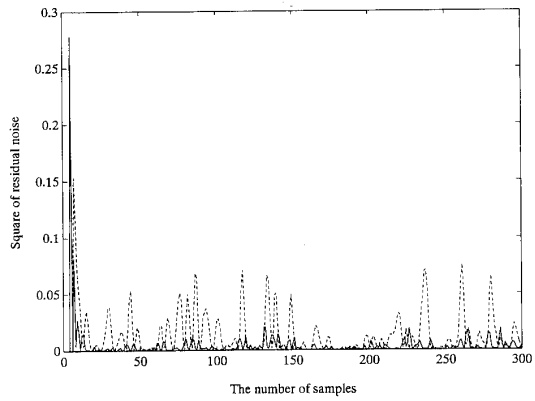

Fig. 7. Residual noise of ANN-based and linear adaptive filters when the input signal is from a nonlinear system. Solid line: ANN-based filter. Dashed line: linear filter.

ANN Adaptive Matched Filtering Response for Noisy ECG Signals

We applied ECG signals from the MIT/BIH arrhythmia database provided on compact disc (CD-ROM) [14] directly to the input of the ANN adaptive matched filter without any preprocessing. We obtained templates using (14). Fig. 9 shows the results of ANN adaptive matched filtering of a very noise ECG signal section from Record 105 of the MIT/BIH database. Fig. 9(b) demonstrates that the whitening filter removes baseline drift from the ECG and enhances the peaks of the *QRS* complexes while suppressing noise and *P* and *T* wave peaks. The *QRS* com-

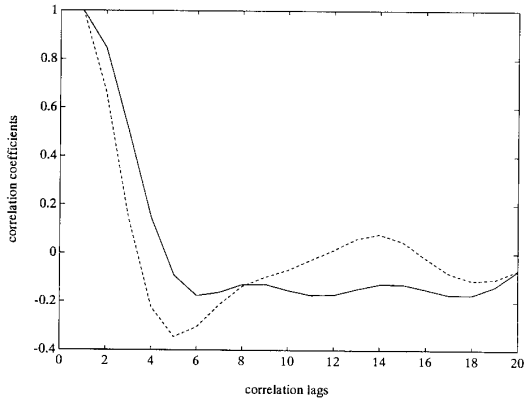

Fig. 8. Autocorrelation function (ACF) of the residual noise in Fig. 7.

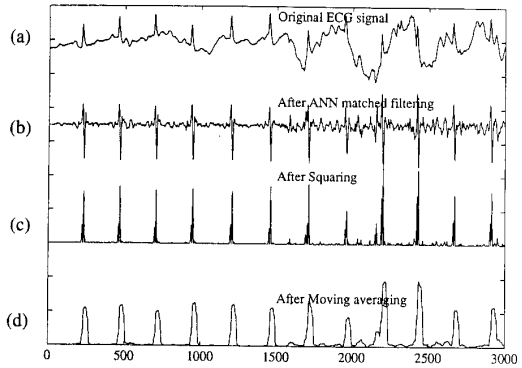

Fig. 9. Results of ANN-based adaptive matched filtering of the ECG. (a) Original signal. (b) Output of ANN-based adaptive matched filter. (c) Output of squaring function. (d) Output of moving average process.

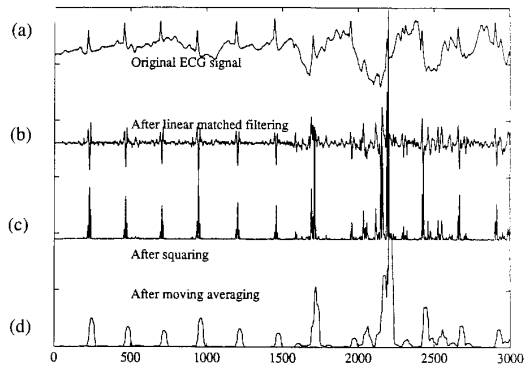

Fig. 10. Results of linear adaptive matched filtering of the ECG. (a) Original signal. (b) Output of linear adaptive matched filter. (c) Output of squaring function. (d) Output of moving average process.

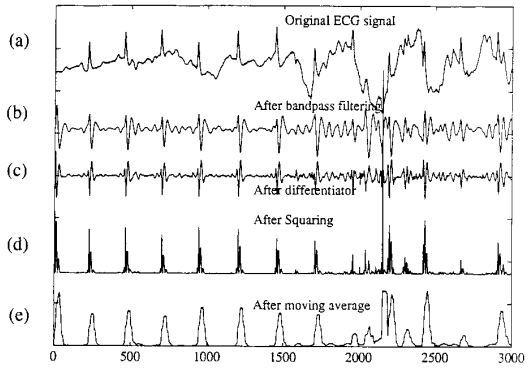

Fig. 11. Results of bandpass filtering of the ECG. (a) Original signal. (b) Output of bandpass filter. (c) Output of differentiator. (d) Output of squaring function. (e) Output of moving average process.

plexes can be easily detected from the output of the ANN adaptive matched filter, even without the additional processing steps of squaring and applying a moving average shown in Fig. 9(c) and (d), which are particularly important in other approaches [2]. In this experiment, we set the number of input units to 6, the number of hidden units to 3, the learning step size to 0.3, and the momentum to 0.5. We also developed methods to adaptively adjust the learning step size and to select the number of hidden units more systematically. These methods are discussed below.

For comparison, we plotted the results of previous methods based on a linear adaptive filter in Fig. 10 and on a bandpass filter in Fig. 11, both with the same input ECG. Unlike the ANN approach, further processing by squaring and applying a moving average is needed in both these approaches in order to reduce the *false positive* (FP) and *false negative* (FN) error rates.

In order to test the effectiveness of the ANN filtering more objectively, we computed the ACF of residual noise from both ANN and linear filtering. We selected a signal segment which does not include the *QRS* complex since it is very nonstationary and already "white." Fig. 13 shows ACF's for both ANN and linear filtering. We can see that the ACF of the residual noise from ANN filtering

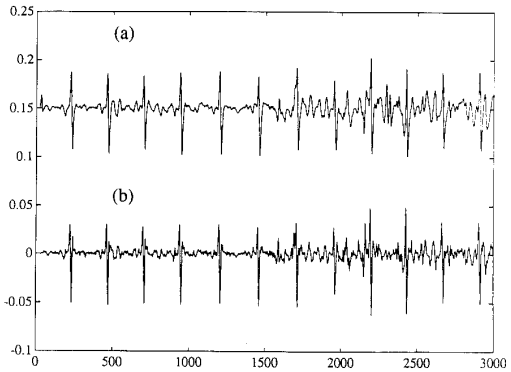

Fig. 12. Comparison of outputs of ANN-based matched filters. (a) Using fixed step size. (b) Using varying step size.

is damped more rapidly than that for linear filtering. This comparison shows that ANN filtering has a better whitening effect than linear filtering.

QRS Complex Detection Based on ANN Adaptive Matched Filtering

In most cases, a simple threshold method can be used to detect *QRS* complexes after ANN matched filtering.

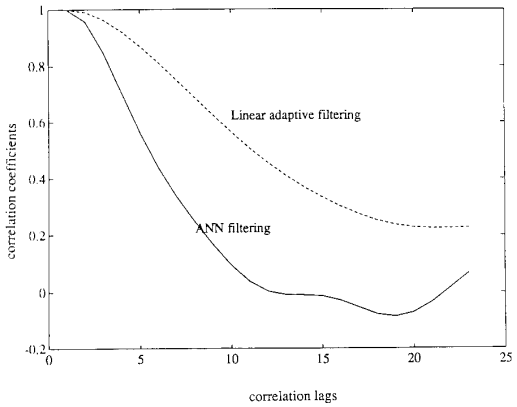

Fig. 13. Autocorrelation function (ACF) of the residual noise of the non-*QRS* components of the signal.

Further processing, however, can improve the detection. We applied squaring and moving average functions based on the methods in [2]. Fig. 9 shows the original ECG signal along with the results of ANN matched filtering, squaring, and moving averaging. We used a simple threshold method, instead of the more complicated double threshold method described in [2] since *QRS* complexes are enhanced more by this ANN filtering method than by other techniques. We also processed this particular portion of the signal using linear adaptive filtering and linear bandpass filtering. These results are plotted in Figs. 10 and 11. It is difficult to detect all *QRS* complexes by just using a simple threshold method for these linear filtering methods. Even using the more complicated double thresholding methods described in [2], linear filtering methods still have higher FP and FN rates than ANN filtering.

In Tables I and II, we list detection rates for the three filtering methods for Records 105 and 108 of the MIT/BIH database, which include 2572 and 1763 beats, respectively. We selected these two records because they are more noisy than others and caused more detection errors using linear filtering methods. In our experiments, the ANN-filtering method had only 10 FP and 4 FN beats for Record 105, which corresponds to a total failure rate $((10 + 4)/2572) \times 100 = 0.5\%$, while the failure rate was 3.46% for the bandpass filtering method and 2.4% for the linear adaptive filtering method. For Record 108, the ANN method had 25 FP and 16 FN beats or a 2.32% failure rate, while the failure rates of bandpass filtering and linear adaptive filtering were 12.54% and 4.42%, respectively. By comparing these results, we can see the significant improvement achieved by the ANN adaptive matched filtering method.

Effects of Input Window Length and Number of Hidden Units

The neural network structure used in *QRS* complex detection is determined by the input data window length, which specifies the number of input units, and the number of hidden layer neurons. Obviously, whenever the number of input or hidden units increases, the number of weights, and hence the computation complexity increases too making it more difficult to perform real-time processing. However, there are more subtle considerations than the computation cost in determining the structure of the neural network.

TABLE I
RESULTS OF EVALUATION OF THREE *QRS* DETECTION ALGORITHMS USING RECORD 105 OF THE MIT/BIH DATABASE

Filtering Methods	Total No. Beats	FP (Beats)	FN (Beats)	Failed Detection Beats	Failed Detection Rate (%)
ANN adaptive filtering	2572	10	4	14	0.50
Linear adaptive filtering	2572	40	22	62	2.40
Bandpass filtering	2572	67	22	89	3.46

TABLE II
RESULTS OF EVALUATION OF THREE *QRS* DETECTION ALGORITHMS USING RECORD 108 OF THE MIT/BIH DATABASE

Filtering Methods	Total No. Beats	FP (Beats)	FN (Beats)	Failed Detection Beats	Failed Detection Rate (%)
ANN adaptive filtering	1763	25	16	41	2.32
Linear adaptive filtering	1763	58	20	78	4.42
Bandpass filtering	1763	199	22	221	12.54

If the number of input units, or equivalently, the data observation window length, is too small, the adaptive whitening filter cannot capture the inherent nonlinear relations between the current data sample and samples outside the observation window, causing poor background noise removal. On the other hand, if the window length q is too large, the correlation between successive *QRS* complexes may be unexpectedly captured by the whitening filter causing unwanted cancellation of the signal (i.e., *QRS* complex). We conducted an experiment which varied the number of input units from 2 to 12 with the number of hidden units fixed at four. Fig. 14 shows the respective detection rates versus the number of input units. From the figure, we see that between 5 and 8 input units seems to be appropriate for the ECG signals studied.

The number of hidden units dictates the network's ability to perform nonlinear mapping between its input and output. Too few hidden units cannot accurately model the underlying nonlinear noise characteristics which cause excessive *QRS* detection errors. Too many hidden units, on the other hand, cause overfitting of the model, resulting in poor noise prediction. For the selection of the number of hidden units, we developed a method based on singular value decomposition (SVD) towards the output

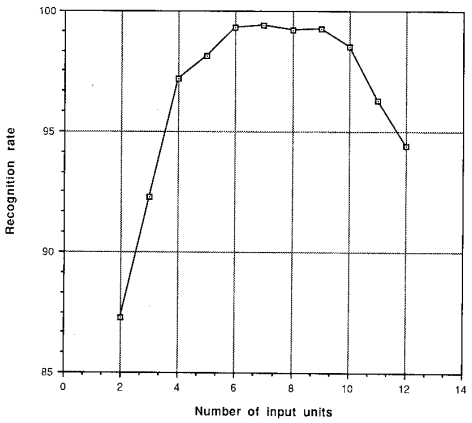

Fig. 14. *QRS* complex recognition rate as a function of the number of input units. The number of hidden units is fixed at 4.

covariance matrix of the hidden layer [13]. This approach is based on the following simple, yet important, observation. If the output of a neuron is linearly dependent on that of another neuron in the same layer, this neuron can be eliminated by properly adjusting the output weights of the other neuron without affecting the following layer neurons. For example, consider the case where the current neuron output is x_1 and another neuron in the same layer has an output x_2, such that $x_1 = ax_2$. Suppose that a neuron in the next layer contains a term $w_1x_1 + w_2x_2$. Clearly, this term can be replaced by $(aw_1 + w_2)x_2$. In other words, x_1 can be removed as it no longer appears in the latter expression. The number of independent neurons in the current hidden layer can be found by checking the rank of the autocorrelation matrix of the output hidden units in the current layer. This matrix will have rank deficiency if at least one hidden unit's output is linearly dependent on the rest of the hidden units. Although it is not very realistic to apply this method for every iteration in real-time processing, we used this method in preprocessing. If there are altogether K ECG signal samples in the preprocessing stage, that means that K training sample vectors $\boldsymbol{X}(n)$, $n = 1, 2, \cdots, K$ can be formed, and there are N hidden units. Let $\boldsymbol{y}(k)$ be the output vector of the hidden layer corresponding to the kth training sample vector. Define the hidden layer output matrix $\boldsymbol{Y}$ as

$$\boldsymbol{Y} = [\boldsymbol{y}(1)\boldsymbol{y}(2) \cdots \boldsymbol{y}(K)]^t \tag{20}$$

which is a $K \times N$ matrix. Then an $N \times N$ sample output correlation matrix of the current layer hidden units can be defined by

$$\boldsymbol{C}_{yy} = \frac{1}{K}\boldsymbol{Y}^t\boldsymbol{Y} = \frac{1}{K}\sum_{k=1}^{K}\boldsymbol{y}(k)\boldsymbol{y}^t(k). \tag{21}$$

If we perform an eigenvalue decomposition of the $\boldsymbol{C}_{yy}$ matrix, and write the result in two parts:

$$\boldsymbol{C}_{yy} = \boldsymbol{E}\Lambda\boldsymbol{E}^t = \boldsymbol{E}_1\Lambda_1\boldsymbol{E}_1^t + \boldsymbol{E}_2\Lambda_2\boldsymbol{E}_2^t \tag{22}$$

where $\Lambda = \text{diag}\{\lambda_1, \cdots, \lambda_p, \lambda_{p+1}, \cdots, \lambda_N\}$ consists of p larger eigenvalues, $\Lambda_1 = \text{diag}\{\lambda_1, \lambda_2, \cdots, \lambda_p\}$, and N-p much smaller eigenvalues, $\Lambda_2 = \text{diag}\{\lambda_{p+1}, \lambda_{p+2}, \cdots, \lambda_N\}$. The $\boldsymbol{E}$, $\boldsymbol{E}_1$, and $\boldsymbol{E}_2$ matrices, respectively, are the corresponding eigenvectors associated with Λ, Λ_1, and Λ_2. It is well known that the matrix $\boldsymbol{E}_1\Lambda_1$ is the optimal approximation of the $\boldsymbol{C}_{yy}$ matrix among all the $N \times N$ matrices, $\boldsymbol{C}$ having rank p. Based on the initial analysis, we then use p hidden units, instead of the original N hidden units for later processing. A test result of the relationship between the *QRS* recognition rate and the number of hidden units is plotted in Fig. 15. From this figure, we see that the acceptable range of p values is 3 to 5, which is consistent with what we found using the above formula. Note that there is no improvement in terms of detection rate when the number of hidden units is larger than 4.

There are other methods for determining the number of hidden units reported in the literature. For example, Siestma and Dow [15] have also demonstrated a manual pruning of a neural network by exploiting redundancy among the outputs of neurons. Rummelhart [16] has also proposed to use an additional cost function to force weights with small magnitudes to converge to zero so that they can be removed. This approach has since be refined by Weigend, Huberman, and Rumelhart [17], Chauvin [18], Lang and Hinton [19], Hanson and Pratt [20], and others.

Using Variant Step Size Based on Signal Energy

In adaptive filtering, the step size is important for adaptation speed. In most neural network applications, a fixed step size is used for updating the weights, as μ in (13). In those applications, all input patterns are prepared and normalized in advance, so that a fixed step size can be used with reasonable learning speed. In real-time filtering, however, the signal vector is sent in for every sampling interval, as in the case of ECG signal processing where input signal energy is different from vector to vector. As we know from the convergence analysis of ANN filtering in the Appendix, there are different upper bounds for hidden-layer and output-layer step sizes, while there is only one upper bound of step size in the linear adaptive filtering case. Here we discuss how to use variant step size instead of fixed step size for ANN filtering. Because of different upper bound values for output and hidden layer step sizes, we consider these two cases separately and use two step sizes, $\mu_0(n)$ for the output layer and $\mu_h(n)$ for the hidden layer.

Output Layer Step Size $\mu_0(n)$

From (A11) in the Appendix, we get the upper bound condition for

$$\mu_0(n) < \frac{\alpha + 1}{\lambda_{h\max}}$$

where $\lambda_{h\max}$ is the largest eigenvalue of the autocorrelation matrix $\boldsymbol{R}_h$ of the output of the hidden layer. And an upper bound on $\lambda_{h\max}$ is

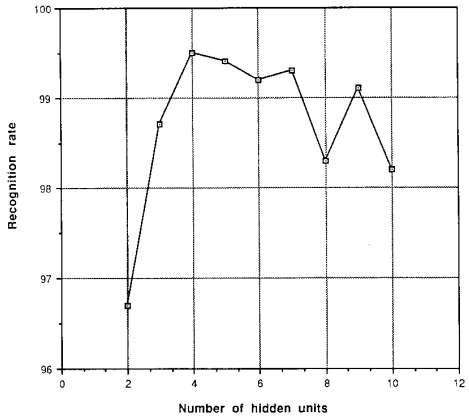

Fig. 15. *QRS* complex recognition rate as a function of the number of hidden units. The number of input units is fixed at 6.

$$\lambda_{h\max} < \sum_{k=1}^{N} \lambda_k = \text{trace } \boldsymbol{R}_h = M\gamma_{hh}(0) \tag{23}$$

where $\gamma_{hh}(0)$ is the hidden layer output signal power. Thus, we may replace the fixed step size μ by

$$\mu_0(n) = \frac{T(1+\alpha)}{\gamma_{hh}(0)} = \frac{T(1+\alpha)}{\sum_{i=1}^{N} y_{hi}^2} \tag{24}$$

where T is a positive value. We selected $T = 0.1$–0.5 for our experiment.

Hidden Layer Step Size $\mu_{hi}(n)$

From (A28), we obtain an upper bound for the ith hidden layer step size

$$\mu_{hi}(n) < \frac{\alpha + 1}{\hat{u}_j k_{1lj} \lambda_{x\max}} \qquad j = 1, 2, \cdots, N$$

where, $\hat{u}_j$ is the upper layer weight connected to the ith hidden layer, k_{1lj} is the tangent of the sigmoid function, which is a parameter of the segment linearized sigmoid function too, $\lambda_{x\max}$ is the largest eigenvalue of the autocorrelation matrix $\boldsymbol{R}_{xx}$ of the input signal. This condition is more complicated than that of the output layer, since more parameters are involved. However, we can still compute variant step sizes based on these known parameters. Replacing $\lambda_{x\max}$ by input signal energy, we can get a similar form of the hidden layer step size to that of (24):

$$\mu_{hj}(n) = \frac{T(1+\alpha)}{\hat{u}_j k_{1lj} \gamma_{xx}(0)} = \frac{T(1+\alpha)}{\hat{u}_j k_{1lj} \sum_{i=0}^{M-1} x_{n-i}^2} \tag{25}$$

Fig. 12(a) and (b) are the ANN matched filtering results for an ECG signal with fixed and variant step size respectively. In this example, we chose T to be 0.2 and α to be 0.3. The figure shows that the filter with variant step size has better noise removal ability than the one with a fixed step size. It also shows that *QRS* complexes from the variant step size method are sharper than those of the fixed step-size method.

Discussion

In this paper, we present an artificial-neural-network-based adaptive matched filter for *QRS* detection of very noisy ECG signals. There are several distinct features of this method.

1) We used the multilayer perceptron neural network structure for an adaptive whitening filter. Our experiments show that this nonlinear adaptive filter can model the inherently nonlinear ECG signal better than linear filtering techniques.
2) The *QRS* template used for matched filtering is updated by an ANN recognition algorithm, which provides better adaptation to signal changes than other matched filter techniques.
3) The number of hidden units is determined by our newly developed eigenvalue decomposition method. This method saves computation time and removes the redundancy introduced by having unnecessary hidden units.
4) The learning step size is varied with the incoming signal energy to improve the adaptability of the filter.

Using an ANN for signal filtering is a relatively new concept in signal recognition applications. This *QRS* detection application for very noisy ECG signals shows the potential of this nonlinear filtering approach, especially for inherently nonlinear biomedical signals. The potential lies in the nonlinear modeling of biomedical signals by ANN approaches, which is also one of our future research directions. There are many other structures of ANN's. We chose a multilayer perceptron with a back-propagation learning algorithm because it is most similar to the transversal adaptive filter with the LMS (least mean square) learning algorithm whose characteristics have been studied most thoroughly. The ANN-based nonlinear adaptive filters, however, are quite different from linear filters in many ways. We did analyses of ANN-based adaptive filters and list them in the Appendix. However, in order to grasp more details of these filters, we need more powerful analytical tools and methods. In any event, it is clear that combinations of recognition and filtering by ANN models will lead to new ways of biomedical signal processing.

Appendix

Convergence Characteristics of the Output Layer Weights

We first analyze the equation for updating the output layer weights as follows:

$$\boldsymbol{u}(n+1) = \boldsymbol{u}(n) + 2\mu e(n)\boldsymbol{y}_h(n) + \alpha[\boldsymbol{u}(n) - \boldsymbol{u}(n-1)]. \tag{A1}$$

We define the correlation matrix of the hidden layer out-

put as

$$\boldsymbol{R}_{hh} = E\{\boldsymbol{y}(n-1)\boldsymbol{y}^H(n-1)\}$$

and its orthonormal factorization form

$$\boldsymbol{R}_{hh} = \boldsymbol{Q}_h \Lambda \boldsymbol{Q}_h^T \tag{A2}$$

where $\Lambda = \text{diag}\,[\lambda_{h1}, \lambda_{h2}, \cdots, \lambda_{hN}]$ is a diagonal eigenvalue matrix of $\boldsymbol{R}_{hh}$ with $0 < \lambda_{h1} \le \lambda_{h2} \le \cdots \le \lambda_{hN}$. Matrix $\boldsymbol{Q}_h$ is the orthonormal matrix (i.e., $\boldsymbol{Q}_h\boldsymbol{Q}_h^T = \boldsymbol{I}$), and its column vectors are corresponding eigenvectors of eigenvalue λ_{hi}. We define a rotated weight error vector as [12][21]

$$\boldsymbol{c}(n) = \boldsymbol{Q}_h^T[\boldsymbol{u}(n) - \boldsymbol{u}_0] \tag{A3}$$

where $\boldsymbol{u}_0$ is the optimum value of the output layer weight vector. Substituting (A3) into (A1), we get

$$\boldsymbol{c}(n+1) = (1+\alpha)\boldsymbol{c}(n) - \alpha\boldsymbol{c}(n-1) + 2\mu\tilde{\boldsymbol{y}}_h(n)e_0(n) - 2\mu(n)\tilde{\boldsymbol{y}}_h\tilde{\boldsymbol{y}}_h^T(n)\boldsymbol{c}(n) \tag{A4}$$

where $\tilde{\boldsymbol{y}}_h(n) = \boldsymbol{Q}_h^T\boldsymbol{y}_h(n)$ is the rotated hidden output vector and $e_0(n)$ is the error after the weight vector has converged to $\boldsymbol{u}_0$. Taking the expectation of (A4) and using the orthogonality principle $E\{\tilde{\boldsymbol{y}}_h(n)e_0(n)\} = 0$, we have

$$E\{\boldsymbol{c}(n+1)\} + (2\mu\Lambda - (\alpha+1)\boldsymbol{I})E\{\boldsymbol{c}(n)\} + \alpha E\{\boldsymbol{c}(n-1)\} = 0 \tag{A5}$$

which is the second-order difference equation of $E\{\boldsymbol{c}(n)\}$. In order to make the system stable, the roots of the following corresponding second-order equation of r_i must be within the unit circle.

$$r_i^2 + (2\mu\lambda_{hi} - \alpha - 1)r_i + \alpha = 0 \qquad i = 1, 2, \cdots, N. \tag{A6}$$

For the value to be real, the discriminate B of (A5) must be nonnegative. That is,

$$B \equiv (2\mu\lambda_{hi} - \alpha - 1)^2 - 4\alpha \ge 0. \tag{A7}$$

By solving this inequality, we get the conditions for μ that

$$\mu \le \frac{(1-\sqrt{\alpha})^2}{2\lambda_{hi}} \quad \text{and} \quad \mu \ge \frac{(1+\sqrt{\alpha})^2}{2\lambda_{hi}}. \tag{A8}$$

If $0 \le \alpha < 1$, then $0 < \mu < 2/\lambda_{hi}$, and the roots of (A5) are

$$r_i = \frac{\alpha + 1 - 2\mu\lambda_{hi} \pm \sqrt{B}}{2} \qquad i = 1, 2, \cdots, N. \tag{A9}$$

For stability, let $|r_i| < 1$ and the following four conditions result:

$$\alpha + 1 - 2\mu\lambda_{hi} + \sqrt{B} < 2 \tag{A10a}$$

$$\alpha + 1 - 2\mu\lambda_{hi} - \sqrt{B} < 2 \tag{A10b}$$

$$\alpha + 1 - 2\mu\lambda_{hi} - \sqrt{B} > -2 \tag{A10c}$$

$$\alpha + 1 - 2\mu\lambda_{hi} + \sqrt{B} > -2. \tag{A10d}$$

If $\mu > 0$, (A10a) is ensured. If $\mu > 0$ and $0 \le \alpha < 1$, (A10b) is automatically satisfied. By rearranging (A10c), we get an upper bound of μ:

$$\mu < \frac{\alpha + 1}{\lambda_{hi}}. \tag{A11}$$

Inequality in (A11) provides a relationship between μ and α. If $\alpha = 0$, we get the well known upper bound of μ for the LMS algorithm, where λ_{hi} is the maximum eigenvalue of the correlation matrix. We know that α should be less than 1, so the upper bound here will always be less than that of the conventional LMS algorithm (i.e., $\mu < 2/\lambda_{hi}$). With the condition in (A11), (A10d) is also satisfied.

If the roots of (A6) are complex, which means $B < 0$, the solution can be written as

$$r_i = \frac{\alpha + 1 - 2\mu\lambda_{hi} \pm j\sqrt{4\alpha - (\alpha + 1 - 2\mu\lambda_{hi})^2}}{2} \qquad i = 1, 2, \cdots, N. \tag{A12}$$

If $|r_i| < 1$, the solution will be stable [22]. By rearranging (A12), this condition can be satisfied by the previous assumption (i.e., $\alpha < 1$).

Convergence Characteristics of the Hidden Layer Weights

The hidden layer weight updating function is

$$\boldsymbol{w}_j(n+1) = w_j(n) + 2\mu e(n)u_j(n)y_{hj}(n) \cdot (1 - y_{hj}(n))\boldsymbol{x}(n) + \alpha\,[\boldsymbol{w}_j(n) - \boldsymbol{w}_j(n-1)]. \tag{A13}$$

In this case, it is much more difficult to pursue an analysis as above because of the nonlinear sigmoid function involved in the weight updating equation. By examining the nature of the sigmoid function, we find that we can use a group of linear functions to approximate it. Therefore, some useful linear analysis algorithms and the results we derived above can be applied to analyze the characteristics of the hidden layer weight updating equation.

Let us first examine the product term $y_{hj}(n)(1 - y_{hj}(n))$ in (A1), which we express here as $g(x) = f(x)[1 - f(x)]$ and $f(x) = 1/(1 + e^{-x})$. We notice that $g(x)$ is near 0 when $|x| > 6$ as shown in Fig. 16. In this case, (A13) can be simplified to

$$\boldsymbol{w}_j(n+1) = \boldsymbol{w}_j(n) + \alpha[\boldsymbol{w}_j(n) - \boldsymbol{w}_j(n-1)]. \tag{A14}$$

Following similar procedures to those above, we define a rotated difference term between the optimal weight vector and the actual weight vector as

$$\boldsymbol{c}(n) = \boldsymbol{Q}_x^T[\boldsymbol{w}_j(n) - \boldsymbol{w}_{j0}]. \tag{A15}$$

By substituting (A15) into (A14), we get

$$\boldsymbol{c}(n+1) - (\alpha+1)\boldsymbol{c}(n) + \alpha\boldsymbol{c}(n-1) = 0 \tag{A16}$$

which is a second-order difference equation purely controlled by α. The corresponding characteristic function of

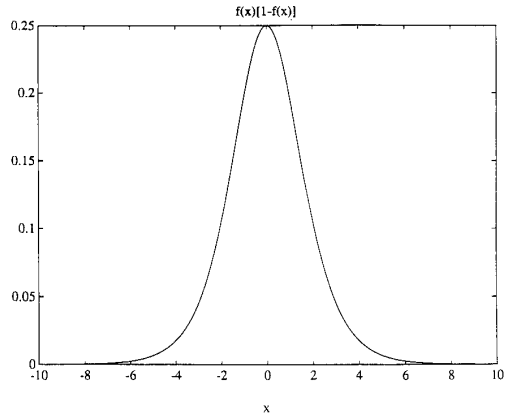

Fig. 16. A plot of function $g(x) = f(x)(1 - f(x))$ where $f(x) = 1/(1 - e^{-x})$. When $|x| > 6$, $g(x)$ is near zero.

(A16) is

$$r_i^2 - (\alpha + 1)r_i + \alpha = 0 \qquad i = 1, 2, \cdots, N. \quad \text{(A17)}$$

Its discriminate

$$B = (\alpha + 1)^2 - 4\alpha = (1 - \alpha)^2$$

is always larger than 0. Thus the function has only real roots, which are

$$r_i = \frac{\alpha + 1 \pm \sqrt{B}}{2} = \frac{\alpha + 1 \pm (1 - \alpha)}{2}$$

$$i = 1, 2, \cdots, N$$

and

$$r_{i1} = 1, \quad r_{i2} = \alpha. \quad \text{(A18)}$$

Here we use the assumption that $\alpha < 1$. We can see that this assumption is also the only condition to make $|r_i| < 1$. The solution to the difference equation is

$$c_i(n) = D_1 + D_2\alpha^n \quad \text{(A19)}$$

and

$$\lim_{n \to \infty} c_i(n) = D_1. \quad \text{(A20)}$$

This equation means that all the weights will converge to a constant value eventually instead of zero. This is equivalent to a local minima problem in which the weights do not converge to optimal values but converge to some other values. This situation can happen if the linear summation output of the hidden units keeps falling down to very nonlinear part of the sigmoid function. This partially explains why normalization of the input data is necessary.

The analysis from (A14) to (A20) reveals a very interesting difference between linear and nonlinear processes. In the linear adaptive filters, there are more straightforward constraints for convergence; for example, an inappropriate scaled input vector can cause the filter to diverge. In this nonlinear adaptive filter, however, the nonlinear function constrains the inappropriately scaled input vector. But the tradeoff is that the weights can deviate from the optimal set or can even be trapped into local minima. We can define this case as another kind of divergence for the nonlinear adaptive filter, although the values of weights and output do not become bizarre like the divergence that occurs in linear filters.

When $|x| < 6$, a segment linearization method is developed to approximate the sigmoid function by a group of linear functions. We expand the sigmoid function by the zero and first order polynomial function:

$$f(x) = f(x_{0i}) + f'(x_{0i})x = k_{0i} + k_{1i}x$$

$$i = 1, 2, \cdots, L \quad \text{(A21)}$$

where $f'(x)$ is the first derivative of $f(x)$ and L is the number of linear segments. We select $x_{0i} = 0, \pm1, \pm2, \pm3, \pm4, \pm5$, respectively.

$$\boldsymbol{k}_0 = [0.04, 0.09, 0.18, 0.33, 0.47, 0.5, 0.53, 0.67, 0.81, 0.91, 0.96]$$

and

$$\boldsymbol{k}_1 = [0.007, 0.018, 0.045, 0.11, 0.2, 0.25, 0.2, 0.11, 0.045, 0.018, 0.007]$$

Therefore the sigmoid function can be linearized as

$$f(x) = \frac{1}{1 + e^{-x}} = \begin{cases} 0 & x < -6 \\ k_{01} + k_{11}x & -6 \le x \le -4.5 \\ k_{02} + k_{12}x & -4.5 \le x < -3.5 \\ \vdots & \vdots \\ k_{0L} + k_{1L}x & 4.5 \le x < 6 \\ 1 & 6 \le x \end{cases} \quad \text{(A22)}$$

Fig. 17 provides an overlay which shows that the segment linearized function is a very good approximation of the original sigmoid function. By substituting the segment linearized function for the original sigmoid function, we rederive the back propagation learning algorithm. The output of the filter is

$$y(n) = \sum_{j=1}^{N} \hat{u}_j f(\boldsymbol{w}_j^T \cdot \boldsymbol{x}_n) + \hat{u}_0$$

$$= \sum_{j=1}^{N} \hat{u}_j (k_{0lj} + k_{1lj}\boldsymbol{w}_j^T \cdot \boldsymbol{x}_n) + \hat{u}_0$$

$$= \sum_{j=1}^{N} \hat{u}_j k_{0lj} + \sum_{j=1}^{N} \hat{u}_j k_{1lj} \boldsymbol{w}_j^T \cdot \boldsymbol{x}_n + \hat{u}_0$$

$$l = 1, 2, \cdots, L. \quad \text{(A23)}$$

The error between the desired output and the actual output is $e(n) = d(n) - y(n)$, and its square is

$$\epsilon = [d(n) - y(n)]^2. \quad \text{(A24)}$$

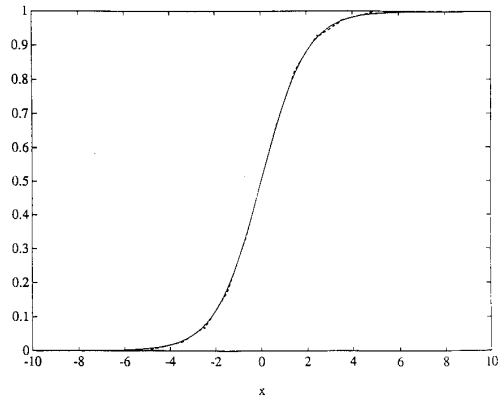

Fig. 17. A piecewise linear approximation of the sigmoid function. Solid line: sigmoid function. Dashed line: piecewise linear approximation.

The partial derivative of the error square with respect to w_j is

$$\frac{\partial \epsilon}{\partial w_j} = 2e(n)\frac{\partial y}{\partial w_j} = -2e(n)\hat{u}_j k_{lj} x_n \qquad \text{(A25)}$$

So the gradient descent algorithm for updating the weight vector w_j of the hidden layer becomes

$$w_j(n+1) = w_j(n) + 2\mu e(n)\hat{u}_j k_{1lj}\boldsymbol{x}(n) + \alpha[w_j(n) - w_j(n-1)]. \qquad \text{(A26)}$$

Comparing (A26) with the original weight updating function of (A13), the nonlinear part $y_{hj}(n)(1 - y_{hj}(n))$ is replaced by a tangent value of $f(x)$ at that segment. Comparison of (A26) with (A1), the output layer weight updating function, shows that they are very similar; both are second-order linear difference equations of weight vectors. The only difference is that in (A26), the two value $\hat{u}_j$, which is the weight connected to jth hidden unit from the upper layer and k_{1lj} are multiplied by the updating term. If we let $\hat{\mu} = \mu\hat{u}_j k_{1lj}$, (A26) becomes

$$w_j(n+1) = w_j(n) + 2\hat{\mu}e(n)\boldsymbol{x}(n) + \alpha\,[w_j(n) - w_j(n-1)] \qquad \text{(A27)}$$

which has the same form as (A1). Thus all the analyses in the last section can be applied to (A27) by replacing μ by $\hat{\mu}$. Let us study the upper bound condition of (A23) under the new step size $\hat{\mu}$. The upper bound of the hidden layer step size $\hat{\mu}$ is

$$\hat{\mu} < \frac{\alpha+1}{\lambda_{xi}} \Rightarrow \mu\hat{u}_j k_{1lj} < \frac{\alpha+1}{\lambda_{xi}} \Rightarrow \mu < \frac{\alpha+1}{\hat{u}_j k_{1lj}\lambda_{xi}}. \qquad \text{(A28)}$$

If we assume that the upper layer weight connected to ith hidden unit has little change (this is the case when the process is near convergence), the upper bound of μ is changed with the different tangent of the sigmoid function. In the nonlinear portion of the sigmoid function, the value of k_{1lj} is very small. The upper bound of μ then gets larger. While in the near linear part of the sigmoid function, the value of k_{1lj} is relatively larger, and the upper bound of μ is smaller.

We know that the local minima in the error surface are introduced by the combination of nonlinear functions, for example, the weighted summation of the sigmoid functions. Preceding analysis of the difference equation for every nonlinear portion of the sigmoid function showed that weights can be trapped to some nonoptimal values. The condition in (A28) gives us a possible way to improve this problem. We can use a larger step size for smaller tangent values, (i.e., the more nonlinear portions of the sigmoid function). This larger step size can help the process "jump" out of the possible local minima.

REFERENCES

[1] P. O. Börjesson, O. Pahlm, L. Sörnmo, and M. Nygårds, "Adaptive detection based on maximum a posteriori estimation," *IEEE Trans. Biomed. Eng.*, vol. BME-29, pp. 341-351, 1982.

[2] J. Pan and W. J. Tompkins, "A real-time QRS detection algorithm," *IEEE Trans. Biomed. Eng.*, vol. BME-32, pp. 230-236, 1985.

[3] P. S. Hamilton and W. J. Tompkins, "Quantitative investigation of QRS detection rules using the MIT/BIH arrhythmia database," *IEEE Trans. Biomed. Eng.*, vol. BME-33, pp. 1157-1165, 1986.

[4] ——, "Adaptive matched filtering for ECG detection," in *Proc. Ann. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 1988, pp. 145-146.

[5] Q. Z. Xue, Y. H. Hu, and W. J. Tompkins, "A weight pattern study of neural networks with ECG signal pattern recognition," in *Proc. Ann. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 1989, pp. 2023-2024.

[6] ——, "Training of ECG signals in neural network pattern recognition," in *Proc. Ann. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 1990, pp. 1465-1466.

[7] S. C. Lee, "Using a translation-invariant neural network to diagnose heart arrhythmia," in *Proc. Ann. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 1989, pp. 2025-2026.

[8] J. Y. Cheung and S. Hull, Jr., "Detection of abnormal electrocardiograms using a neural network approach," in *Proc. Ann. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 1989, pp. 2015-2016.

[9] K. P. Lin and W. H. Chang, "Classification of QRS pattern by an associative memory model," in *Proc. Ann. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 1989, 2017-2018.

[10] R. P. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Mag.*, pp. 4-22, Apr. 1987.

[11] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 2nd ed. New York: McGraw-Hill, 1984.

[12] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice Hall, 1985.

[13] Q. Z. Xue, Y. H. Hu, and W. J. Tompkins, "Analysis of the hidden units of back-propagation model by singular value decomposition (SVD)," in *Proc. Int. Conf. Neural Networks*, Vol. I, 1990, pp. I739-I742.

[14] MIT-BIH Database Distribution, Massachusetts Inst. Technol., 77 Massachusetts Avenue, Room 20A-113, Cambridge, MA 02139.

[15] J. Sietsma and R. Dow, "Neural net pruning—Why and how," in *Proc. IEEE Int. Conf. Neural Networks*, Vol. II, 1988, pp. 205-212.

[16] D. E. Rumelhart, "Learning and generalization," Plenary Address, *IEEE Int. Conf. Neural Networks*, 1988.

[17] A. S. Weigend, B. A. Huberman, and D. E. Rumelhart, "Predicting the future: A connectionist approach," Tech. Rep. No. PDP-90-01, Dep. Psychol., Stanford Univ., 1990; also in *Int. J. Neural Syst.*, vol. 1, no. 3, 1990.

[18] Y. Chauvin, "A back-propagation algorithm with optimal use of hidden units," in *Advances in Neural Information Processing Systems, Vol I*, D. Touretzky and M. Kaufmann, Eds. 1989, pp. 519-526.

[19] K. J. Lang and G. E. Hinton, "Dimensionality reduction and prior knowledge in E-set recognition," in *Advances in Neural Information Processing Systems, Vol. II.* D. Touretzky and M. Kaufman, Eds. 1990, pp. 179-185.

[20] S. J. Hanson and L. Y. Pratt, "Comparing biases for minimal net-

work construction with back-propagation,'' in *Advances in Neural Information Processing Systems, Vol. I*, D. Touretzky and M. Kaufmann, Eds. 1989, pp. 177–185.
[21] S. Haykin, *Adaptive Filter Theory, Information and Sciences*. Englewood Cliffs, NJ: Prentice-Hall, 1986.
[22] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975.

Qiuzhen Xue (S'87–M'91) received the M.S. degree in electrical engineering from the Shanghai University of Technology in 1982, and the M.S. and Ph.D. degrees in electrical engineering from the University of Wisconsin-Madison in 1987 and 1991, respectively.

From 1983 to 1985 he was with the Department of Electrical Engineering, Shanghai University of Technology, as an instructor and a research associate. From 1986 to 1991 he was a Research Assistant at the Waisman Center of the University of Wisconsin-Madison. Since June 1991, he has been with the diagnostics division of Miles Inc, where he is now a Senior Research Associate Scientist. His current interests include biomedical signal processing; pattern recognition; and neural networks.

Yu Hen Hu (S'79–M'80–SM'87) received the B.S.E.E. degree from National Taiwan University, Taipei, Taiwan, ROC in 1976, the MSEE and Ph.D. degrees in electrical engineering from University of Southern California, Los Angeles, CA, in 1980 and 1982, respectively.

From 1983 to 1987, he has been an assistant professor of the Department of Electrical Engineering, Southern Methodist University, Dallas, TX. He joined the Department of Electrical and Computer Engineering, University of Wisconsin, Madison, Wisconsin in 1987 as an assistant professor (1987–1989), and is currently an associate professor. His research interests include VLSI signal processing, artificial neural networks, spectrum estimation, fast algorithms and parallel computer architectures, and computer aided design tools for VLSI using artificial intelligence. He has published more than 80 journal and conference papers in these areas. He is a former associate editor (1988–1990) for the IEEE TRANSACTIONS OF ACOUSTIC, SPEECH, AND SIGNAL PROCESSING in the areas of system identification and fast algorithms.

Dr. Hu is a member of SIAM and Eta Kappa Nu.

Willis J. Tompkins (S'61–M'66–SM'77–F'92) received the B.S. and M.S. degrees in electrical engineering from the University of Maine, Orono, in 1963 and 1965, respectively, and the Ph.D. degree in biomedical electronic engineering from the University of Pennsylvania, Philadelphia, in 1973.

From 1965 to 1968, he was an electrical engineer at Sanders Associates, Inc., Nashua, NH, where he worked on research and development of data storage systems. He was employed from 1973 to 1974 at the Hospital of the University of Pennsylvania as a biomedical engineer. Since 1974, he has been on the faculty of the University of Wisconsin-Madison. Currently Professor of Electrical and Computer Engineering, he teaches undergraduate, graduate, and short courses on the topic of computers in medicine. His research interests include applications of microcomputer-based medical instrumentation and online biomedical computing. He is coeditor with J. G. Webster of the Prentice-Hall textbooks, *Design of Microcomputer-Based Medical Instrumentation* (Englewood Cliffs, NJ: Prentice-Hall, 1981) and *Interfacing Sensors to the IBM PC* (Englewood Cliffs, NJ: Prentice-Hall, 1988). He is coeditor with J. G. Webster, A. M. Cook, and G. C. Vanderheiden of the textbook, *Electronic Devices for Rehabilitation* (New York: Wiley, 1985).

Dr. Tompkins is a past President of the IEEE Engineering in Medicine and Biology Society. He is also a member the IEEE Computer Society, the Association for the Advancement of Medical Instrumentation, the Biomedical Engineering Society, and the American Society for Engineering Education. He is a Registered Professional Engineer in Wisconsin.