Chapter 6

Speech Enhancement by using Adaptive Neural Networks for Sensorineural Impairment

6.1 Introduction

Adaptive filters are excellent noise cancellers in digital signal processing technology. Adaptive filtering techniques are implemented for digital hearing aid (SNHL persons). Because, of their high improvement in SNR, low complexity and their capability to track non-stationary environment. Although the adaptive noise cancellation methods are extremely effective, they are having very high convergence. Convergence performance of the standard LMS algorithm can be improved by using frequency domain filtering. The orthogonal property of frequency domain filtering can be implemented by using different types of transforms.

The performance of TLMS depends on the orthogonal capabilities of the data independent transform used to preprocess the input. The usual transform used for TLMS is FFT. But the main disadvantages of FFT are, it is a complex transformation, less improvement in SNR and decorrelation efficiency is also less. However, there are some real transforms like DWHT and DHT, which are having good decorrelation efficiency and also having less computational complexity. But, still SNR improvement is less. Hence, real transforms are used, which are having good decorrelation efficiency and less computational complexity. Therefore, DWT-LMS and DCT-LMS are also implemented. It was found that the performance of DCT-LMS is significant compared to all the other algorithms in terms of convergence performance and DWT-LMS is excellent in terms of SNR improvement.
Speech enhancement using non-neural techniques has a long and diverse history, and continues to be an area of active research. Our goal is to enhance the speech signal by utilizing neural networks. This chapter presents speech processing in time-domain filtering, using neural networks in digital hearing aid especially for SNHL persons.

6.2 Neural Networks

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the network function is determined largely by the connections between elements. We can train a neural network to perform a particular function by adjusting the values of the connections (weights) between elements. Commonly neural networks are adjusted, or trained, so that a particular input leads to a specific target output. The network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically many such input/target pairs are used, in supervised learning, to train the network [16,112].

Batch training of a network proceeds by making weight and bias changes based on an entire set (batch) of input vectors. Incremental training changes the weights and biases of a network as needed after presentation of each individual input vector. Incremental training is sometimes referred to as "on line" or "adaptive" training. Neural networks have been trained to perform complex functions in various fields of application including pattern recognition, identification, classification, speech, vision, control systems etc [77]. The supervised training methods are commonly used, but other networks can be obtained from unsupervised training techniques or from direct design methods. In summary, there are different design and learning techniques that enrich the choices that a user can make.
6.3 Adaptive Linear Neural networks

Adaptive linear system responds to changes in its environment, as it is operating. Networks of this sort are often used for noise cancellation in signal processing technologies and also in control system applications. The adaptive linear element (ADALINE) networks use linear transfer function. The LMS learning algorithm is the most common learning method used to train ADALINE [32 & 86]. The basic building block of adaptive filter is the adaptive linear combiner as shown in Fig. 6.1. When the input

\[ X_k = [x_0, x_1, \ldots, x_n] \]

and the corresponding desired response \( d_k \) are applied to the linear combiner during training, the weights are given by

\[ W_k = [w_0, w_1, \ldots, w_n]^T. \]

Weights are adjusted to minimize mean square error. Thus, the linear combiner learns to produce an output,

\[ s_k = X_k^T W_k \]

which is the best linear least squares estimate of the desired response. These ideas are fundamental to learning in an adaptive filter and also in neural networks.

Figure 6.1 Adaptive linear neural networks with linear transfer function. (ADALINE)
The LMS algorithm can train only single layer linear networks. But, single layered network with linear transfer is not sufficient for noise reduction in speech signals [56 & 68]. Hence, in this work 3-layered network is used with non-linear tangent hyperbolic transfer function.

6.4 Noise reduction in speech signal by using neural networks

The earliest and most straightforward use of neural networks for speech enhancement is a direct nonlinear time-domain filter. This is a multi-layer network, which is used to map a windowed segment of the noisy speech to estimate the pure speech. The number of inputs depends on the sampling rate of the speech signal. The number of outputs is usually equal to the number of inputs. Standard backpropagation training methods is being employed to minimize the mean-squared error between the target and the output of the network. This method forms a direct mapping of noisy speech to pure speech.

6.4.1 Backpropagation training algorithm for NN noise canceller

The most popular and successful learning method for training the multilayer perceptrons is the Back propagation algorithm. The back propagation learning was reported by Rumelhart Hinton and Williams in 1986 [67]. The algorithm employs an iterative gradient-descent method of minimization, which minimizes the mean squared error between the desired output and network output (supervised learning). Fig. 6.2 shows a sigmoid ADALINE, which incorporates a sigmoid non-linearity. The input-output relation of the sigmoid can be denoted by

$$y_k = sgm(s_k)$$  \hspace{1cm} 6.4$$

A typical sigmoid function is the hyperbolic tangent is given by

$$y_k = \tanh(s_k) = \frac{1-e^{-2s_k}}{1+e^{-2s_k}}$$  \hspace{1cm} 6.5$$
ADALINE can be adapted by minimizing the mean square of the sigmoid error $\tilde{e}_k$ defined as

$$\tilde{e}_k = d_k - y_k = d_k - \text{sgm}(s_k)$$  \hspace{1cm} (6.6)

Now, the objective is to minimize $E[(\tilde{e}_k)^2]$ averaged over the set of training signals, by proper choice of weight vector. To accomplish this, a backpropagation algorithm can be used for the sigmoid ADALINE network [57, 66 & 97]. The instantaneous gradient estimate obtained during presentation of the $k^{th}$ input vector $X_k$ is given by

$$\tilde{V}_k = \frac{\partial (\tilde{e}_k)^2}{\partial W_k} = 2\tilde{e}_k \frac{\partial \tilde{e}_k}{\partial W_k}$$ \hspace{1cm} (6.7)

Differentiating the equation 6.6 yields

$$\frac{\partial \tilde{e}_k}{\partial W_k} = -\frac{\partial \text{sgm}(s_k)}{\partial W_k} = \text{sgm}'(s_k) \frac{\partial (s_k)}{\partial W_k}$$ \hspace{1cm} (6.8)

where $s_k$ is as expressed in equation 6.3.

Therefore,

$$\frac{\partial (s_k)}{\partial W_k} = X_k$$ \hspace{1cm} (6.9)
Substituting into equation 6.8 gives,
\[
\frac{\partial \tilde{e}_k}{\partial W_k} = -sgm'(s_k)X_k
\]  
\[\text{6.10}\]

Substituting this into equation 6.7 yields
\[
\tilde{V}_k = -2\tilde{e}_k sgm'(s_k)X_k
\]  
\[\text{6.11}\]

Then the weight iteration algorithm becomes
\[
W_{k+1} = W_k + \mu (-\tilde{V}_k)
\]  
\[\text{6.12}\]

By substituting the gradient equation 6.11 in 6.12,
\[
W_{k+1} = W_k + 2\mu \tilde{e}_k sgm'(s_k)X_k
\]  
\[\text{6.13}\]

The above algorithm is the backpropagation algorithm for the sigmoid ADALINE element [132 & 138].

Figure 6.3 Implementation of backpropagation for sigmoid adaptive linear neural network.

Implementation of the backpropagation algorithm given in equation 6.13 is as shown in Fig. 6.3. If sigmoid is the hyperbolic tangent function as expressed in equation 6.5, then the derivative \( sgm'(s_k) \) is given by
\[ sgm'(s_k) = \frac{\partial (\tanh(s_k))}{\partial(s_k)} \]

By simplifying the above equation
\[ sgm'(s_k) = 1 - (\tanh(s_k))^2 = 1 - y_k^2 \]

Where
\[ (\tanh(s_k))^2 = y_k^2 \]

Accordingly, equation 6.13 becomes
\[ W_{k+1} = W_k + 2\mu\hat{e}_k(1 - y_k^2)X_k \]

During the training period of Back Propagation NN, we start modifying the weights at the output layer, and then we proceed backwards on the hidden layers one by one until we reach the input layer.

6.5 Results and evaluation

In this chapter, direct time-domain neural network filtering approach with backpropagation training algorithm has been implemented for the enhancement of the speech signal in digital hearing aid for SNHL persons. The performance of the algorithm has been compared using output SNR, time plots and intelligibility tests.

6.5.1 Performance evaluation by using output SNR, eigenvalue ratio and time plots

The network is trained on 3 different sentences from different speakers. The network was tested using a speech signal, which not in the training set. The algorithm is evaluated for corrupted speech signals with different types of noises like cafeteria, low frequency and babble noise with different SNR. The various parameters like \( \mu \), number of layers in the network and number of neurons in each layer were changed and the performance of the algorithm was evaluated.
The input signal is a speech sentence in English and is recorded with sampling frequency 22050 Hz in different noisy conditions to evaluate the effectiveness in removing the noise from the speech signal.

The performance of the algorithm was studied, for different values of $\mu$, number of layers and numbers of neurons in each layers. From the studies we noticed that for $\mu = 0.01$, number of layers are three (one input layer, one hidden layer and one output layer) and the number of neurons are 40 in each layers (40:40:40) gives better SNR and intelligibility improvement.

For different input SNR, the output SNR and convergence ratios are calculated. Off-line implementations show that the SNR improvement in direct time-domain neural network filtering approach with backpropagation training algorithm is 11.02 dB for 0dB input SNR. The learning curve (steep curve) shows that, the network can converge to the optimal solution quickly as shown in Fig 6.5. Fig. 6.4 shows the time plots for pure signal, corrupted signal (input signal) with -5dB SNR and Fig.6.5 show the learning curve for the neural network.

<table>
<thead>
<tr>
<th>SNR of the input signal in dB</th>
<th>SNR of the output signal in dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>-5</td>
<td>9.97</td>
</tr>
<tr>
<td>0</td>
<td>11.02</td>
</tr>
<tr>
<td>+5</td>
<td>12.29</td>
</tr>
</tbody>
</table>

*Table 6.1 SNR of the output signal for different input SNR to NN filtering*
Figure 6.4 Pure signal, contaminated signal and NN filtered signal
6.5.2 Intelligibility Test

In order to measure the performance of clinical intelligibility of the algorithms listening tests were carried out. The tests were conducted as explained in the section 2.6.3. The results are displayed in Tables 6.1, 6.2 and 6.3 for $-5\text{dB}$ input SNR. The result indicates that a considerable improvement is obtained, particularly for moderate to severe SNHL subjects.

<table>
<thead>
<tr>
<th>Group 1</th>
<th>Group 2</th>
<th>Group 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>96 %</td>
<td>78 %</td>
<td>63 %</td>
</tr>
</tbody>
</table>

*Table 6.2 Average intelligibility score for the noiseless signal*

<table>
<thead>
<tr>
<th>Types of noise</th>
<th>Cocktail party noise</th>
<th>Babble noise</th>
<th>Low frequency noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group 1</td>
<td>73 %</td>
<td>78 %</td>
<td>83 %</td>
</tr>
<tr>
<td>Group 2</td>
<td>31 %</td>
<td>34 %</td>
<td>38 %</td>
</tr>
<tr>
<td>Group 3</td>
<td>15 %</td>
<td>13 %</td>
<td>16 %</td>
</tr>
</tbody>
</table>

*Table 6.3 Average intelligibility score for the signal plus noise*
<table>
<thead>
<tr>
<th>Types of noise</th>
<th>Cocktail party noise</th>
<th>Babble noise</th>
<th>Low frequency noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group 1</td>
<td>95 %</td>
<td>92 %</td>
<td>93.5 %</td>
</tr>
<tr>
<td>Group 2</td>
<td>76 %</td>
<td>74 %</td>
<td>73 %</td>
</tr>
<tr>
<td>Group 3</td>
<td>68 %</td>
<td>65 %</td>
<td>63 %</td>
</tr>
</tbody>
</table>

*Table 6.4 Intelligibility improvements by adaptive neural network noise canceller for three groups of subjects.*

The result of recognition test for filtered signals is displayed in Table 6.3. It is seen that after adaptive neural network processing the intelligibility improvement is achieved. Neural network filter showed an average intelligibility improvement of 1 % with normal subjects, 1 % with mild to moderate SNHL subjects and 5 % with moderate to severe SNHL subjects as compared to NLMS with cocktail party noise.

### 6.6 Conclusion

Off-line implementations shows that the SNR improvement in direct time-domain neural network filtering approach with backpropagation training algorithm is almost equal to non-neural methods like DWHT-LMS and DHT-LMS.

Artificial neural networks provide an analytical alternative to conventional techniques, which are often limited by strict assumptions of normality, linearity, variable independence etc. Backpropagation networks also tend to be slower to train than other types of networks and sometimes require thousands of epochs. Fortunately, there are several fast versions, which increase the speed of backpropagation algorithms. If the learning curve is too steep, it represents that the network can learn quickly and vice-versa. A number of researchers have reported superior results over linear filtering by using methods similar to the one described above [132, 66 & 68]. In general, the direct time-domain neural network
filtering approach is most applicable for reducing noises like babble noise, low frequency noise additive noise etc., but it does not suits for all types of noises.