An Efficient Recursive Total Least Squares Algorithm for Training Multilayer Feedforward Neural Networks

Nakjin Choi, *Jun-Seok Lim, Koeng-Mo Sung

School of Electrical Engineering and Computer Science, Seoul National University
*Department of Electronics Engineering, Sejong University

nakjin@acoustics.snu.ac.kr, jslim96@chollian.net, kmsung@acoustics.snu.ac.kr

Abstract

We present a recursive total least squares (RTLS) algorithm for multilayer feedforward neural networks. So far, recursive least squares (RLS) has been successfully applied to training multilayer feedforward neural networks. But, when input data contain additive noise, the results from RLS could be biased. Such biased results can be avoided by using the recursive total least squares (RTLS) algorithm. The RTLS algorithm described in this paper gives better performance than RLS algorithm over a wide range of SNRs and involves approximately the same computational complexity of $O(N^2)$.

1. Introduction

The property that is of primary significance for an artificial neural network is the ability of the network to learn from its environment, and to improve its performance through learning. The multilayer feedforward neural networks (MFNNs) have attracted a great deal of interest due to its rapid training, generality, and simplicity. In the past decade, the use of the recursive least squares (RLS) algorithm for training MFNNs has been investigated extensively [2]-[3].

In the RLS problem, the underlying assumption is that we know the input vector exactly and all the errors are confined to the observation vector. Unfortunately, this assumption is frequently not true. Because quantization errors, human errors, modeling errors, and instrument errors may preclude the possibility of knowing the input vector exactly [4]. Particularly, in the training of MFNNs, the inputs of the hidden and output neurons of the net contains an additive noise because of the use of quantization.

To overcome this problem, the total least squares (TLS) method has been devised. This method compensates for the errors of input vector and the errors of observation vector, simultaneously. Most N-dimensional TLS solutions have been obtained by computing a singular value decomposition(SVD), generally requiring $O(N^3)$ multiplications.

In this paper, we present an efficient recursive total least squares (RTLS) algorithm for training multilayer feedforward neural networks. This RTLS algorithm was first presented by Nakjin Choi et al.[5][6]. This algorithm recursively calculates and tracks the eigenvector corresponding to the minimum eigenvalue, the estimated synaptic weights, from the inverse correlation matrix of the augmented sample matrix. Then, we demonstrate that this algorithm outperforms the RLS algorithm in training MFNNs through computer simulation results. Moreover, we show that the recursive TLS algorithm involves approximately the same order of computational complexity $O(N^2)$ as RLS algorithm.

In Section II, we will explain the training of MFNNs and in Section III, we derive an efficient RTLS algorithms for MFNNs training. In Section IV, the results and analysis of our experiment are given followed by our conclusions in Section V.

2. The Training of Multilayer Feedforward Neural Networks

The MFNNs have attracted a great deal of interest due to its rapid training, generality, and simplicity. The architectural graph in Fig. 1 illustrates the layout of a multiplayer feedforward neural network for the case of a single hidden layer.

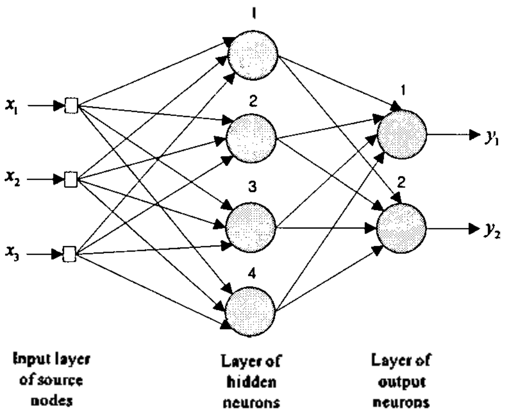
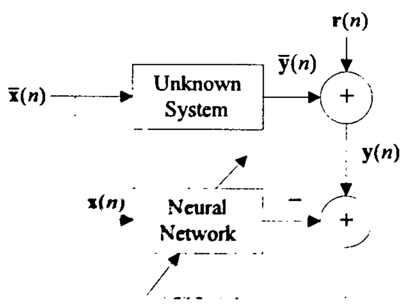
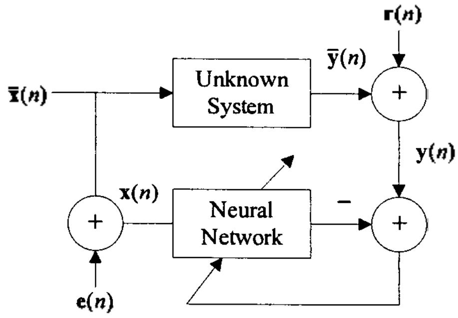


Fig. 1. MFNN with one hidden layer

An MFNN is trained by adjusting its weights according to an ongoing stream of input-output observations $\{x_k(n), y_k(n) : t=1,\cdots N;\ k=1,\cdots p\}$ where denotes the number of training samples sequences. The objective is to obtain a set of weights such that the neural networks will predict future outputs accurately. This training process is equivalent to the process of parameter estimation. Thus, training MFNN can be considered as a nonlinear identification problem where the weight values are unknown and need to be identified for the given set of input-output vectors.



(a) Output measurement noise



(b) Input and output measurement noise

Fig. 2. Training of Neural Networks

Consider the training of neural networks in Fig. 2. Here, the training problem can be posed as a parameter identification and the dynamic equation for the neural network can be cast into the following form

$$\mathbf{y}(n) = \mathbf{f}[\mathbf{w}(n), \mathbf{x}(n)] + \boldsymbol{\varepsilon}(n) \tag{1}$$

where $\mathbf{w}(n)$ is the parameter of the MFNN at time step n; is the observed output vector of the networks; $\boldsymbol{\varepsilon}(n)$ is white Gaussian noises with zero mean. The input vector $\mathbf{x}(n)$ for the training of neural network is usually taken to be clear like Fig. 2 (a). However, the fact that the unknown system input must be sampled and quantized (along with the desired signal) will result in a broad-band quantization noise contaminating the neural network input. So, Fig. 2(b) model which contains input and output measurement noise is more practical than Fig. 2(a) model which contains input measurement noise only.

The total least squares (TLS) method can compensates for the errors of input vector $\mathbf{x}(n)$ and observation vector $\mathbf{y}(n)$, simultaneously.

3. Efficient RTLS Algorithm for MFNNs

The TLS problem is a minimization problem described as follows

$$\underset{\mathbf{E},\mathbf{r}}{\text{minimize}} \|[\mathbf{E} \,|\, \mathbf{r}]\|_F \quad \text{subject to} \quad (\mathbf{A}+\mathbf{E})\mathbf{x} = \mathbf{b}+\mathbf{r} \tag{2}$$

where $\mathbf{A}$ is an M×N input matrix, $\mathbf{b}$ is an M×1 output vector. $\mathbf{E}$, $\mathbf{r}$ are the M×N input error matrix and M×1 output error vector respectively. Once a minimization $\mathbf{E}$, $\mathbf{r}$ are found, then any $\mathbf{x}$ satisfying

$$(\mathbf{A}+\mathbf{E})\mathbf{x} = \mathbf{b}+\mathbf{r} \tag{3}$$

is assumed to be able to solve the TLS problem in eq.(2).

Let's define the M×(N+1) augmented data matrix $\overline{\mathbf{A}}$ as follows

$$\overline{\mathbf{A}} = [\mathbf{A} \mid \mathbf{b}]. \tag{4}$$

Golub and Van Loan proved that the TLS solution $\mathbf{x}_{TLS}$ involves the most right singular vector of the $\overline{\mathbf{A}}$ as follows [4]

$$\mathbf{x}_{TLS} = -\frac{1}{v_{N+1,N+1}} \begin{bmatrix} v_{N+1,1} \\ \vdots \\ v_{N+1,N} \end{bmatrix} \tag{5}$$

where $$\mathbf{v}_{N+1} = \begin{bmatrix} v_{N+1,1} \\ \vdots \\ v_{N+1,N+1} \end{bmatrix}$$

Now we develop the adaptive algorithm which calculates the TLS solution recursively.

We know that the most right singular vector $\mathbf{v}_{N+1}(n)$ can be obtained from the SVD of the inverse correlation matrix $\mathbf{P}(n)$ which is defined as follows

$$\mathbf{P}(n) = \mathbf{R}^{-1}(n), \quad n=1,2,\ldots,M \tag{6}$$

where $\mathbf{R}(n) = \overline{\mathbf{A}}^H(n)\overline{\mathbf{A}}(n)$

The minimum eigenvector $\mathbf{v}_{N+1}(n-1)$ at time index $n-1$ can be represented as a linear combination of the orthogonal eigenvectors $\mathbf{v}_1(n), \cdots \mathbf{v}_{N+1}(n)$ at time index n.

$$\mathbf{v}_{N+1}(n-1) = c_1(n)\mathbf{v}_1(n) + \cdots + c_{N+1}(n)\mathbf{v}_{N+1}(n) \tag{7}$$

Because $\mathbf{P}(n)$ and $\mathbf{P}(n-1)$ are highly correlated, the coefficient $c_{N+1}(n)$ by which the minimum eigenvector $\mathbf{v}_{N+1}(n)$ is multiplied, produces a larger value than any other coefficients. That is,

$$c_{N+1}(n) \ge c_N(n) \ge \cdots \ge c_1(n) \tag{8}$$

Now, consider the new vector $\mathbf{v}(n)$ which is defined as follows

$$\mathbf{v}(n) = \mathbf{P}(n)\mathbf{v}_{N+1}(n-1) \tag{9}$$

By using the singular value decomposition (SVD) and Eq. (7), we can rewrite the vector $\mathbf{v}(n)$ as Eq. (10)

$$\begin{aligned} \mathbf{v}(n) &= \mathbf{P}(n)\mathbf{v}_{N+1}(n-1) \\ &= \left(\sum_{i=1}^{N+1} \frac{1}{\sigma_i^2(n)} \mathbf{v}_i(n)\mathbf{v}_i^H(n)\right)\left(\sum_{j=1}^{N+1} c_j(n)\mathbf{v}_j(n)\right) \\ &= \sum_{i=1}^{N+1} \frac{c_i(n)}{\sigma_i^2(n)} \mathbf{v}_i(n) \end{aligned} \tag{10}$$

From the Eq. (8) and the property of SVD, the order of the magnitudes of the coefficients $\mathbf{v}_i(n)$ becomes

$$\frac{c_{N+1}(n)}{\sigma_{N+1}^2(n)} \ge \frac{c_N(n)}{\sigma_N^2(n)} \ge \cdots \ge \frac{c_1(n)}{\sigma_1^2(n)} \tag{11}$$

So $\mathbf{v}(n)$ in Eq.(10) converges to the scaled minimum eigenvector as follows

$$\mathbf{v}(n) \approx \frac{c_{N+1}(n)}{\sigma_{N+1}^2(n)} \mathbf{v}_{N+1}(n) \tag{12}$$

Therefore, the minimum eigenvector $\mathbf{v}_{N+1}(n)$ can be approximately calculated from the $\mathbf{v}(n)$ as in Eq. (13).

$$\hat{\mathbf{v}}_{N+1}(n) = \frac{\mathbf{v}(n)}{\|\mathbf{v}(n)\|} \tag{13}$$

4. Simulation

The MFNN model to identify the unknown system is connected with one hidden layer and one output layer. And it has 1 source node, 25 hidden neurons, and 1 output neuron. The bias of hidden layer ranges from -6.0 to 6.0 which is the dynamic range of input data x. The sampling frequency is 2Hz, which satisfies the Nyquist sampling rate. Also, the synaptic weights of the first hidden layer is fixed to 1. This model can be found in [7].

The unknown system is modeled as a nonlinear system. In this simulation, we assume that the relationship between input $\bar{x}$ and $\bar{y}$ of the unknown system is

$$\bar{y} = \sin(3\bar{x}), \qquad -6 \le \bar{x} \le 6 \tag{14}$$

In this MFNN model, the output of the first hidden layer is

$$\boldsymbol{\varphi} = [\varphi_1(x-6.0) \quad \varphi_2(x-5.5) \quad \cdots \quad \varphi_1(x+6.0)]^T \tag{15}$$

and the output of total MFNN model is

$$\hat{y} = \mathbf{w}^H \varphi \text{ where } \mathbf{w} = [w_1 \quad w_2 \quad \cdots \quad w_{25}]^T \tag{16}$$

The input and output measurement noise in Fig. 2 is a zero-mean white noise process, independent of the input vector.

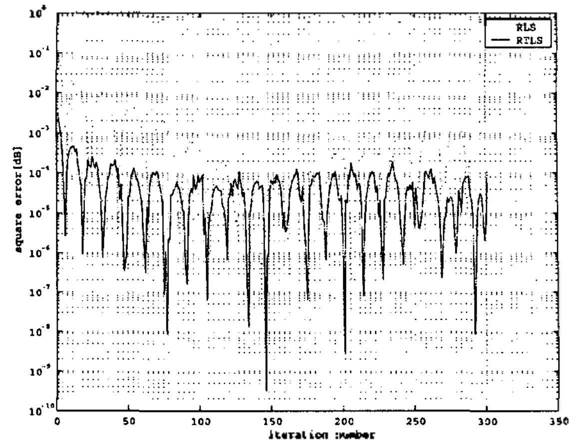
It is assumed that there is a 12, 14, and 16 bits μ-law pulse code modulation (PCM) quantizer which will produce SNRs of 73.8, 85.8, and 97.8 dB respectively. Fig. 3 shows the errors derived from RLS and RTLS with these SNRs. It is observed that the presented RTLS outperforms conventional RLS and provides better performance particularly at high SNRs.

5. Conclusions

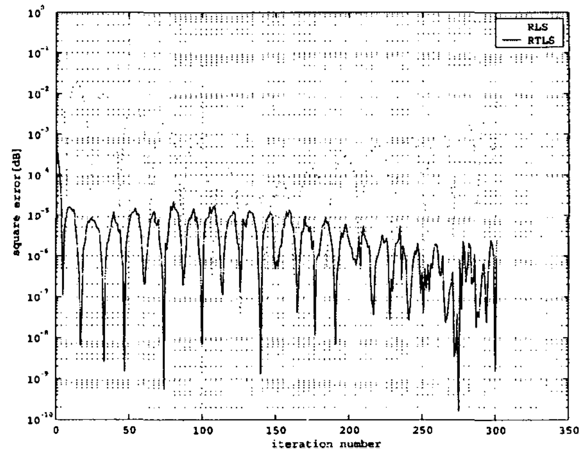
In this paper, an efficient recursive total least squares algorithm is presented. This algorithm was found to outperform the RLS algorithm in neural network training. It involves approximately the same computational complexity as RLS algorithm. In order to validate its performance, we applied this algorithm to training multilayer feedforward neural network in the various quantization noise conditions. In each case, the result showed to be better than that of the RLS.

References

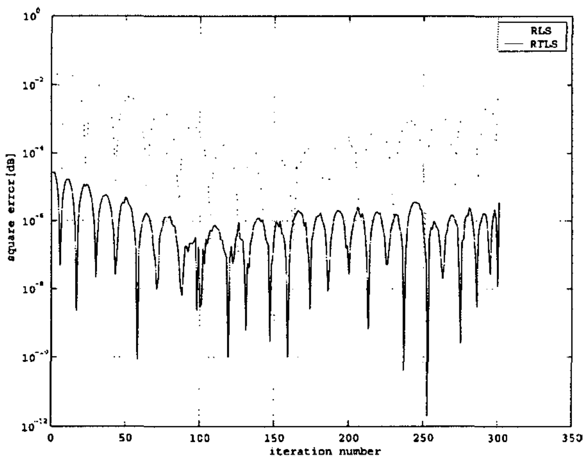
1. S. Haykin, Neural Networks : A Comprehensive Foundation, 2nd Ed., Prentice-Hall, 1999.
2. K. W. Wong and C. S. Leung, "On-line successive synthesis of wavelet networks," Neural Processing Lett., vol. 7, pp. 91-100, 1998.
3. C.S. Leung, A.C. Tsoi, "Two-regularizers for recursive least squared algorithms in feedforward multilayered neural networks," IEEE Trans. Neural Networks, vol. 12, pp. 1314-1332, 2001.
4. G. H. Golub and C. F. Van Loan, "An analysis of the total least squares problem", SIAM J. Numer. Anal., vol. 17, no. 6, pp. 883-893, Dec. 1980.
5. Nakjin Choi, J.S. Lim, J. Song, K.M. Sung, "Adaptive System Identi-fication Using an Efficient Recursive Total Least Squares Algorithm," Journal of the Acoustical Society of Korea, Vol. 22, No3E, pp.93-100, 2003.9.
6. Nakjin Choi, J.S. Lim, J.Song and K.M. Sung,"An Efficient Recursive Total Least Squares Algorithm For Fequency Estimation," ICA, vol. 25. (6A.06.02), 2001.
7. Kevin M. Passino, Biomimicry for Optimization, Control and Automation, Springer-Verlag, London, UK, 2004.



(a) 12 bit quantization noise (SNR=73.8dB)



(b) 14 bit quantization noise (SNR=85.8dB)



(c) 16 bit quantization noise (SNR=97.8dB)

Fig. 3 .Learning curves(RLS vs. RTLS)