

Pruned Neural Network for Artifact Reduction in EEG Signal

T.V. Sreenivas, Unnikrishnan, V.S. and D. Narayana Dutt
Department of Electrical Communication Engineering
Indian Institute of Science, Bangalore-560012, India

Abstract

Artificial neural networks are promising for separating signals from interference. However, their success depends on the amount of training data and network complexity which is prohibitively high. Pruned neural networks (PNN) are explored in this paper for the specific problem of separating artifacts from EEG signals. It is shown that PNN provides better performance than the fully interconnected network of similar complexity.

1 Introduction

Recovering EEG signals in the presence of embedded artifacts such as EOG, EMG, etc., is one of the major problems in biomedical signal processing. Relatively low level of the EEG signal and its spectral overlap with the artifacts cause most of the difficulties for signal enhancement [1]. With the potential of artificial neural networks for many tasks in pattern recognition [2], they have been successfully applied to the problem of noise reduction in speech processing [3]. However, these networks are large and require exponentially larger data for training. Hence, it is important to explore the architecture of the neural network such that satisfactory performance is obtained with minimal complexity. In this study, pruned neural networks (PNNs) are explored for the specific problem of artifact reduction in EEG signals.

2 EEG and Artifacts

Simultaneously recorded EOG and EEG signals sampled at 100Hz are obtained for processing. The recorded EEG, $x[n] = s[n] + y[n]$, is corrupted due to the artifact EOG, $y[n]$. To obtain an estimate of pure EEG, $\hat{s}[n]$, an estimate of $y[n]$, $\hat{y}[n]$, is obtained using the neural network and then subtracted from $x[n]$, because only pure EOG is available for training the neural network. Thus, the output of the network is $\hat{y}[n]$ and $\hat{s}[n]$ is the noise being separated. A stationary segment of 4s of the two signals is used partly for training and partly for testing the neural network. The recorded signal $SNR = \sum_n y^2[n] / \sum_n (x[n] - y[n])^2 = -0.5dB$.

3 Pruned Neural Network

A multilayer perceptron network with N-nodes at each layer and two hidden layers can be used to map arbitrary decision regions in a N-dimensional space [4]. If the signal

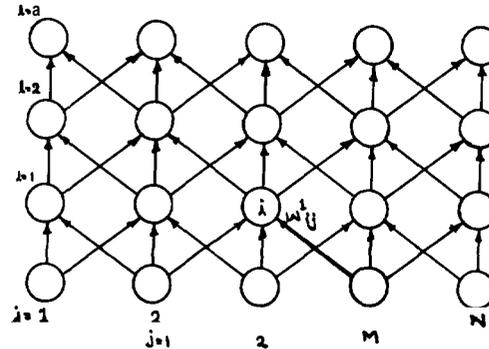


Fig.1: Pruned neural network; N=5, M=3.

and noise are separable in the N-dimensional space, it would be possible to recover the pure signal using the perceptron structure. The training of such networks is achieved using the back-propagation algorithm [5]. The most computational phase of the algorithm is the updating of the weights w_{ij}^l which is given below.

$$w_{ij}^l = w_{ij}^l + \eta x_j \frac{\delta y_i^l}{\delta x} \sum_k w_{ki}^{l+1} \frac{\delta y_k^{l+1}}{\delta x} + \mu \delta w_{ij}^l, \quad 1 \leq i, j, k, m \leq N, \quad (1)$$

where y_i^l is the output of i^{th} node at l^{th} layer and x is its input. The computational complexity of the above procedure is given by $\mathcal{O}(N^2 + N^3 + N^4)$. If the number of connections to each node from the previous layer is reduced to M , the computational complexity of updating the weights is given by $\mathcal{O}(NM + NM^2 + NM^3) = \mathcal{O}(N^2p + N^3p^2 + N^4p^3)$ where $p = M/N$ is called the pruning factor. Considering only the last term, $p = 0.5$ achieves a computational saving of 87.5%.

Considering the fact that an N-node perceptron maps the input patterns into hyper regions in the N-dimensional space, the reduced interconnections result in restricting the mapping capability to M-dimensions. However, a fully connected M-node network spans an M-dimensional space and has a computational complexity of $\mathcal{O}(M^4) = \mathcal{O}(N^4p^4)$ which is slightly lower than that of a PNN of N-nodes. The PNN with slightly more number of weights can provide more degrees of freedom for learning and also improved fault tolerance. Both these types of networks are explored for EEG signal enhancement.

The choice of M for PNN is proposed to be derived using the autocorrelation properties of the output signal. At

present, $R_{yy}[m]$ is determined and M is chosen such that $R_{yy}[m] \leq 0.1R_{yy}[0]$, $m \geq M$.

4 PNN Training

To minimize the amount of training data, the following procedure is used for selecting the input frames. The i^{th} frame F_i consists of the samples from x_i to x_{i+N-1} . Thus, $(N_{tr} - N)$ number of consecutive frames are available for training, where N_{tr} is the number of samples in the training data. The consecutive frames are sequentially presented to the network and again all frames repeated until the required convergence is achieved; this is called the sequential training method. In the repeated training method, a frame is repeatedly applied to the network a few times before taking up the next frame. This method is supposed to give better convergence [5] than the sequential method.

In the back-propagation algorithm, the coefficients η and μ have to be chosen for optimum convergence. By using an identical set of random weights for starting the network training, convergence is tested for various values of the above coefficients. It is found that $\eta = 0.3$ and $\rho = 0.01$ are optimum and these values are frozen for further iterations.

5 PNN Performance

The performance of PNN with $N = 40$, $M = 19$ and a full network with $N = 19$ is shown in the Table below for the test data case.

Iterations	1 K	3 K	6 K	12 K	15 K	20 K
SNR: PNN	3.60	4.33	4.43	4.54	4.57	4.67
SNR: Full-net	2.95	4.23	2.87	2.83	2.88	2.89

It can be seen that the PNN performs consistently better than the full-net by more than 1dB enhancement in output SNR. In terms of signal enhancement, the SNR improvement is more than 3dB and 5dB, respectively for the two networks, after about 20,000 iterations of network training. Fig.2 shows the performance of the two networks for both training and test data. As expected, the training data gives a much higher SNR ($\sim 15dB$). It is interesting that the PNN is better than the full-net for the training data also.

The effect of repeated training is illustrated in regions A and B of Fig. 2. The region below 14,000 iterations is sequential training and in region-A two repetitions are used and in region-C the training is continued with three repetitions. It can be seen that increasing the number of repetitions gives a jump in the performance of the network, but saturates soon. However, the jump in performance is not significant because the test data performance remains almost unaltered.

6 Conclusions

Pruned neural networks are promising for signal enhancement from interferences and their architecture should be tailored to the signal properties such that an optimum performance is obtained with minimal computational complexity.

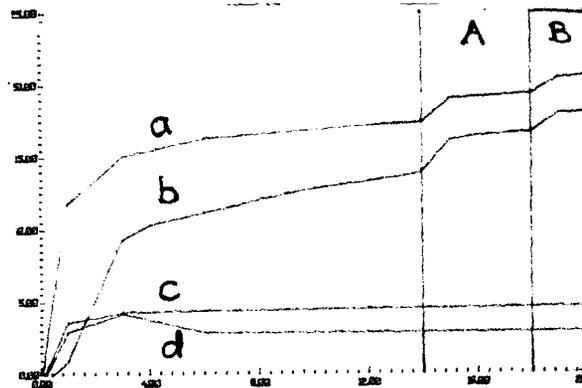


Fig.2: SNR performance of PNN (a,c) and Full-net (b,d) with iterations; (a,b) training data and (c,d) test data.

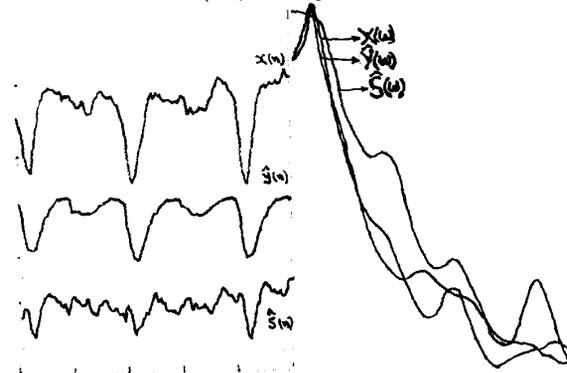


Fig.3: Signals and their respective autoregressive modelled spectra.

References

- [1] Barlow, J.S. 'Artifact processing (Rejection and minimization) in EEG data processing,' *EEG Informatics*, Revised Edition, Edited by A.Remond.
- [2] Pao, Y.H. *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley Publishing Company, Inc., 1989.
- [3] Tamura, S. and Waibel, A. 'Noise reduction using connectionist models', *Proc. IEEE Int. Conf. Acoust. Speech and Signal Processing*, New York, pp 553-556, Apr 1988.
- [4] Lippmann, R.P. 'An introduction to computing with neural nets', *IEEE ASSP Magazine*, pp 4-22, Apr 1987.
- [5] Knight, K. 'Connectionist ideas and Algorithms,' *Communications of the ACM*, Vol.33, No.11, pp 59-74, Nov 1990.