International Journal of Advanced Science and Technology
Vol. 28, March, 2011

Detection of Epilepsy Disorder Using Discrete Wavelet Transforms Using MATLABs

Laxman Tawade*, Hemant Warpe*
Vidya Pratishthan's College of Engineering, Baramati, India
tawadelaxman@rediffmail.com

Abstract

EEG (Electroencephalograph) is a technique for identifying neurological disorders. There are various neurological disorders like Epilepsy, brain cancer, etc. Epilepsy is one of the common neurological disorders. In this paper we propose a technique of detecting epilepsy disorder using discrete wavelet transform using MATLAB. The back propagation algorithm is also used in the classification network. This paper also provides a technique of detecting epilepsy disorder with great accuracy.

Keywords: *Electroencephalograph (EEG), discrete wavelet transforms (DWT)*

1. Introduction

EEG [1] is a medical imaging technique that reads scalp electrical activity generated by brain.EEG has been found a very powerful tool in neurology. The voltage range for EEG signal is 3-100 µV which is 100 times weaker than ECG signal. The frequency range of EEG signals is 5-60 Hz. Epilepsy is a common chronic neurological disorder characterized by recurrent unprovoked seizures. These seizures are transient signs and/or symptoms of abnormal, excessive or synchronous neuronal activity in the brain. Epilepsy should not be understood as a single disorder, but rather as a group of syndromes with vastly divergent symptoms but all involving episodic abnormal electrical activity in the brain. Epilepsy is one of the most common of the serious neurological disorders. Practically, EEG is used for diagnosis of Epilepsy. It takes doctors a lot of time for diagnosis. So we have developed tool for detection of Epilepsy.

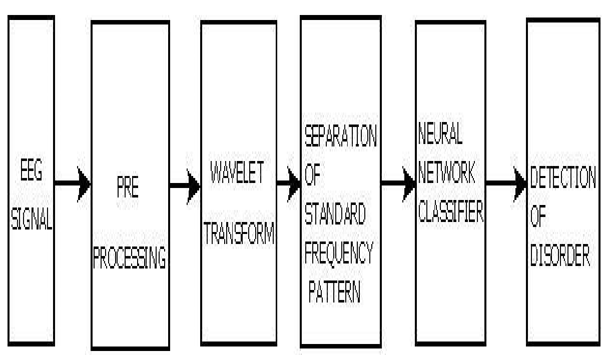


Figure 1. Epileptic Disorder Detection

The EEG signals collected from various hospitals are in .eeg format. It is first converted to excel format which is supported by MATLAB. Then it is given to a pre-processing block.

The main aim of pre-processing block is to remove various noises from EEG signal like line noise; eye blink etc. which gets added to EEG signal during EEG recording. The output of pre-processor block is then given to Discrete Wavelet Transform block which decomposes the EEG signal into different frequency bands using discrete wavelet transform. Out of four frequency bands, delta band is selected for detection of epileptic disorder. Delta wave is trained using Neural Network Classifier which uses Back Propagation Algorithm to compare the normal and epileptic delta waves. At last depending on the output of Neural Network Classifier epileptic disorder is detected. EEG signal is recorded up to 100Hz by doctors using RMS software. But the signal up to 60Hz is only useful for diagnosis of Epilepsy. So by using band pass filter we are band limiting the signal up to 60Hz. During recording EEG signal is contaminated with various artifacts [2] like 50 Hz line noise, eye blink, eye movement, muscle activity, etc. These artifacts must be removed before processing the EEG signal. 50 Hz line noise is removed by using a notch filter. Our next step is to decompose the signal into various frequency bands i.e. alpha (9-13 Hz), delta (1-3 Hz), theta (4-8 Hz), beta (14-30 Hz) using DWT. The DWT [5] analyzes the signal at different frequency bands with different resolutions by decomposing the signal into a coarse approximation and detail information. DWT employs two sets of functions, called scaling functions and wavelet functions, which are associated with low pass and high pass filters, respectively. The decomposition of the signal into different frequency bands is simply obtained by successive high pass and low pass filtering of the time domain signal. The original signal x[n] is first passed through a half band high pass filter g[n] and a low pass filter h[n]. After the filtering, half of the samples can be eliminated according to the Nyquist's /2 radians instead of □ rule, since the signal now has a highest frequency of□ . The signal can therefore be sub sampled by 2, simply by discarding every other sample. This constitutes one level of decomposition and can mathematically be expressed as follows:

$$y_{high}[k] = \sum_{n} x[n].g[2k-n]$$

$$y_{low}[k] = \sum_{n} x[n].h[2k-n]$$

Where $y_{high}[k]$ and $y_{low}[k]$ are the outputs of the high pass and low pass filters respectively, after sub sampling by 2. This decomposition halves the time resolution since only half the number of samples now characterizes the entire signal. However, this operation doubles the frequency resolution, since the frequency band of the signal now spans only half the previous frequency band, effectively reducing the uncertainty in the frequency by half. The above procedure, which is also known as the sub band coding, can be repeated for further decomposition[3,4]. At every level, the filtering and sub sampling will result in half the number of samples (and hence half the time resolution) and half the frequency band spanned (and hence doubles the frequency resolution). Figure 2 illustrates this procedure, where x[n] is the original signal to be decomposed, and h[n] and g[n] is low pass and high pass filters, respectively. The bandwidth of the signal at every level is marked on the figure as "f". We are performing 4 levels DWT of the EEG signal. The output of last DWT is a signal with frequency range 1 to 4 Hz. To separate delta wave we are passing the signal through a band pass filter. The output of band pass filter is a delta wave. Delta wave is given as a input to a Neural Network Classifier. Neural network is trained using back propagation algorithm. Back propagation has been used for several kinds of applications including classification, function approximation, and forecasting.

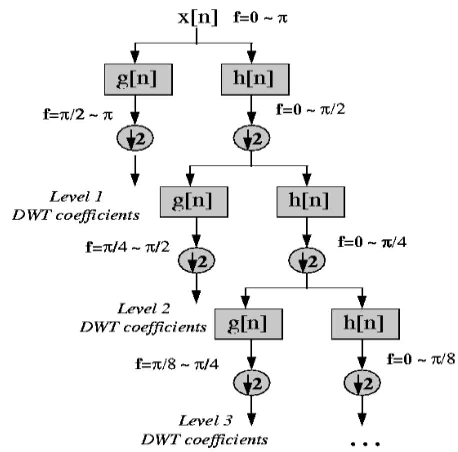


Figure 2. Sub Band Coding for Decomposition

The back propagation algorithm is a generalization of the least mean squared algorithm that modifies network weights to minimize the mean squared error between the desired and actual outputs of the network. Back propagation uses supervised learning in which the network is trained using data for which inputs as well as desired outputs are known. Once trained, the network weights are frozen and can be used to compute output values for new input samples. The feed forward process involves presenting an input pattern to input layer neurons that pass the input values onto the first hidden layer. Each of the hidden layer nodes computes a weighted sum of its inputs passes the sum through its activation function and presents the result to the output layer.

2. Architecture of Back Propagation Network

The back propagation algorithm assumes feed forward neural network architecture. In this architecture, nodes are partitioned into layers numbered 0 to L, where the layer number indicates the distance of a node from the input nodes. The lowermost layer is the input layer numbered as layer 0, and the topmost layer is the output layer numbered as layer L. Back propagation addresses networks for which $L > 2$, containing "Hidden layers" numbered 1 to L- 1. Hidden nodes do not directly receive inputs from nor send outputs to the external environment. For convenience of presentation, we will assume that $L = 2$ in describing the back propagation algorithm, implying that there is only one hidden layer, as shown in figure. The algorithm can be extended easily to cases when L >2. The presentation of the algorithm also assumes that the network is strictly feed forward, i.e., only nodes in adjacent layers are directly connected; this assumption can also be done away with. Input layer nodes merely transmit input values to the hidden layer nodes, and do not perform any computation. The number of input nodes equals the dimensionality of input patterns, and the number of nodes in the output layer is dictated by the problem under consideration. For instance, if the task is to approximate a function mapping n-dimensional input vectors to m-dimensional output vectors, the network contains n input nodes and m output nodes. An additional "dummy" input node with constant input (= 1) is also often used so that the bias or threshold term can be treated just like other weights in the network. The number of nodes in the hidden layer is up to the discretion of the network designer and generally depends on problem complexity. Each hidden node and output node applies a sigmoid function to its net input [7].

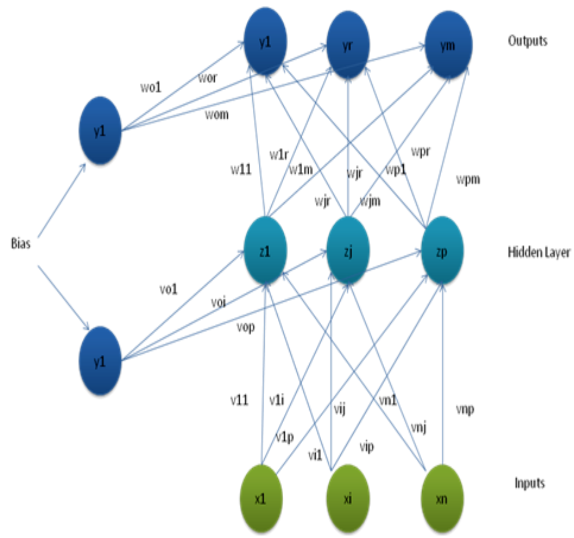


Figure 3. Architecture of Backpropogation

3. Training Algorithm

The training algorithm for the above network is given as,[6]

Let X be the input training vector X=(x1, x2,…..,xn), t be the output target vector T= (t1, t2… tm).

δk = error at output unit Yk

δj = error at hidden unit Zj

Let α be the learning rate, let Voj be bias on hidden unit j and Zj is output on hidden unit j. Wok be bias on output unit k, Yk is the output unit k.

Step 1: Initialize the weights to small random values.

Step 2: While the stopping condition is false do steps 3 to 10.

Step 3: For each of the training input pairs do steps 4 to 9.

Step 4: Each input unit receives the input signal xi and transmits this signal to all the units in the layer above (hidden layer).

Step 5: Each hidden unit Zj, j varying from 1 to p sums its weighted input signals.

$$Zinj = Voj + \sum_{j=1}^{n} Xi\ Vij$$

Step 6: Applying the activation function, Zj=f(Zinj) and send this to all units in layer above.

Step 7: Each output unit Yk,k=1 to m sums its weighted input signals

$$Yink = Wok + \sum_{j=1}^{p} Zj\ Wjk$$

Apply its activation function to calculate the output signal Yk=f(Yink).

Step 8: Each output unit i.e. Yk, k=1 to m, receives a target pattern corresponding to an input pattern. Error information term is calculated as δk=(tk-Yk)f `(Yink).

Step 9: Each hidden unit Zj,j=1 to m sums its δ inputs from the units in the layer above i.e.

$$\delta inj=\sum_{k=1}^{m} \delta j\ Wjk$$

The error information term is calculated as $\Delta j = \delta inj\ f`(Zinj)$.

Step 10: Updation of weights and biases. Each output unit Yk,k=1 to m, update the bias and weights .The weight correction term is given by

$$\Delta Wjk = \alpha\ \delta k\ Zj$$

$$\Delta Wok = \alpha\ \delta k$$

Thus,

$$Wjk(new)=Wjk(old) + \Delta Wjk$$
$$Wok(new)=Wok(old) + \Delta Wok$$

Each hidden unit Zj,j=1 to p, update its weight and bias. The weight correction term is given by,

$$\Delta Vij = \alpha\ \Delta j\ Xj$$
$$\Delta Voj = \alpha\ \Delta j$$

Thus,

$$Vij(new) = Vij(old) + \Delta Vij$$
$$Voj(new) =Voj(old) + \Delta Voj$$

Step 11: Check for stopping condition.

Selection of parameters in Back Propagation Algorithm:

To choose initial weight:-

To get best results initial weights are set to random values between -0.5 to 0.5 or -1 to 1. Selection of learning rate α: A high learning rate leads to rapid learning but the weights may oscillate, while a lower learning rate leads to slower learning.

Methods suggested for choosing the learning rate is as follows:-

1.) Start with high learning rate and steadily decrease it. Change in the weight vector must be small, in order to reduce oscillations or any divergence.
2.) Increase the learning rate to worsen performance.
3.) Learning in back propagation:
    a. Sequential learning or pre-pattern method.
    b. Batch learning or pre-epoch method.

Sequential learning: The given input pattern is propagated forward and the error is determined and back propagated and the weights are updated.

In batch learning: The weights are updated only after the entire set of training network has been presented to the network.

The weight update is performed after every epoch. In BPN the weight change is in a direction i.e. a combination of current gradient and the previous gradient. If the momentum is added to weight update formula, the convergence is faster. The weights from one or more previous training patterns must be saved in order to save the momentum. For BPN with momentum , the new weights for training step(t+2) is based on (t) and (t+1).It is found that momentum allows the net to perform large weight adjustments as long as the correction proceeds in the same general direction for several patterns. The weight updating formula for BPN with momentum is given by

$$Wjk(t+1) = Wjk(t) + \alpha\, \delta k\, Zj + M[Wjk(t) - Wjk(t-1)]$$
$$Vjk(t+1) = Vjk(t) + \alpha\, \delta j\, xj + M[Vjk(t) - Vjk(t-1)]$$

By using this algorithm, if after a certain number of iterations the output does not match with input by 70% then the patient is diagnosed as epileptic.

4. Result

EEG Signals are obtained from the various hospitals such as Rubi Hall Pune, India as shown below. These signals are in .eeg format which are not supported by the MATLAB software. The original EEG signal is shown below. We have used EEG recording software provided by the doctors and widely used all over the India to convert EEG signal in .eeg format to .xls format supported by MATLAB.

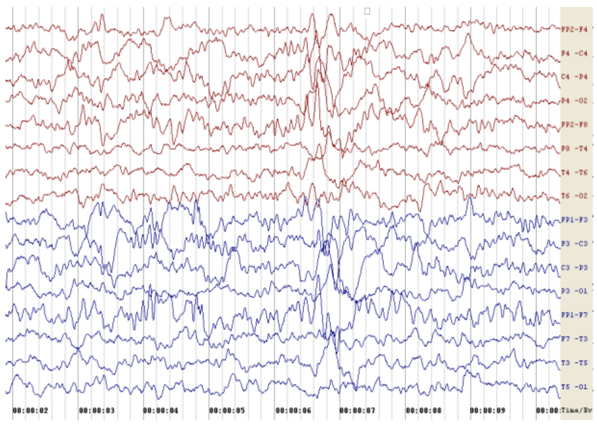


Figure 4.1 EEG

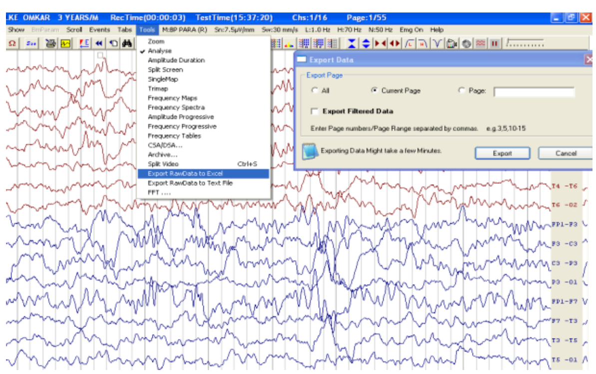


Figure 4.2 Conversion to Excel Format Preprocessing:

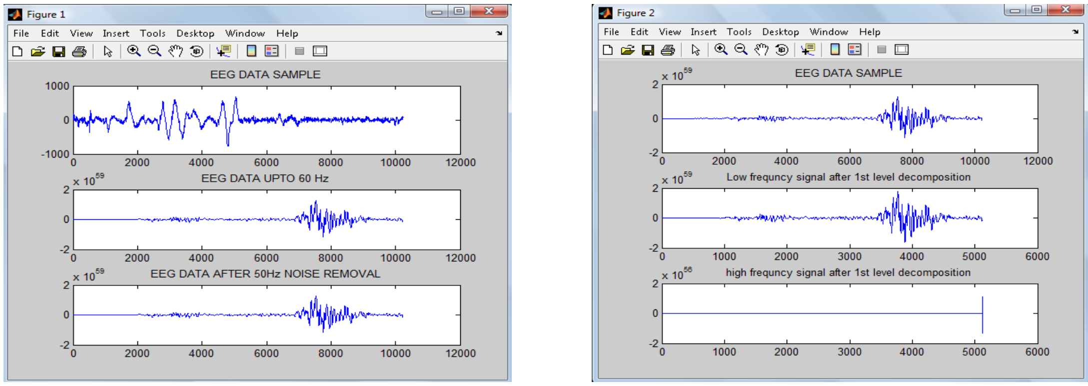
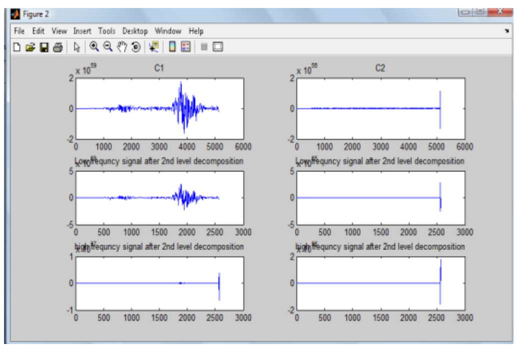


Figure 5.1 First Level Decomposition



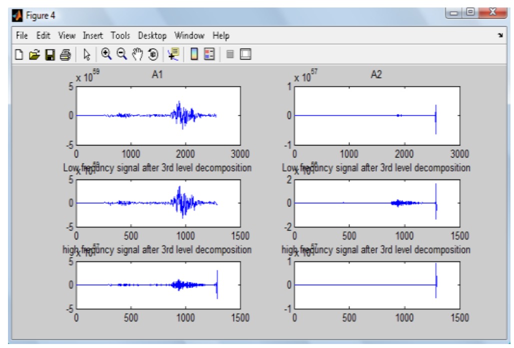
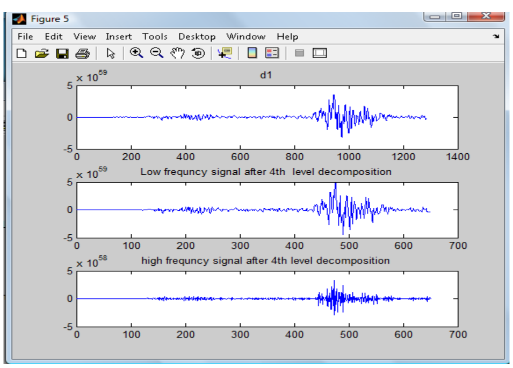


Figure 5.2 Second Level Decomposition **Figure 5.3 Third Level Decomposition**



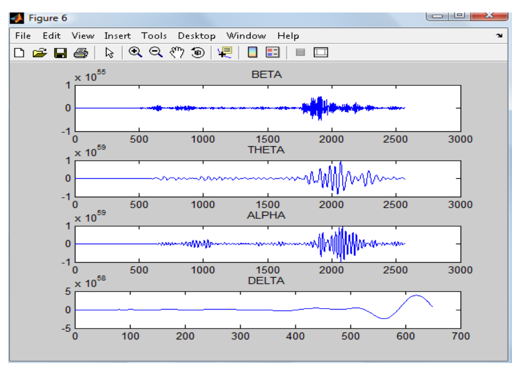


Figure 5.4 Fourth Level Decomposition **Figure 5.5 Separated Wave**

5. Conclusion

This paper proposes a technique of detecting epilepsy disorder using discrete wavelet transform using MATLAB. The back propagation algorithm is also used in the classification network. These experiments provide experimental verification that the use of this tool can be used for detection of epilepsy within few seconds. We are also able to detect the epileptic disorder with accuracy up to 96%.

References

[1] M. Teplan "Fundamentals of EEG Measurement" volume 2, section 2, 2002

[2] James N. Knight (2003) Signal fraction analysis and artifact removal in EEG. Master's Thesis, Department of Computer Science, Colorado State University, Fort Collins, CO.

[3] N.Hazarika, Classification of EEG signals using the Wavelet transform, Signal Processing [H.W. Wilson - AST]; May 1997; Vol. 59, ISS: 1; pg. 61

[4] Lisha Sun Guoliang Chang Hongrong Tang " Wavelet Packet Entropy in the Analysis of EEG Signals "Signal Processing, 2006 8th International Conference

[5] M.Akin, M.A.Arserim, M.K.Kiymik, I.Turkoglu " A NEW APPROACH FOR DIAGNOSING EPILEPSY BY USING WAVELET TRANSFORM AND NEURAL NETWORKS" Proceedings – 23rd Annual Conference – IEEE/EMBS Oct.25-28, 2001, Istanbul, TURKEY

[6] "Introduction to neural networks using MATLAB 6.0" By S. N. Sivanandam, Sumathi & Deepa.

[7] "Elements of artificial neural networks" a book by Kishan Mehrotra, Chilukuri K. Mohan, Sanjay Ranka

Author

Laxman Tawade is pursuing BE degree in Electronic & Telecommunication from institute Vidya Pratishthan's College of Engineering, Baramati, India. Currently he is member of International Association of Computer Science and Information Technology (IACSIT). He has published two papers in international conference and 1 paper in International Journal. His research interest includes biomedical signal processing, optical fiber communication and optical access networks based on WDM-PON.