Original research article

Source Position from EEG Signal with Artificial Neural Network

Tanaporn Payommai*
Department of electronics communication and Computer, Faculty of Industrial Technology, Valaya Alongkorn Rajabhat University under the Royal Patronage, Phaholyothin Road, Khlong Nuang, Klong Luang, Pathum Thani 13180, Thailand

Received 18 September 2016; Received in revised form 6 December 2016
Accepted 12 January 2017; Available online 24 March 2017

ABSTRACT

Electroencephalography (EEG) is recording of the electrical signals on the scalp. These signals come from sources of activity within the brain; however it can be difficult to determine where the sources originate from just by looking at the signals. Through signal processing, these EEG signals can be analyzed and displayed as more useful information. This research explored the evolution of EEG (Brain-waves) topography. The aim of this research was to extract the origins of brain-waves within the brain from EEG data and develop an algorithm to analyze and display this information. This was done in the MATLAB environment by creating: a working software to display and pre-process multichannel EEG data; software/algorithms that could localize sources of EEG within the brain; and a clinician-friendly GUI block. Neural networks are a supervised machine learning technique that can be used to train a system based on previously seen data. Using this approach, it is possible to accurately extract signal positions within the brain.

Keywords: Electroencephalography (EEG); Neural networks

Introduction

Brain-wave analysis is the process of studying and analyzing the electrical activity given off by the brain. It is an ongoing study with new advances every few years. Currently, there are many techniques that can be used to analyze the activity of the brain [1]. EEG is a method for measuring electrical impulses given off by the brain. The EEG signals are measured by placing a series of sensors at set positions on the scalp. This is a non-invasive and relatively cheap technique to perform, and as such, will be the technique used for the analysis.

Source localization techniques are employed to extract the source locations from a set of measuring devices. There are many techniques which use either Magnetic Resonance Imaging (MRI) data or EEG data to process and locate source origins [1]. However, most of these techniques use an iterative method to locate the source origins. Although the source locations are considered quite accurate, the time taken to produce these results is not desired when looking at a very large set of data. As such, a neural network was used to dramatically reduce the time, as the iterative process is done before hand in the training step.

Neural Networks are a supervised training method in which the input and output data is known, and a network is

***Corresponding author:** tanaporn@vru.ac.th

trained to look at that data and learn how to produce the output based on the input. For a new set of data with known inputs, but unknown outputs, the neural network will guess the output based on what it has previously seen. The initial learning stage requires a lot of memory and time to process if there is a large amount of training data. However, once trained, the network will be able to reproduce the output given an input within a very short time. This is ideal as once it is trained it will not need to be trained again unless the number of signals being input is changed.

There are already quite a few programs which can locate source origins, such as EEGLab [2] and ICALab [3-7]. However, these programs require a higher level of understanding to use effectively. The main objective of this research is to identify sources of activity within the brain using EEG data, and to display the position of brain activity and observe how those sources move over a period of time. This is to all be done in a simple and easy to use Graphical User Interface (GUI). In the research [8] the possibilities that lie within the domain of Brain-Computer Interfaces were investigated and explored, using friendly equipment that has recently become available. The Brain-Computer Interfaces (BCI) is a driving force for utilizing EEG that is the process of recording brain activity from the scalp using electrodes. The artificial neural networks (ANNs) proposed brain signal processing which is analyzed to classify EEG and MEG for brain images [9]. EEG data was divided into frequency bands and indicated that the low initial power increase mainly improved the frequency [10]. The performance of EEG analysis software used in clinical and research settings has been examined by using BCI but the forecast has some errors [11].

The overall goal at this stage of the research is to implement an algorithm to locate the origins of brain activity, and display the data as it moves over time. This research will only look at simulated data.

The organization of the rest of the research is as follows. Section 2 details the methods employed in this research, viz., head modeling, electrode positioning, neural network training, and GUI development. Results are presented in section 3. Discussion and future work are presented in section 4. Finally, Section 5 contains a conclusion.

Methods

This research will look at the localization of sources from EEG signals. This will be done by first simulating the potential voltages on the scalp of a source within the brain using a head model. Then by using the simulated potentials we pass that data to a learning algorithm to train a network.

Head model

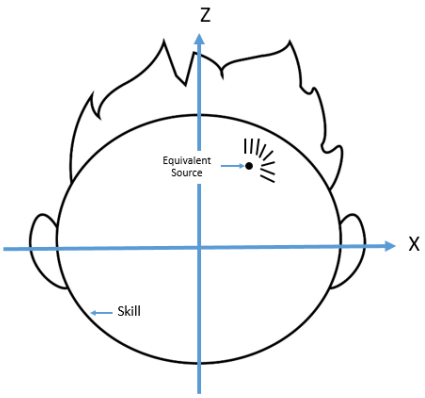

Fig. 1. Three concentric shell head model.

Fig. 1 shows the head model that will be used for this research. It is a three concentric shell model, in which the shells are the brain, the skull, and the scalp.
The voltage on the scalp is calculated as

$$V(\theta,\emptyset) = \frac{1}{4\pi\sigma}\sum_{n=1}^{\infty}\frac{2n+1}{n}bn-1\frac{(2n+1)^2\epsilon}{d^n(n+1)}$$
$$(m_r np_n^0(\cos\theta) + m_t P_n^1(\cos\theta)\cos\emptyset)) \qquad (1)$$

Where d_n is given as

$$d_n = (n + ne + e)\left(\frac{ne}{n+1} + 1 + 1(1 - e)(F_1 - F_2)\right) - n(1-e)^2\,\frac{F1}{F2} \quad (2)$$

And F_1 and F_2 are calculated as

$$F_1 = \left(\frac{r1}{R}\right)^{2n+1} \text{ and } F_2 = \left(\frac{r2}{R}\right)^{2n+1} \quad (3)$$

In equations (1) – (3), b is the eccentricity of dipole location, m_r is the radial component of the dipole moment, m_t is the tangential component of the dipole moment, r_1 is the radius of the sphere representing the brain, r_2 is the outside radius of the shell representing the skull, R is the outside radius of the shell representing the scalp, e is the brain/skull conductivity ratio (=80), σ is the conductivity of the brain, and P_n^i denotes the legendry polynomial. Equations (1) - (3) calculate the voltage at point P (Fig. 2.), given a dipole source position in the z-axis. As sources are said to be independent of each other, multiple dipoles can be represented by first calculating the potential at certain points for each source, then simply adding them together.

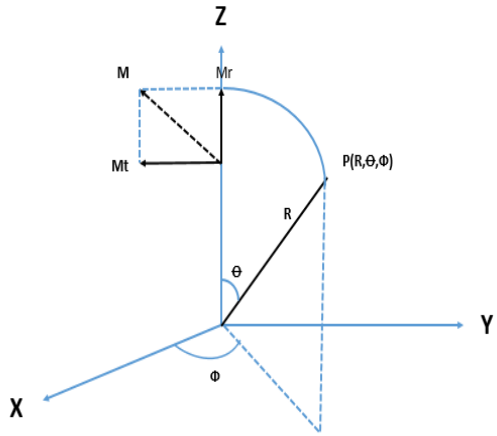

Fig. 2. Diploe M is used to calculate the voltage at scalp position P.

2.2 Electrode positioning

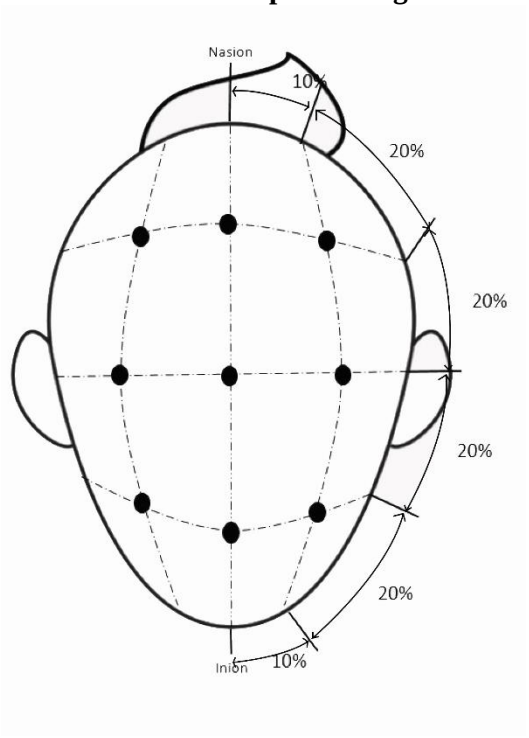

Fig. 3. 10-20 system of electrode placement.

The point P is the position of the electrodes on the scalp. These points are predetermined positions set by an international standard. The electrode placement system used for this report is the 10-20 system of electrode placement. In this system, electrodes are placed at 10% and 20% intervals as shown in Fig. 3. There are other types of electrode placement systems available which increase the number of electrodes used, such as the 10-10 system where electrodes are placed at 10% intervals of each other. This increases the number of measurements resulting in more accurate source positions. However, the 10-20 system was chosen due to computational complexities.

Neural networks

Fig. 4 shows the process flow diagram of the neural network. The neural network was trained using source parameters generated by the head model. A series of fixed dipole positions representing the ocular

dipoles, as well as dipole moment parameters that were randomly generated, were used to calculate the scalp potentials. Using the calculated voltages as inputs, they were fed into a neural network with their respective original source parameters used as target values. The network was trained using the Neural Network Toolbox in MATLAB until it was sufficiently trained to be able to accurately guess the source location, given a set of scalp voltages. The neural network training involved training a network for a set of randomized data, as well as testing on another generated test data set in order to the test the generalization of the network.

A large number of training points are required to train the network for as many possibilities as possible. An increase in training points resulted in a better trained network. However, increasing the training points too much would lead to a longer time spent training as well as using up more memory.

As neural networks themselves utilize various algorithms, various parameters and training algorithms had to be decided upon. Fig. 5 shows a block diagram of a neural network. Here there are three layers: the input layer, the hidden layer, and the output layer. During training the input and output layers are known, and the hidden layers are unknown. The hidden layer contains a set of weights (neurons) that is updated for each iteration of the input and output data. As these weights are updated, a more accurate solution is achieved. Initially, the choice of the number of neurons within each hidden layer, as well as the number of layers, had to be decided upon. As the parameters of neural networks vary from application to application, using existing literature as a starting point and performing trial-and-error tests was the most efficient way of choosing these parameters. As such, two hidden layers with 30 nodes in each layer was deemed efficient.

Various tests were undertaken in order to analyze the effectiveness of changing the number of layers and neurons. By increasing the number of hidden layers, the computational power of the network increases resulting in a more accurate solution at the cost of computational time and memory requirement. In this research, a two hidden layer network was deemed to be sufficient with little error. Increasing the number of layers did not produce a network that was more generalized for test data, hence it was deemed unnecessary to create a more complex network that would require more computational time. A similar result was found with the number of neurons. Too few neurons would not produce a network that would accurately calculate the source positions, whereas increasing the number of neurons above 30 did not produce a network that performed significantly better.

The Neural Network Toolbox offers a range of training algorithms, including the traditional gradient descent method. Four training algorithms were investigated: the LM algorithm, gradient descent, Bayesian regularization and one step secant backpropagation. The LM algorithm and gradient descent were accurate and fast at converging for smaller sets of data, but failed to generalize for larger sets of data that were used in the training. Bayesian regularization was the strongest algorithm that provided the most generalized solution, but took the longest to converge. One step secant backpropagation provided the fastest

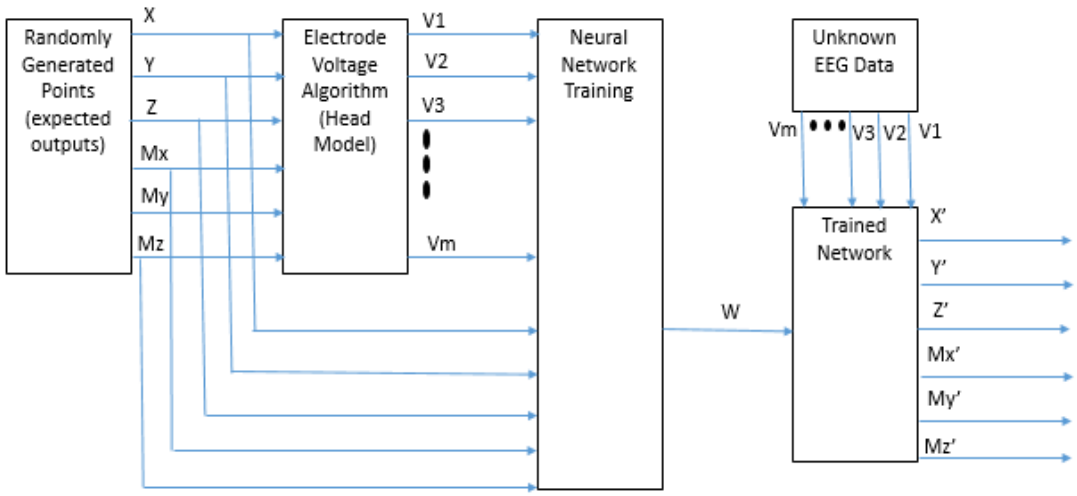

Fig. 4. Source location process flow diagram.

converging and decent generalization with large sets of data and was deemed to be the most effective in this research, as it provided similar generalization to Bayesian regularization for the same training data.

GUI development

The GUI was developed using MATLAB's graphical development package called GUIDE. The development environment has very basic functions which can be expanded with the use of the JAVA script language. However, due to having no knowledge of the JAVA language, the entire GUI was developed using GUIDE.

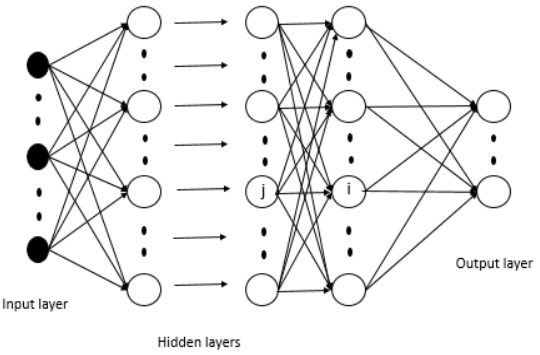

Fig. 5. Neural network architecture.

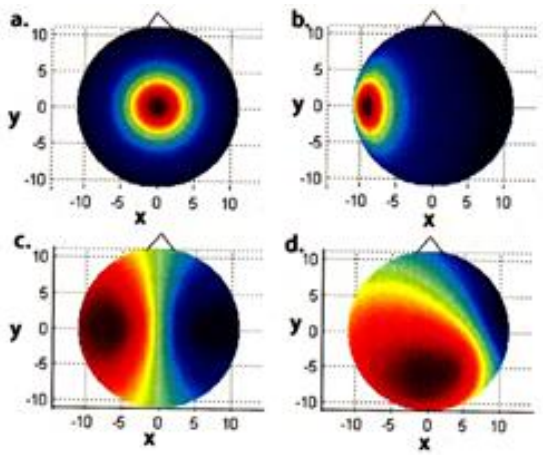

Fig. 6. Contour map of voltages calculated on the scalp given a source position. a) Dipole located in the center of the head with only a radial component. b) The rotation of (a) to a random point within the brain. c) Dipole located in center of the head with only a tangential component. d) the rotation of (c) to a random point within the brain.

Results

Head model implementation

Various tests were undertaken to check the accuracy of the head model. The first step was to generate scalp voltage at any point in the scalp of the head. This was done by implementing the formula into a MATLAB file that calculated the voltage at a point given the azimuth and latitude angles. To check the linear property of the voltage, a simple test was undertaken by doubling the magnitude of the dipole moment. The result was a voltage that was double the original result, which proved that the scalp voltage implementation was correct.

The equation for the head model requires the dipole to be situated on the z-axis as shown in Fig. 2. This means that we must rotate a source position from any point with the brain to the z-axis in order to calculate the potential given off by the source at the scalp. To do this the rotation matrices were used.

$\boldsymbol{R_z}$ rotates the source to the $\boldsymbol{z-x}$ plane.

$\boldsymbol{R_y}$ rotates the source to the z-axis.

$\boldsymbol{R_{zx}}$ rotates the sources orientation to the $\boldsymbol{zx}$ plane.

Fig. 6 a and b shows a dipole placed in the center of the brain with only a radial component, and the same dipole rotated to a different position in the brain, respectively. As the two source dipoles are directed perpendicular to the scalp, the contour map of the calculated voltage was identical as expected. The distorted image was due to the mapping of a 3-dimensional sphere on a 2-dimensional plane.

Another test was to place a tangential dipole that was also centered. As shown in Fig. 6c, the expected positive voltages on one side of the head are mirrored by the negative voltages on the other side of the head, which symbolizes the negative voltages below the dipole.

Neural network

The Neural Network Toolbox provided by MATLAB was used to train a network to locate source positions given a set of potentials. The training was done on an Intel i7 2.8 GHz processor with 8GBs of memory. The algorithm used to train the network was the one-step secant, as the performance and error was comparable to that of Bayesian regularization for generalization, and was within acceptable limits.

In this research, 30,000 random points within the brain were generated. These points were used as the target data for the neural network. The data was also fed through the head model algorithm to create 30,000 sets of scalp potentials. Each set of scalp potentials contained 19 potentials situated at the electrode positions shown in Fig. 4. The set of potential data was then used as the input data of the neural network.

An early stopping method was applied to the training phase in order to stop the localization from getting worse. An extra 200 source locations and scalp potentials were also generated to be used as test data. This data was fed through the network at each

iteration and checked to see if the accuracy got better over time.

Localization accuracy

The accuracy of the created network was done by using a bipolarity test. The test was done by taking the measured potential at one point and comparing it with the calculated potential at the same point. The residual variable (RV) between the measured and calculated points is determined by

$$RV = \frac{\sum_{i=1}^{N}(v_{m,i}-v_{c,i})^2}{\sum_{i=1}^{N}(v_{m,i})^2} \quad (4)$$

where $V_{m,i}$ is the measured potential at scalp electrode 'i', $V_{c,i}$ is the calculated potential at scalp electrode 'i', and N is the number of electrodes available. The optimal value for the residual variable would be 0, indicating that the original signal was able to be recreated with 100 percent accuracy. However, this is not possible in real world situations. The dipolarity is calculated from RV as

$$\text{Dipolarity (D)} = \sqrt{1 - RV} \quad (5)$$

Table 1. The location accuracy of a set of sources.

No.	Dipolarity
1	98.44%
2	97.08%
3	92.29%
4	79.06%
5	98.10%
6	99.70%
7	99.36%
8	95.32%
9	91.25%
Average	94.51%

Table 1 shows the location accuracy of a set of sources found within simulated EEG data, after training a network using 30,000 training points within the entire brain using an intel i7 2.8GHz with 8GB of memory. It was shown that an average accuracy of 94.51% is achieved.

Movement of dipole over time using GUI

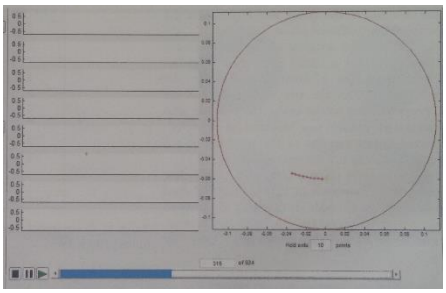

Fig. 7. Movement of diode over time.

Fig. 7 shows the GUI created to show how a source moves within the brain over time. Here, 10 dipole locations were extracted from a set of simulated EEG data and displayed within the graph. The playback feature was implemented to allow the user to view the movement of the dipole at any given time.

Discussion and Future Work

As shown in the results, the head model was successfully implemented and evaluated. Voltages at electrodes based on the international 10-20 system could be calculated for any arbitrarily positioned and orientated dipole.

Following the generation of scalp potentials, a neural network was successfully trained and tested to calculate source positions on a previously unseen set of potential data. The accuracy of the network was acceptable; however, with more training data, a more accurate solution could be created.

As of now, this research only deals with simulated data, as EEG data is currently not available. Upon receiving real EEG data, it will then be possible to continue this research to process the EEG data and show, using images, how the eyes move over time,

as well as how the sources within the brain move over time. Graphical representation is crucial as it allows people to see the information without having to look at large volumes of EEG data. Another objective is to compare the localization accuracy with existing techniques.

Conclusion

It was shown that we were able to create a network which was able to accurately guess the position of sources from simulated EEG data. We found that using 30,000 sets of training data to look for 1 source within the brain resulted in 95% accuracy of the source, which can be further increased with more training data.

Upon further research, being able to implement a more realistic head model which describes the relationship between the source and electrode sensors is recommended.

References

[1] Yao J, Dewald JP. Evaluation of different cortical source localization methods using simulated and experimental EEG data 2005;25(2): 369-82.

[2] Delorme A, Makeig S. EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics, J Neurosc Methods 2004;134(1): 9-21.

[3] Cichocki A, Amari S. Adaptive blind signal and image processing: learning algorithms and applications. Wiley; 2003.

[4] Brunner C, Delorme A, Makeig S. EEGLAB - An open source MATLAB toolbox for electrophysiological research Vol. 25 No. 2; 2013. p. 369-82.

[5] Abeyratne U, Kinouchi Y, et al. Artificial neural networks for source localisation in the human brain. 1991;4(1):3-21.

[6] Hoey GV, Clercq JD, et al. EEG dipole source localization using artificial neural networks 2000;4(45): 997-1011.

[7] Urszula S, Markowska-kaczmar U, Kozik A. Blinking artefact recognition in EEG signal using artificial neural network, 1999.

[8] Erik Andreas L. Classification of EEG signals in a brain computer interface system [M. Sc. thesis]. Trondheim: Norwegian University of Science and Technology; 2011.

[9] Jeng-Ren D, Tzyy-Ping J, Scott M. Brain signal analysis, 2009.

[10] Bert K. Application of neural network for EEG analysis No. 29; 1994. p. 39-46.

[11] Forney L. Electroencephalography classification by forecasting with recurrent neural network [M. thesis]. Fort Collins: Colorado state university; 2011.

[12] Abdulhamit S, Kiymika MK, Ahmet A, Etem K. Neural network classification of EEG signals by using AR with MLE Preprocessing for Epileptic Seizure Detection. Math Comput Appl 2005;10(1): 57-70.