Classification of EEG Signals from Four Subjects During Five Mental Tasks

Charles W. Anderson and Zlatko Sijerčić

Department of Computer Science
Colorado State University
Fort Collins, CO 80523
{anderson,sijercic}@cs.colostate.edu
Office: 970-491-7491, FAX: 970-491-2466

Abstract

Neural networks are trained to classify half-second segments of six-channel, EEG data into one of five classes corresponding to five cognitive tasks performed by four subjects. Two and three-layer feedforward neural networks are trained using 10-fold cross-validation and early stopping to control over-fitting. EEG signals were represented as autoregressive (AR) models. The average percentage of test segments correctly classified ranged from 71% for one subject to 38% for another subject. Cluster analysis of the resulting neural networks' hidden-unit weight vectors identifies which EEG channels are most relevant to this discrimination problem.

1 Introduction

Visual inspection of multiple time series of EEG signals in their unprocessed form is still the predominant way of discriminating and classifying EEG patterns in the medical community and requires highly trained medical professionals. Since the early days of automatic EEG processing, representations based on a Fourier transform have been most commonly applied. This approach is based on earlier observations that the EEG spectrum contains some characteristic waveforms that fall primarily within four frequency bands—delta (1–3 Hz), theta (4–7 Hz), alpha (8–13 Hz), and beta (14–20 Hz). Such methods have proved beneficial for various EEG characterizations, but the Fourier Transform and its discrete version, the FFT, suffer from large noise sensitivity. Numerous other techniques from the theory of signal analysis have been used to obtain representations and extract the features of interest for classification purposes. For an overview of these techniques, see [5].

Neural networks and statistical pattern recognition methods have been applied to EEG analysis; some of this work is briefly reviewed in this section. The experiments reported in this article use neural networks to learn EEG classification functions, but go beyond this typical use by applying cluster analysis to the resulting weight vectors. This analysis reveals some of the relationships between EEG features and their classification that exist in the learned classification function. It also provides information that can support and generate hypotheses about brain activity underlying the studied cognitive behaviors. The representations used to code the information in EEG signals greatly influences what can be learned from the analysis of the neural network classifiers.

Some recent work that deals with the problem of EEG representations finds time domain methods based on parametric models very useful for EEG feature extraction. Tseng, et al., [21] evaluated different parametric models on a fairly large database of EEG segments. Using inverse filtering, white noise tests, and one-second EEG segments, they found that autoregressive (AR) models of orders between 2 and 32 yielded the best EEG estimation. For a method which avoids the use of signal segmentation and provides an on-line AR parameter estimation that fits nonstationary signals, like EEG, see [6]. In a problem of classifying EEGs of normal subjects from those with psychiatric disorders, Tsoi, et al., [22] used AR representations in a preprocessing stage and artificial neural networks in the classification stage. Inouye, et al. [9], used the entropy of the power spectra and a mutual information measure to determine directional EEG patterns during mental arithmetic and a resting state. Rotation and change in size of mental images and its corresponding patterns of cerebral activations are considered in [18].

Finding a suitable representation of EEG signals is the key to learning a reliable discrimination and to understanding the extracted relationships [1, 2]. In this article, the coefficients of sixth-order AR models are used to represent the EEG signals. Standard, feed-forward neural networks are trained as classifiers using error backpropagation with early stopping and ten-fold crossover. The representation and training procedure are defined in Section 2. Results are presented in Section 3. Section 4 contains a description and results of the cluster analysis performed on trained networks. Section 5 summarizes the conclusions and limitations of the classification experiments.

2 Method

2.1 EEG Data Acquisition and Representation

All data used in this article was obtained previously by Keirn and Aunon [13, 12] using the following procedure. The subjects were seated in an Industrial Acoustics Company sound controlled booth with dim lighting and noiseless fans for ventilation. An Electro-Cap elastic electrode cap was used to record from positions C3, C4, P3, P4, O1, and O2, defined by the 10-20 system of electrode placement [10]. The electrodes were connected through a bank of Grass 7P511 amplifiers and bandpass filtered from 0.1–100 Hz. Data was and recorded at a sampling rate of 250 Hz with a Lab Master 12 bit A/D converter mounted in an IBM-AT computer. Eye blinks were detected by means of a separate channel of data recorded from two electrodes placed above and below the subject's left eye.

For this paper, the data from four subjects performing five mental tasks was analyzed. These tasks were chosen by Keirn and Aunon to invoke hemispheric brainwave asymmetry [16]. The five tasks are: the *baseline task*, for which the subjects were asked to relax as much as possible; the *letter task*, for which the subjects were instructed to mentally compose a letter to a friend or relative without vocalizing; the *math task*, for which the subjects were given nontrivial multiplication problems, such as 49 times 78, and were asked to solve them without vocalizing or making any other physical movements; the *visual counting task*, for which the subjects were asked to imagine a blackboard and to visualize numbers being written on the board sequentially; and the *geometric figure rotation*, for which the subjects were asked to visualize a particular three-dimensional block figure being rotated about an axis. Data was recorded for 10 seconds during each task and each task was repeated five times per session. Most subjects attended two such sessions recorded on separate weeks, resulting in a total of 10 trials for each task. With a 250 Hz sampling rate, each 10 second trial produces 2,500 samples per channel. These are divided into half-second segments that overlap by one quarter-second, producing at most 39 segments per trial—segments containing eye blinks are discarded.

2.2 AR Representation of EEG Signals

Keirn and Aunon [13] and others [1, 2] achieved the best classification results using a Fourier Transform based on AR coefficients. In this article, the first representation studied is composed of just the AR coefficients. Let $a_{i,c}$ be the i^{th} coefficient of the AR model for channel c, where $c = \{C3, C4, P3, P4, O1, O2\}$ and $i = 1, \ldots, n$ with n being the order of the model. The prediction, $x_{i,c}$, of the order n, AR model is given by

$$x_{i,c}(t) = \sum_{i=1}^{n} a_{i,c} x_{i,c}(t-i).$$

The coefficients that minimize the squared error of this prediction were estimated using the Burg method [11].[1] The AIC criterion is minimized for orders of two and three [20], but based on previous results by Keirn and Aunon, an order of six was used. The 36 coefficients (6 channels x 6 orders) for each segment are concatenated into one feature vector consisting of the six coefficients for the C3 channel, then for the C4, P3, P4, O1, and O2 channels. A total of 1,385 half-second windows compose the 10 trials, with 277 windows from each of the five tasks. Each trial contains the same number of windows from each task, though the trials contain a different total number of windows, ranging from 100 to 175.

2.3 Neural Network Classifier

The classifier implemented for this work is a standard, feedforward, neural network with one or two hidden layers and one output layer, trained with the error backpropagation algorithm [19, 7]. The activation function for all units is the asymmetric sigmoid function. The topology of a network will be denoted by a

[1] The Burg method was implemented using the MATLAB function `ar`. See the Mathworks, Incorporated, web page at http://www.mathworks.com for more information.

hyphenated pair of numbers indicating the number of units in the first hidden layer and in the second hidden layer. For example, a 10-5 network has 10 units in the first hidden layer and 5 in the second. A 10-0 network has 10 units in a single hidden layer.

Training the network is accomplished by initializing all weights to small, random values and then performing a gradient-descent search in the network's weight space for a minimum of a squared error function of the network's output. The error is between the network's output and the target value for each input vector. For the five-task experiments, the target values were set to 1,0,0,0,0 for the baseline task, 0,1,0,0,0 for the letter task, 0,0,1,0,0 for the math task, 0,0,0,1,0 for the counting task, and 0,0,0,0,1 for the rotation task. Different learning rates were used for the hidden layers and the output layer. After trying a large number of different values, we found that a learning rate of 0.1 for the hidden layers and 0.01 for the output layer produced the best performance.

The classification performance of a neural network depends on the initial weight values and on the data used to train and test. If the data contains noise or does not completely specify the target function, a neural network will over-fit the training data and it will not correctly interpolate and extrapolate the training data, i.e., it will not generalize well.

To limit the amount of over-fitting during training, the following 10-fold, cross-validation procedure was performed. Eight of the ten trials were used for the training set, one of the remaining trials was selected for validation and the last trial was used for testing. The error of the network on the validation data was calculated after every pass, or epoch, through the training data. After 3,000 epochs, the network state (its weight values) at the epoch for which the validation error is smallest was chosen as the network that will most likely perform well on novel data. This best network was then applied to the test set; the result indicates how well the network will generalize to novel data. With 10 trials, there are 90 ways of choosing the validation and test trials with the remaining eight trials combined for the training set. Results described in the next section are reported as the average classification accuracy on the test set averaged over all 90 partitions of the data. Each of the 90 repetitions started with different, random, initial weights.

The neural networks were trained using a CNAPS Server II,[2] a parallel, SIMD architecture with 128, 20 MHz, processors, upgradable to 512 processors. The experiments reported in this article were implemented on the CNAPS machine using Buildnet, a library of C-callable functions that implement the error backpropagation algorithm (and others). Training a neural network with a single hidden layer containing 20 hidden units (a 20-0 network) took an average of 3.2 minutes on the CNAPS, while on a Sun SparcStation 20, the same experiment took an average of 20 minutes. An experiment of 90 repetitions required 4.8 hours on the CNAPS and 30 hours on the SparcStation.

3 Results

To illustrate the cross-validation procedure, Figure 1 shows the RMS error after every 300 epochs, averaged over output units and over patterns in the training set, validation set, and test sets, for the three curves, respectively. Though not plotted, the initial RMS error is 0.5, because the initial output of the network is 0.5 and the desired output values are 0 or 1. The training error decreases throughout the training period of 3,000 epochs, but a clear minimum occurs in the validation error. A vertical line is drawn at epoch 396 at which the error for the validation set is the lowest. The error and classification performance on the test set is calculated at that epoch as an indication of how well this network will generalize to novel data. This plot was obtained from a 20-0 network.

This training and testing process was repeated 89 more times for each network topology. To compare various networks, the number of test segments for which the correct output unit produced the highest output are counted and expressed as a percentage of the total number of test segments.

The AR representation was used to determine if two hidden layers provide any advantage over a single hidden layer for this recognition problem. Figure 2a. shows the average percent of the test segments that are classified correctly for various network topologies (h_1 and h_2 are the number of units in the first and second hidden layers). The 90% confidence intervals show that the variation in performance for different values of h_2 are not significant. Thus, two hidden layers provide no advantage. However, the improvement

[2]The CNAPS Server is a product of Adaptive Solutions, Incorporated. See http://www.asi.com for more information.

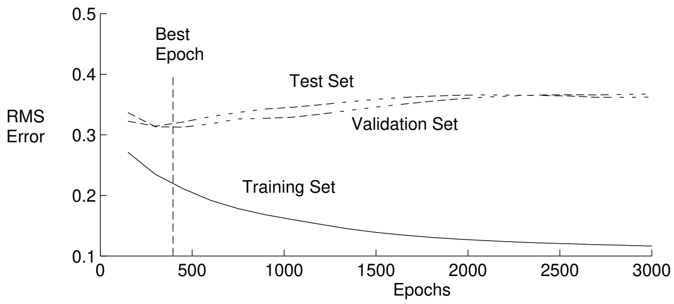


Figure 1: RMS error versus training epochs for training, validation, and test sets.

in performance with higher values of h_1 are significant, until h_1 becomes equal to 10. The best performance is achieved with a 20-0 network, resulting in an average of 54% correct.

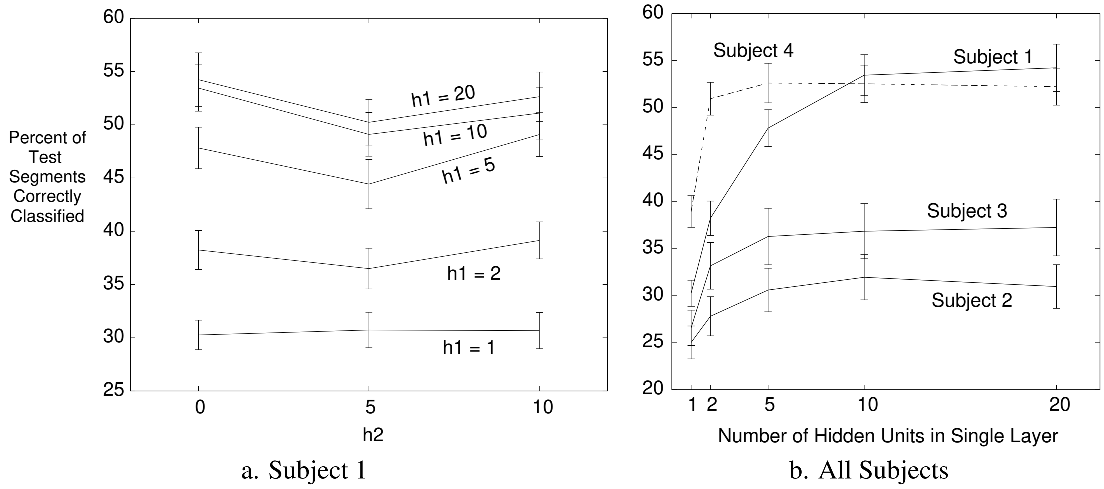
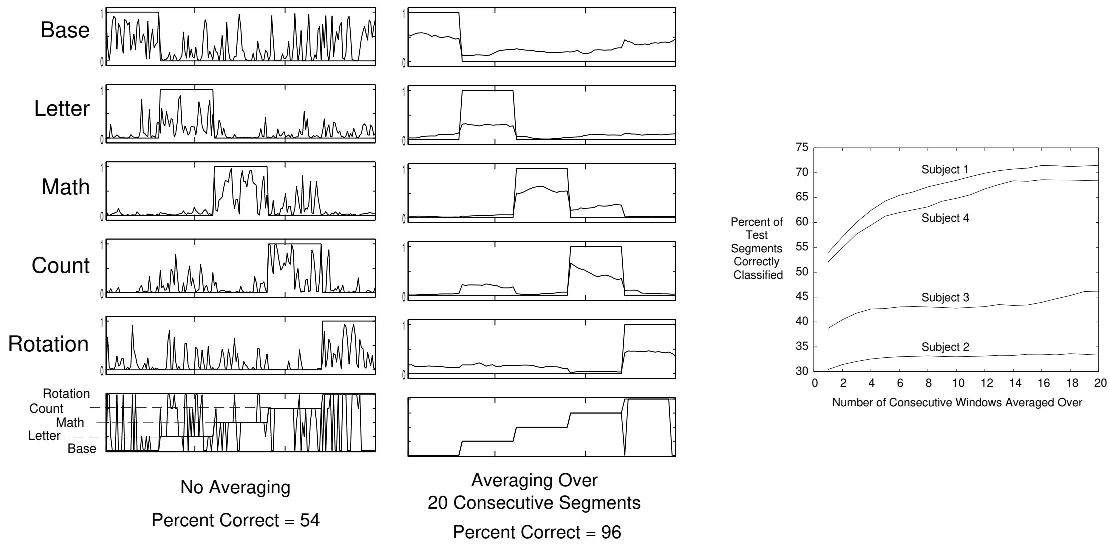


Figure 2: Average percent of test segments correctly classified for different network topologies: a.) Results just for Subject 1, where h_1 is the number of units in the first hidden layer and h_2 is the number in the second hidden layer; error bars show 90% confidence intervals; b.) Results for all subjects with a single hidden layer.

For the remaining experiments, a single hidden layer is used. Figure 2.b summarizes the average percent of test segments classified correctly for various-sized networks using each of the four representations. Again, 90% confidence intervals are included. For Subject 1, better performance results from five or more hidden units. For the other subjects, the increased performance for five or more hidden units is not statistically significant.

Inspection of how the network's classification changes from one segment to the next suggests that better performance might be achieved by averaging the network's output over consecutive segments. To investigate this, a 20-unit network trained with data from Subject 1 is studied. The graphs in the left column of Figure 3a show the output values of the network's five output units for each segment of test data from one trial. On each graph the desired value for the corresponding output is also drawn. The bottom graph shows the true task and the task predicted by the network. For this trial, 54% of the segments are classified correctly when no averaging across segments is performed. The other column of graphs shows the network's output and predicted classification that result from averaging over 20 consecutive segments. Potential confusions that the classifier might make can be identified by the relatively high responses of an output unit for test segments that do not correspond to the task represented by that output unit. For example, in the third graph in



a. Subject 1 b. All Subjects

Figure 3: a.) Network output values and desired values for one test trial for Subject 1. b.) The percent of averaged windows classified correctly versus the number of consecutive windows averaged over. 20 consecutive windows span approximately five seconds.

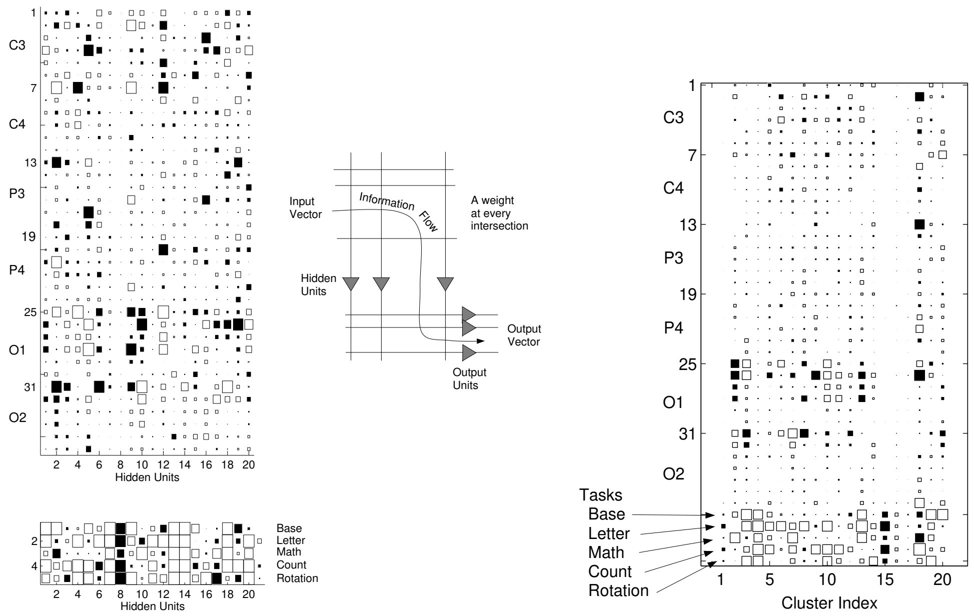
the right column, the output value is high during math segments, as it should be, but it is also relatively high during count segments. Also, the output of the count unit, shown in the fourth graph, is high during count segments, but is also relatively high during letter segments.

For this trial, averaging over 20 segments results in 96% correct, but performance is not improved this much on all trials. The best classification performance for the 20-0 network, averaged over all 90 repetitions, is achieved by averaging over all segments. Figure 3b shows how the percent correct varies with the number of consecutive segments averaged for each subject. Averaging over consecutive segments improves the classification accuracy of data from Subjects 1 and 4 by about 16%, but only improves the accuracy for the other two subjects by about 5%.

4 Analysis of the Neural Network Classifier

One of the networks with 20 hidden units that was trained on Subject 1 data is shown in Figure 4a. Positive weights are drawn as filled boxes, negative weights as unfilled boxes. The width and height of a box is proportional to the weight's magnitude. The weights of the hidden layer are drawn as the upper matrix of boxes and the weights of the output layer are drawn as the lower matrix. The weights of the first hidden unit appear in the left-most column of the upper matrix, while the weights of the first output unit, the one corresponding to the baseline task, are drawn as the first row of the lower matrix.

As an example of how these diagrams can provide clues about what was learned, consider the second hidden unit, i.e., the unit whose weights appear in the second column from the left of the upper matrix and whose output is connected to the units in the output layer through the weights in the second column of the lower matrix. This unit is connected through a strong positive weight to the third output unit, the one corresponding to the math task, and through small magnitude or negative weights to the other output units. Thus, when this hidden unit's output value is high the network more strongly predicts that the input vector is from a math task and less strongly predicts that it is from the other tasks. The most noticeable input weights of this unit are the two pairs of oppositely-signed weights at positions 25 and 31, the components corresponding to the first order coefficients for the O1 and O2 channels. This suggests the math data is related to an asymmetry in the first-order AR coefficients for EEG data recorded from the occipital region (O1 vs. O2). The



a. One network

b. Clustering of multiple networks

Figure 4: a) A 20-0 network. The columns of the upper matrix represent the weights in each hidden unit and the rows of the lower matrix represent the weights in each output unit. Positive weights are filled, negative weights are unfilled. b) Results of k-means clustering for 20 clusters with Subject 1 data.

tenth hidden unit approximately represents the inverse of the effect of the second hidden unit. Hidden unit 12 has relatively large weights for higher orders in the C3, O1, and O2 channels. This unit has the strongest positive effect on the baseline task.

This diagram pertains to just one of the 90 training repetitions. It is obviously too tedious to visually study all 90 repetitions to look for patterns in weight vectors. One approach to dealing with this quantity of information is to cluster the weight vectors from all 90 repetitions. Vectors to be clustered are formed by concatenating the input weights from a hidden unit with its output weights—the weights with which the unit is connected to the five output units. The results of applying the k-means clustering algorithm with $k = 20$, i. e., for 20 clusters, is shown in Figure 4b. The k-means algorithm was initialized by randomly selecting k hidden unit weight vectors as the initial cluster centers.

Five cluster centers (1, 4, 15, 16, and 17) consist of input weights near zero. This indicates that a number of hidden units maintain input weight values near zero during training. The second cluster and the third cluster correspond to the first two types of units described in the above discussion of Figure 4; the second cluster suppresses (is connected negatively to) the math task output unit and the third cluster suppresses all but the math task unit. Most of the other clusters also contain significant weights for the O1 and O2 channels. One cluster that includes large weights in other channels is cluster 18, for which the first order weights are relatively large, positive values for the C3, P3, and O1. These electrodes record from the left hemisphere. This cluster has positive output weights for the baseline, letter, and math tasks, and negative for the counting and rotation task, suggesting a hemispheric asymmetry in the EEG signals related to the first three tasks.

Prior to training all representation components were normalized to have the same mean and variance. This removes biases that would arise from differing input component variances, allowing the direct comparison of the magnitudes of the weights in these clusters.

This demonstrates how the cluster analysis of a large number of resulting weight vectors can lead to an understanding of what relationships the networks have extracted from the data. It also shows how assumptions about the data, such as the removal of known noise sources, can be verified.

The clustering results can also be used to form hypotheses about which representation components are most relevant to the classification problem. These hypotheses can then be used to direct a pruning procedure in which subsets of the least relevant components are removed and the networks retrained. Yet another use of the clusters is to construct a new network consisting of a hidden layer of fixed units whose weights are determined by the clusters. Since the clustering is performed on sets of weight vectors from training runs based on different partitions of the data, the performance of a network constructed of cluster centers might be similar to that achieved by a committee of networks trained on different data partitions.

5 Conclusion

When the output of the network was averaged over 20 consecutive, half-second segments and each segment was represented by sixth-order AR models, approximately 70% of the EEG test patterns for two subjects were classified as the correct mental task. For two additional subjects, 45% and 33% were correctly classified.

Cluster analysis was applied to learned weight vectors, revealing some of the acquired relationships between representation components and mental tasks. The results of clustering can be used both for the construction of lower-dimensional representations and for investigating hypotheses regarding differences in brain activity related to different cognitive behavior.

One of the strengths of this study is its rigorous training procedure involving cross-validation, early stopping, and a large number of training repetitions, made possible by parallel hardware. Early stopping is one of the simplest methods for limiting the complexity of a network. Other methods might lead to better generalization performance. Finnoff, Hergert, and Zimmerman [4] compare the performance of a number of weight decay, pruning, and early stopping methods, on a variety of artificial data sets. They found much variation in the relative rankings of these methods across different data sets, though their results suggest that a simple weight decay mechanism would produce better generalization than the early stopping method applied in this article.

The most likely route to better performance is to test other EEG signal representations. Perhaps performance would be improved by replacing the linear AR predictive model with a nonlinear model and using the coefficients of the nonlinear model as the signal representation. Iasemidis, et al., [8] for example, demonstrate that AR models of EEG during epileptic seizures fail to capture time dependencies that can be captured using new, nonparametric methods developed for analyzing chaotic signals. Fernandez, et al., [3] found significant relationships between mental tasks and features calculated as the relative power in certain frequency bands during the task minus the power during a rest period. Referencing the components of the signal representation to the rest condition might improve the performance reported in this article.

Acknowledgements

This work was supported by the National Science Foundation through grant IRI-9202100.

References

[1] C. W. Anderson, S. V. Devulapalli, and E. A. Stolz. Determining mental state from EEG signals using neural networks. *Scientific Programming*, 4(3):171–183, Fall 1995.

[2] C. W. Anderson, S. V. Devulapalli, and E. A. Stolz. EEG signal classification with different signal representations. In F. Girosi, J. Makhoul, E. Manolakos, and E. Wilson, editors, *Neural Networks for Signal Processing V*, pages 475–483. IEEE Service Center, Piscataway, NJ, 1995.

[3] T. Fernández, T. Harmony, M. Rodríguez, J. Bernal, J. Silva, A. Reyes, and E. Marosi. EEG activation patterns during the performance of tasks involving different components of mental calculation. *Electroencephalography and Clinical Neurophysiology*, 94:175–182, 1995.

[4] W. Finnoff, F. Hergert, and H. G. Zimmerman. Improving model selection by nonconvergent methods. *Neural Networks*, 6(6):771–783, 1993.

[5] A. S. Gevins and A. Rémond. *Methods of Analysis of Brain Electrical and Magnetic Signals*, volume 1 of *Handbook of Electroencephalography and Clinical Neurophysiology (revised series)*. Elsevier Science Publishers B.V., New York, NY, 1987.

[6] S. Goto, M. Nakamura, and K. Uosaki. On-line spectral estimation of nonstationary time series based on ar model parameter estimation and order selection with a forgetting factor. *IEEE Transactions on Signal Processing*, 43(6):1519–1522, June 1995.

[7] M. H. Hassoun. *Fundamentals of Artificial Neural Networks*. The MIT Press, Cambridge, MA, 1995.

[8] Leonidas D. Iasemidis, J. Chris Sackellares, and Robert S. Savit. Quantification of hidden time dependencies in the EEG within the framework of nonlinear dynamics. In Ben H. Jansen and Michael E. Brandt, editors, *Nonlinear Dynamical Analysis of the EEG*, pages 30–47. World Scientific Publishing, Singapore, 1993.

[9] T. Inouye, K. Shinosaki, A. Iyama, and Y. Matsumoto. Localization of activated areas and directional EEG patterns during mental arithmetic. *Electroencephalography and Clinical Neurophysiology*, 86(4):224–230, 1993.

[10] H. Jasper. The ten twenty electrode system of the international federation. *Electroencephalographic Clinical Neurophysiology*, 10:371–375, 1958.

[11] S. M. Kay. *Modern Spectral Estimation: Theory and Application*. Prentice-Hall, 1988.

[12] Zachary A. Keirn. Alternative modes of communication between man and machine. Master's thesis, Purdue University, 1988.

[13] Zachary A. Keirn and Jorge I. Aunon. A new mode of communication between man and his surroundings. *IEEE Transactions on Biomedical Engineering*, 37(12):1209–1214, December 1990.

[14] T. P. Krause, L. Shure, and J. N. Little. *Signal Processing Toolbox*. The Mathworks Inc., Natick, MA, 1994.

[15] Shiao-Lin Lin, Yi-Jean Tsai, and Cheng-Yuan Liou. Conscious mental tasks and their EEG signals. *Medical & Biological Engineering & Computing*, 31:421–425, 1993.

[16] M. Osaka. Peak alpha frequency of EEG during a mental task: Task difficulty and hemispheric differences. *Psychophysiology*, 21:101–105, 1984.

[17] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1988.

[18] F. Rosler, M. Heil, J. Bajric, A.C. Pauls, and E. Henninghausen. Pattern of cerebral activation while mental images are rotated and changed in size. *Psychophysiology*, 32:135–149, 1995.

[19] D. E. Rumelhart, G. E. Hinton, and R. W. Williams. Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland, and The PDP Research Group, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1. Bradford, Cambridge, MA, 1986.

[20] E. Stolz. Multivariate autoregressive models for classification of spontaneous electroencephalogram during mental tasks. Master's thesis, Electrical Engineering Department, Colorado State University, Fort Collins, CO, 1995.

[21] S-Y. Tseng, R-C. Chen, F-C. Chong, and T-S. Kuo. Evaluation of parametric methods in EEG signal analysis. *Med. Eng. Phys.*, 17:71–78, January 1995.

[22] A. C. Tsoi, D. S. C. So, and A. Sergejew. Classification of electroencephalogram using artificial neural networks. In Jack D. Cowan, Gerald Tesauro, and Joshua Alspector, editors, *Advances in Neural Information Processing Systems 6*, pages 1151–1158. Morgan Kaufmann, 1994.