Effects of Variations in Neural Network Topology and Output Averaging on the Discrimination of Mental Tasks from Spontaneous Electroencephalogram

Charles W. Anderson
Assistant Professor
Department of Computer Science
Colorado State University
Fort Collins, CO 80523
anderson@cs.colostate.edu

Abstract

Electroencephalogram, or EEG, signals are an important source of information for the study of underlying brain processes. Such studies now provide a framework for the development of a new modality of human-computer interaction based on EEG. Current research in this area only detects a small number of mental states. In this article, EEG from one subject who performed three mental tasks are classified by neural networks. Using a sixth-order autoregressive (AR) model of half-second windows of six-channel EEG, a classification accuracy of 89% on test data is achieved. A cross-validation study of a variety of neural network topologies showed that a network with one hidden layer of 20 units produced the best performance. It was also found that averaging the output of the network over consecutive inputs improved performance. K-means clustering of the resulting neural networks' weights identified key components of the AR representation.

Keywords: feedforward neural networks; pattern classification; electroencephalogram; mental state; k-means; autoregressive models

1 INTRODUCTION

Most EEG research seeks to understand the dynamic processes in the brain that are the basis of physical and mental behavior. Nunez (1995), Barlow (1993), and Gevins and Rémond (1987) survey the state-of-the-art of this field. In addition to serving as tools to probe the mind, EEG signals are being investigated as a new mode of human-computer communication. If a small number of mental states can be reliably detected, then a person could compose sequences of such states to indicate commands to a computer, just as letters are composed to form words.

In this article, we describe the methods and results of our experiments with EEG signals recorded from one subject while the subject performed three mental tasks—resting, multiplication, and letter composition. The EEG signals were divided into half-second windows and represented by autoregressive, or AR, models fit to each window. We compared the classification performance of one and two-hidden-layer neural networks in a cross-validation paradigm. Our best result was 89% correct classification of untrained data.

The remainder of this article contain the following sections. Section 2 is a summary of work related to ours. Section 3 describes the procedures we followed to record the EEG signals, the instructions to the subject for each mental task, the autoregressive models used to represent the EEG signals, and the neural network training process. Our results are presented in Section 4, followed by the analysis of the resulting neural networks in Section 5. Our conclusions and possible future work are stated in Section 6.

2 RELATED WORK

The work of Keirn and Aunon (1990) (see also, Keirn, 1988) formed the foundation of the new results we present in this article. Keirn and Aunon investigated the classification of five different mental tasks: a baseline resting task, mental multiplication, geometric figure rotation, mental letter composing, and visual counting. Data was recorded from seven subjects using six channels and was transformed into features based on spectral estimates, calculated from both the Fourier transform of the windowed autocorrelation function and a scalar AR model. Features included asymmetry ratios and power values for each channel from four standard frequency bands—delta (1–3 Hz), theta (4–7 Hz), alpha (8–13 Hz), and beta (14–20 Hz). Asymmetry ratios were taken across all right to left combinations of leads and are given by $(R-L)/(R+L)$, where R is a power value from a certain frequency band of a right hemisphere lead, and L is defined similarly for a left hemisphere lead. A second set of features was generated from the AR coefficients themselves concatenated together from all channels. Features were extracted from a single quarter-second (and two-second) window per trial, for ten trials of each of the tasks. Classification was performed with a quadratic Bayesian classifier. By averaging results over five subjects, it was found that all task pairs could be reliably discriminated 84.6% of the time using the AR coefficients as features.

This is an encouraging result, but their study was limited in the following ways. A single quarter-second or two-second window was selected from each 10 second recording session. A window was chosen near the middle of the session, assuming during that period the subject was most likely concentrating on the requested mental task. Another limitation is the use of a quadratic Bayesian classifier, which assumes the classes have a Gaussian distribution. Also, classifiers were constructed and tested on data from single subjects and pairs of tasks. Questions remained regarding generalization across subjects and more than pair-wise discriminations.

In previous work, we extended Keirn and Aunon's work in several ways (Anderson et al., 1995a; Anderson et al., 1995b). The Bayesian classifier was replaced with neural networks of varying size and, thus, complexity. Overlapping half-second windows, that together cover the 10-second period of every recording session, were used. We achieved a classification accuracy of 73% between the baseline and multiplication tasks using Keirn and Aunon's frequency-band representation. We also found that signal representations consisting of untransformed data or a Karhunen-Loève Transform of the data resulted in classification performance that was not significantly better than chance.

The tasks and representations used by Keirn and Aunon were motivated by the work of Doyle et al. (1974), who tested ten subjects performing eight tasks, two of which are primarily mental tasks. Power values were calculated for one-second windows for every frequency from 0 to 29 Hz, and an average power was calculated for the 30-64 Hz range. Each subject repeated the task set twice and the power spectra were averaged for each subject into five bands: delta, theta, alpha, beta 1 (14–20 Hz), and beta 2 (21–29 Hz). Band power values from homologous electrode pairs were combined as ratios to highlight asymmetries in the hemispheres. Results showed significant asymmetries primarily in the power levels of the alpha and beta bands. Temporal electrodes showed more asymmetry than did the parietal electrodes.

Galbraith and Wong (1993) recorded one channel of EEG from 25 subjects during resting and mental arithmetic tasks. Two-second windows were represented by the relative power in the four standard frequency bands and by Gaussian distribution parameters. Using a linear discriminator and a stepwise procedure for eliminating variables, they found that the power in the frequency bands, primarily the alpha band, was most useful in the linear discrimination. Also, the variance of the amplitude distribution proved to be significant in forming the linear discrimination.

Tumey et al. (1991) studied EEG recorded while a subject's eyes were open and alert, closed and alert, and while eyes were closed and the subject performed a visualization task. Data recorded from two channels were fed into a phase-space algorithm where the signal was plotted against a lagged version to generate an attractor pattern over 10 seconds. A box counting algorithm was employed to quantize the attractor into bins and the counts of each bin were composed into a feature vector. A backpropagation neural network (see Section 3) was trained on half of the feature vectors and tested on the remaining half of the data. It was found that 100% of the test vectors were classified correctly, even with test data recorded days after the original experiment.

Other work has focused on detecting patterns in EEG that indicate planned, but not executed, motor actions. For example, Peltoranta and Pfurtscheller (1994) studied finger movement. EEG was band-pass filtered to 5–16 Hz (extended alpha band) and divided into one-second windows. The peak power (power of frequency with maximum power) in each window was calculated using the coefficients of an AR model fit to each window. Model orders from two to six were tested, with little difference between them. Adding the peak frequency did not improve the results. The performance of several classifiers was compared, including LVQ3 (Kohonen, 1995), k-means, and backpropagation neural networks. Their best results—90% correct—were obtained using LVQ3 with one or two reference vectors. The neural network trained with backpropagation was not significantly worse, though required more computational effort to train. Varying the number of hidden units (see Section 3) between 5 and 25 did not have a significant effect on performance. Peltoranta and Pfurtscheller (1994) describe in detail the procedures followed to calculate the signal features and to train the classifiers.

Pfurtscheller et al. (1994) studied the discrimination of four motor tasks, left and right hand finger, toe, and tongue movement. They had previously found the typical event-related desynchronization in the alpha band with finger movement, but had also found an event-related synchronization in the gamma band near 40 Hz. To investigate this further, they compared discrimination performance using features based on power in three bands separately and in combination. They used the LVQ algorithm to classify. Results showed that the bands 10–12 Hz and 38–40 Hz were equally effective, producing 58% correct, and the best results of 70% were obtained when these two bands were combined with 30–33 Hz.

Flotzinger et al. (1994) investigated the effect of several methods for normalizing recorded EEG. They recorded 17 channels from one subject for 1.5 seconds prior to movement of one finger or the other. A visual cue indicated which finger to move. This was repeated for 800 trials. Several methods were compared for determining a reference with respect with which all recorded signals were normalized. These included an average over all electrodes and a local average over neighboring electrodes weighted by distance. After normalizing, signals were bandpass filtered to 9–11 Hz and their power was calculated by squaring the result. Power values were averaged over successive samples for 125, 250, or 500 milliseconds and results compared. Classification was performed with Kohonen's LVQ3 algorithm (Kohonen, 1995). The best classification results on test data were 80% correct. Results were not significantly affected by varying the window size or by the use of the normalization methods. They found that restricting the data to just six channels reduced accuracy by approximately 3%.

3 METHOD

3.1 EEG Signal Recording

We used data obtained previously by Keirn and Aunon (1990; Keirn, 1988) who used the following procedure. The subjects were seated in an Industrial Acoustics Company sound controlled booth with dim lighting and noiseless ventilation fans. An Electro-Cap elastic electrode cap was used to record from positions C3, C4, O1, O2, P3, and P4, shown in Figure 1 and defined by the 10-20 system of electrode placement (Jasper, 1958). These six channels were referenced to electrically linked mastoids at A1 and A2. The impedance of all electrodes was kept below five Kohms. Data were recorded at a sampling rate of 250 Hz with a Lab Master 12 bit A/D converter mounted in an IBM-AT computer. Before each recording session, the system was calibrated with a known voltage. The electrodes were connected through a bank of Grass 7P511 amplifiers with analog bandpass filters from 0.1–100 Hz. 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. An eye blink was defined as a change in magnitude greater than 100 μVolts within a 10 milliseconds period.

— *Insert figure 1 about here.* —

We analyzed the data from one subject performing the following three mental tasks. The subject, selected arbitrarily from Keirn and Aunon's data, was a 48-year-old, left-handed, male, university employee. All tasks were performed with the subject's eyes open. The tasks were chosen by Keirn and Aunon to invoke hemispheric brainwave asymmetry (Osaka, 1984). The three tasks were:

Baseline Task: The subject was not asked to perform a specific mental task, but to relax as much as possible and think of nothing in particular. This task is considered a baseline task for alpha wave production and was used as a control measure of the EEG.

Letter Task: The subject was instructed to mentally compose a letter to a friend or relative without vocalizing. Since the task was repeated several times, the subject was told to try to pick up where they left off in the previous task.

Math Task: The subject was given nontrivial multiplication problems, such as 49 times 78, and was asked to solve them without vocalizing or making any other physical movements. The problems were not repeated and were designed so that an immediate answer was not attainable. The subject was asked after each trial whether or not they found the answer; no problem was completed before the end of the 10-second recording trial.

Data were recorded for 10 seconds during each task and each task was repeated five times per session. The subject attended two such sessions recorded during separate weeks, resulting in a total of 10 trials for each task.

3.2 AR Representation of EEG Signals

With a 250 Hz sampling rate, each 10-second trial produced 2,500 samples per channel. They were divided into half-second windows that overlapped by a quarter-second, producing 39 windows per trial. Samples from the first half-second of a baseline, letter, and math trial are shown in Figure 2. Data from all six channels are superimposed. As described in the next section, each half-second window was classified independently.

— *Insert figure 2 about here.* —

Keirn and Aunon (1990) and others (Anderson et al., 1995a; Anderson et al., 1995b) achieved the best classification results using a Fourier Transform based on AR coefficients. For the following experiments, we used the AR coefficients directly to represent the data in each window. To define the AR model, let $a_{i,c}$ be the i^{th} coefficient of the AR model for channel c, where $c = \{C3, C4, O1, O2, P3, P4\}$ and $i = 1, \ldots, n$ with n being the order of the model. The order n AR model of the 125 samples, $x_{i,c}$, from channel c in a window is given by

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

The coefficients were estimated using the Burg method (Kay, 1988), implemented by the MATLAB[1] function `ar`. The AIC criterion is minimized for orders of two and three, but based on previous results

[1]MATLAB is programming environment by Mathworks, Incorporated. For more information, see http://www.mathworks.com.

by Keirn and Aunon, an order of six was used. The 36 coefficients (6 channels x 6 orders) for each window were concatenated into one feature vector.

$$(a_{1,C3}, a_{1,C4}, a_{1,O1}, a_{1,O2}, a_{1,P3}, a_{1,P4}, a_{2,C3}, a_{2,C4}, \ldots, a_{6,P4}).$$

Figure 3 shows the AR representation of 26 consecutive half-second windows of data from one trial of each of the three tasks. For the trial shown in the figure, 26 of the 39 windows are eye-blink free. The width and height of a box is proportional to the coefficient's magnitude. Positive and negative coefficients are shown with filled and unfilled boxes, respectively. Each column depicts the 36 coefficients from one window in the order given above. The first order coefficients have the largest magnitude in all three tasks. The coefficients associated with the O1 and O2 electrodes are smaller on average. There is considerable variation in the coefficients from one window to the next; no obvious difference is apparent between the tasks. The additional nine trials of data are not shown. A total of 843 half-second windows compose the 10 trials with 281 windows from each of the three tasks. Each trial contains the same number of windows from each task though the trials contain a different total number of windows ranging from 60 to 114.

— Insert figure 3 about here. —

The AR data was normalized before being submitted to the classification experiments to minimize the differences in variation among the coefficients. The normalization procedure scaled each component of the AR coefficient vectors independently so that each component had a mean of 0.5 and a standard deviation of 1/6. Values less than 0 or greater than 1 were set to 0 or 1, respectively. The [0,1] range is required for training with the Buildnet library for the CNAPS computer (described later in this section). The result of this normalization on the data in Figure 3 is shown in Figure 4. To highlight differences, 0.5 was subtracted from all values in Figure 4 before displaying them.

— Insert figure 4 about here. —

3.3 Neural Network Classifier

The classifier implemented for this work was a standard, feedforward, neural network (see Figure 5) with one or two hidden layers and one output layer, trained with the error backpropagation algorithm. The topology of a network is denoted by a hyphenated pair of numbers indicating the number of units in the first hidden layer and the number of units 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. All networks had three units in the output layer, one for each mental task classification. The error backpropagation algorithm is briefly summarized below; details can be found in Rumelhart et al. (1986) and recent textbooks on artificial neural networks, such as Hassoun (1995).

— Insert figure 5 about here. —

The components of an input vector composed of 36 AR coefficients for one window were distributed to each unit in the first hidden layer. All of the units had weight vectors whose components were multiplied by the input components. Each unit summed these weighted inputs and produced a value that was transformed by a nonlinear activation function, for which we used the common asymmetric sigmoid function. The second hidden layer, if there was one, accepted as input the activations of the first hidden layer and computed its output activations. The output of the final layer was then computed by multiplying the output vector from the hidden layer by the weights into the final layer. More summations and activations at these units gave the actual output of the network. Given an input vector of AR coefficients, the network's output was calculated, and the network's classification of the input vector was indicated by which of the three output units had the largest output value.

The network was trained 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 was between the network's output and the target value for each input vector. For the three-task experiments, the three target values were set to 1, 0, and 0 for the baseline task; 0, 1, and 0 for the letter task; and 0, 0, and 1 for the math task.

The error backpropagation algorithm is derived by decomposing the gradient calculation into computations performed in each layer, starting with the final layer and passing results backwards through the network. The amount by which the weights are adjusted on each step is parameterized by learning rate constants. We used one learning rate for the hidden layers and a different rate for the output layer. After trying a large number of different values, we found that a learning rate of 0.1 for the hidden layer 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 it. 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 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 4,000 epochs, the network state (its weight values) at the epoch for which the validation error was smallest was chosen as the network that would most likely perform well on novel data. This best network was then applied to the test set and the result indicated how well the network would 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 from Adaptive Solutions, Incorporated.[2] Our CNAPS system is a parallel, SIMD architecture with 128, 20 MHz, processors, upgradable to 512 processors. It can be programmed at three levels, using assembly language, C with parallel programming extensions, or existing libraries that implement standard error backpropagation and other algorithms. The experiments described here were performed with a combination of MATLAB and C programs and Adaptive Solutions' Buildnet library of error backpropagation routines for the CNAPS server. 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. On a Sun SparcStation 20, training took an average of 20 minutes. An experiment with 90 repetitions required 4.8 hours on the CNAPS and 30 hours on the SparcStation. Implementation details are described by Anderson et al. (1995a).

4 RESULTS

To illustrate the cross-validation procedure, Figure 6 shows the RMS error, 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 was 0.5, because the initial output of the network was 0.5 and the desired output values were 0 or 1. The training error decreased throughout the training period of 4,000 epochs, but a clear minimum occured in the validation error. A vertical line was drawn at epoch 481 at which the error for the validation set was the lowest. The error and classification performance on the test set was calculated at that epoch as an indication of how well this network would generalize to novel data. This plot was obtained from a 10-0 network.

— Insert figure 6 about here. —

This training and testing process was repeated 89 more times for each network topology. To compare various networks, the number of times each output value was on the correct side of 0.5 was counted and expressed as a percent of the number of test patterns. A network whose three output values were constant zero (or constant one) would result in a percent correct of 67%. Figure 7 shows the distribution of percent correct values from the 90 repetitions with each network topology. The graphs in rows differ by the number of units in the first hidden layer while the graphs in columns differ by the number of units in the second hidden layer. The dark vertical line drawn on each histogram is at the distribution's mean.

— Insert figure 7 about here. —

These histograms clearly show that performance increased with a higher number of units in the first hidden layer, though as the number increased beyond five units the performance increase was greatly reduced. Significant differences are determined by considering Figure 8 in which the average percents correct and their 90% confidence intervals are plotted. Separate lines are plotted for different numbers of units in the first hidden layer, and the horizontal axis shows the number of units in the second layer.

[2] See http://www.asi.com for more information.

This graph shows that the number of second layer units did not affect performance. However, differences between networks with 1, 2, and 5 units in the first layer were significant. The best performance was achieved with a 20-0 network, resulting in 83% correct.

— Insert figure 8 about here. —

The performance measure of most interest to us is the percent of EEG test windows actually classified correctly. To calculate this, we counted the number of windows for which the output unit with the highest output corresponded to the correct task. In this case, if all outputs were a constant zero (or constant one), then approximately one third of the windows would be classified correctly. The best value for this measure of performance was again obtained with the 20-0 network. It resulted in 76% correct classification averaged over all test windows and over the 90 repetitions.

Inspection of how the network's classification changed from one window to the next suggested that better performance might be achieved by averaging the network's output over consecutive windows. The left column of graphs in Figure 9 show the output values of the network's three output units for each window of a 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, 79% of the windows were classified correctly. Most of the errors occurred for letter task windows, and no errors occurred for baseline task windows. The two other columns of graphs show the network's output and predicted classification that resulted from averaging over four and eight consecutive windows. For this trial, averaging over 10 windows resulted in 100% correct, but performance was not improved that much on all trials. In fact, the best classification performance of 89% was obtained by averaging the output of the 20-0 network over 10 consecutive windows.

— Insert figure 9 about here. —

Figure 10 shows how the percent correct varied with the number of consecutive windows averaged for 20-0, 10-0, and 5-0 networks. The percent correct decreases quickly as the number of windows grows beyond 35, because few trials contained 35 or more windows. For those trials that did contain more than 35 windows, averaging the outputs over that many windows resulted in lower performance.

— Insert figure 10 about here. —

5 ANALYSIS OF THE NEURAL NETWORK CLASSIFIER

To understand what was learned by a neural network, symbolic rules can be extracted that capture some of the information (e.g., Alexander and Mozer, 1995). However, we find it is more useful to numerically analyze our neural network classifiers for our EEG classification problem. We used graphical and numerical tools to interpret what was learned by the 20-0 network. First, we depict the weights graphically, we investigate the dimensionality of the representation provided by the hidden layer, and, finally, we cluster the hidden units' weight vectors to identify the most common hidden unit weights across all repetitions.

Two of the 90, 20-0 networks are shown in Figure 11. 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 18th hidden unit in the left network. It is connected through a strong positive weight to the output unit corresponding to the math task, and negative weights to the other output units. Thus, this hidden unit's output probably tends to be high for math tasks and low for the other two tasks. The most noticeable input weights of this unit are the two pairs of oppositely-signed weights on the inputs corresponding to the first and second order coefficients for the O1 and O2 channels.

— Insert figure 11 about here. —

There is much variation in the weights between the two networks. How many of these hidden units are significant to the classification task? One way to answer this question is to consider the dimensionality of the space defined by the hidden layer output vectors given a test set of input vectors. If this space is lower than the number of hidden units, then all hidden units are not necessary. To investigate this, the eigenvalues of the covariance matrix of hidden layer output vectors was calculated. Figure 12 is a plot of the eigenvalues, divided by the maximum eigenvalue for each repetition and sorted from largest to smallest. For each eigenvalue index, the maximum, mean, and minimum eigenvalue over all 90 networks resulting from the 90 repetitions is plotted. If we consider only normalized eigenvalues greater than 0.01 to be significant (Sirovich, 1989), then the figure shows that the number of significant eigenvalues ranged from 7 to 16, with a mean of 11.

— Insert figure 12 about here. —

The hidden layer transformation clearly produced a representation with a number of significant dimensions much less than 20, the number of hidden units in this case. What are the most common hidden unit weight vectors? One way to answer this is to look for the most common weight vectors. A simple approach to determine this is to apply the k-means clustering algorithm to the set of hidden unit weight vectors collected from all 90 repetitions. Figure 13 shows the results for clustering with $k = 3$ and $k = 5$, i.e., 3 and 5 clusters. The k-means algorithm was initialized by randomly selecting k hidden unit weight vectors as the initial cluster centers. The $k = 3$ results were very consistent over different initial cluster centers. One cluster center for both cases consisted of weights near zero. This indicates that the components of a fair number of weight vectors remained near zero during training. The other two vectors for the $k = 3$ case contained only four weights of significant magnitude. These weights were associated with the first and second order coefficients for the O1 and O2 electrodes. The fact that the O1 and O2 weights for a particular order were of different signs suggests that an asymmetry in the AR coefficients across hemispheres in the occipital region is relevant to the baseline, letter, and math discrimination problem.

— Insert figure 13 about here. —

To understand how the O1 and O2 coefficients might be related to the tasks, we searched for vectors similar to the cluster centers in trained networks, such as those in Figure 6. Hidden unit 18 in the left network of the figure is similar to the first cluster center in both $k = 3$ and $k = 5$ cases. It is strongly connected with a positive weight to the math task output unit and with negative weights to the other tasks.

6 CONCLUSION

EEG signals recorded from a subject performing three mental tasks was discriminated with an 89% accuracy using an AR representation of the EEG and a multilayer neural network classifier. This required averaging the output of the classifier over 10 consecutive half-second windows of data, amounting to five seconds of EEG data. This is a fairly high accuracy, but five seconds may be too much time for this method to be the basis of a practical human-computer interface.

Analysis of the neural networks trained to perform this discrimination task generated more questions than it answered. Clustering of the weight vectors of trained hidden units suggested that the first and second order AR coefficients for the O1 and O2 electrodes were most relevant. The clustering results could be used to prune away input components that don't appear to be relevant. The clustering results could also be used to seed the hidden unit weights with initial values before training. To test these ideas, the discrimination experiments must be repeated using only those AR coefficients deemed significant by the analysis. This might result in better generalization to novel data. Other approaches to pruning out insignificant parameters in the neural network might also lead to better generalization.

The discrimination problem studied here is really a problem of detection—given a window of EEG data we want to know the probability that the data was generated by a person doing each of the mental tasks. A probability threshold could then be varied to find a satisfactory balance between false and true detections. The procedure described here could be modified to do this by using an output layer of the neural network that generates a multinomial distribution. Then, with a small change to the error backpropagation procedure, the network could be trained to approximate the a posterior task probabilities (Rumelhart et al., 1995). This is currently being investigated.

The most likely avenue to better discrimination accuracy is to continue the search for a better representation of the EEG signals. Here, we used linear predictive models to reduce the dimensionality of the signals. Perhaps a nonlinear predictive model would capture more of the information in the signals relevant to the discrimination problem. If a neural network is used as the nonlinear predictor, the complexity of the model could be controlled by varying the number of hidden units.

The main limitations of our work are the small number of mental tasks and the use of data from only one subject. A key question that remains is, to what degree does the AR representation capture task-related information that is invariant across subjects? To-date, attempts to discriminate between mental states across subjects have been unsuccessful (Childers et al., 1987; Lin et al., 1993). If such an invariant representation cannot be found, then a human-computer interface that has an EEG component must be trained by each user, lowering its practicality.

Recent developments in hardware for acquiring EEG signals and in parallel hardware make the construction of a portable, experimental device feasible. A number of data acquisition devices are available that can record multichannel EEG using a desktop or portable PC. If an EEG-based interface must be trained for each user and a low-dimensional representation cannot be found, then parallel hardware might be required to adapt the interface to a user in a reasonable amount of time. The CNAPS architecture that we used is currently available as a PC board, so the training procedure used here can be directly transferred to a PC-based system.

Acknowledgements

The AR representation was developed by Erik Stolz and Dr. Sanyogita Shamsunder of the Electrical Engineering Department at Colorado State University (CSU). Charles Martin of the Computer Science Department at CSU wrote functions in MATLAB (from Mathworks, Inc.) for displaying neural network weights and outputs. The suggestions from several anonymous reviewers significantly improved the clarity of this article. This work was supported by the National Science Foundation through grant IRI-9202100.

References

Alexander, J. A., and Mozer, M. C. 1995. Template-based algorithms for connectionist rule extraction, in: *Advances in neural information processing systems, volume 7*, edited by Tesauro, G., Touretzky, D., and Leen, T., Cambridge, MA, MIT Press, 609–616.

Anderson, C. W., Devulapalli, S. V., and Stolz, E. A. 1995a. Determining mental state from EEG signals using neural networks, *Scientific programming*, **4**(3), 171–183.

Anderson, C. W., Devulapalli, S. V., and Stolz, E. A. 1995b. EEG signal classification with different signal representations, in: *Neural networks for signal processing V*, edited by Girosi, F., Makhoul, J., Manolakos, E., and Wilson, E., Piscataway, NJ, IEEE Service Center, 475–483.

Barlow, J. S. 1993. *The electroencephalogram: Its patterns and origins*, Cambridge, MA, MIT Press.

Childers, D. G., Perry, N. W., Fischler, I. A., Boaz, T., and Arroyo, A. A. 1987. Event-related potentials: A critical review of methods for single-trial detection, *CRC critical reviews in biomedical engineering*, **14**(3), 185–200.

Doyle, J. C., Ornstein, R., and Galin, D. 1974. Lateral specialization of cognitive mode: II, EEG frequency analysis, *Psychophysiology*, **11**(5), 567–578.

Flotzinger, D., Pfurtscheller, G., Neuper, C., Berger, J., and Mohl, W. 1994. Classification of non-averaged EEG data by learning vector quantisation and the influence of signal preprocessing, *Medical and biological engineering and computing*, **32**, 571–576.

Galbraith, G. C., and Wong, E. H. 1993. Moment analysis of EEG amplitude histograms and spectral analysis: Relative classification of several behavioral tasks, *Perceptual and motor skills*, **76**, 859–866.

Gevins, A. S., and Rémond, A. 1987. *Methods of analysis of brain electrical and magnetic signals*, Volume 1 of *Handbook of electroencephalography and clinical neurophysiology (revised series)*, New York, NY, Elsevier Science Publishers B.V.

Hassoun, M. H. 1995. *Fundamentals of artificial neural networks*, Cambridge, MA, The MIT Press.

Jasper, H. 1958. The ten twenty electrode system of the international federation, *Electroencephalography and clinical neurophysiology*, **10**, 371–375.

Kay, S. M. 1988. *Modern spectral estimation: Theory and application*, Prentice-Hall.

Keirn, Z. A., and Aunon, J. I. 1990. A new mode of communication between man and his surroundings, *IEEE transactions on biomedical engineering*, **37**(12), 1209–1214.

Keirn, Z. A. 1988. Alternative modes of communication between man and machine, Master's thesis, Purdue University.

Kohonen, T. 1995. *Self-organizing maps*, Berlin Heidelberg, Springer-Verlag.

Lin, S.L., Tsai, Y.J., and Liou, C.Y. 1993. Conscious mental tasks and their EEG signals, *Medical and biological engineering and computing*, **31**, 421–425.

Nunez, P. L. 1995. *Neocortical dynamics and human EEG rhythms*, New York, Oxford University Press.

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

Peltoranta, M., and Pfurtscheller, G. 1994. Neural network based classification of non-averaged event-related EEG responses, *Medical and biological engineering and computing*, **32**, 189–196.

Pfurtscheller, G., Flotzinger, D., and Neuper, C. 1994. Differentiation between finger, toe and tongue movement in man based on 40 Hz EEG, *Electroencephalography and clinical neurophysiology*, **90**, 456–460.

Rumelhart, D. E., Hinton, G. E., and Williams, R. W. 1986. Learning internal representations by error propagation, in: *Parallel distributed processing: Explorations in the microstructure of cognition*, Volume 1, edited by Rumelhart, D. E., McClelland, J. L., and PDP Group, Cambridge, MA, Bradford, 318–362.

Rumelhart, D. E., Durbin, R., Golden, R., and Chauvin, Y. 1995. Backpropagation: The basic theory, in: *Backpropagation: Theory, architectures, and applications*, edited by Chauvin, Y., and Rumelhart, D. E., Hillsdale, NJ, Lawrence Erlbaum Associates, Inc., 1–34.

Sirovich, L. 1989. Chaotic dynamics of coherent structures, *Physica D*, number 126.

Tumey, D. M., Morton, P. E., Ingle, D. F., Downey, C. W., and Schnurer, J. H. 1991. Neural network classification of EEG using chaotic preprocessing and phase space reconstruction, in: *Proceedings of the 1991 IEEE seventeenth annual northeast bioengineering conference*, 51–52.

Figure Captions

Figure 1: Placement of the electrodes according to the 10-20 system.

Figure 2: One-half second of data from one subject performing each task. Data from the C3, C4, O1, O2, P3, P4 channels are superimposed.

Figure 3: AR representation of eye-blink free windows from one trial of data from one subject performing each task. Positive coefficients are shown as filled boxes and negative coefficients are unfilled. The width and height of a box is proportional to the magnitude of the corresponding coefficient. The highest magnitude coefficients across tasks are the first order coefficients.

Figure 4: Normalized AR representation of data shown in Figure 6. Normalized data lie in the range [0,1], but for this figure the mean of 0.5 is subtracted from all values.

Figure 5: Feedforward neural network with one or two hidden layers.

Figure 6: RMS error versus training epochs for training, validation, and test sets. The epoch at which the error on the validation data is lowest is 481. The network's weights at epoch 481 are saved as the "best" weights. The error on the test set using these best weights is designated as the generalization error of the network.

Figure 7: Distributions of the percent of test patterns correctly classified for each network topology. Each histogram represents 90 repetitions differing in initial weight values and division of data into training, validation, and test sets. The mean of each distribution is drawn as a dark vertical line. These distributions are summarized in Figure 8 by their means and confidence intervals.

Figure 8: Average percent of test patterns correctly classified with error bars showing 90% confidence intervals. Figure 7 shows the actual distributions of the 90 samples used to calculate the means and confidence intervals.

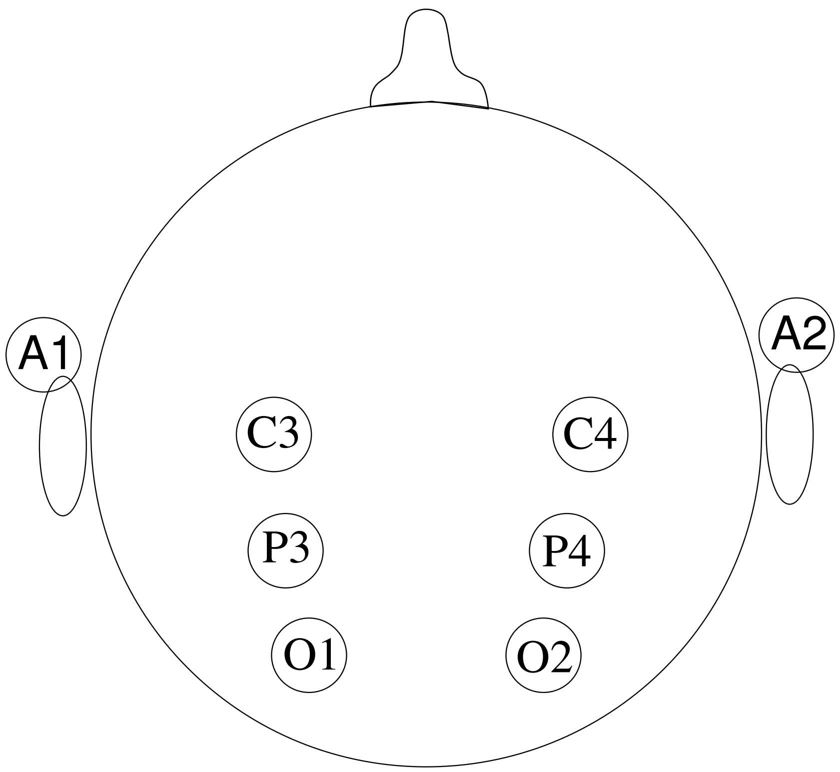
Figure 9: Network output values and desired values for one test trial. The first three rows of graphs show the values of the three network outputs over the 105 test windows. Windows from the three tasks are concentrated in each graph, hiding the rest period between the recording of data from each task. Also, the order of the tasks does not reflect the order in which the tasks were performed. The fourth row of graphs plots the true task and the task predicted by network, determined by the output unit having the largest output. The first column of graphs is without averaging over consecutive windows, the second is for averaging the network output over four consecutive windows, while the third column is for averaging over eight windows.

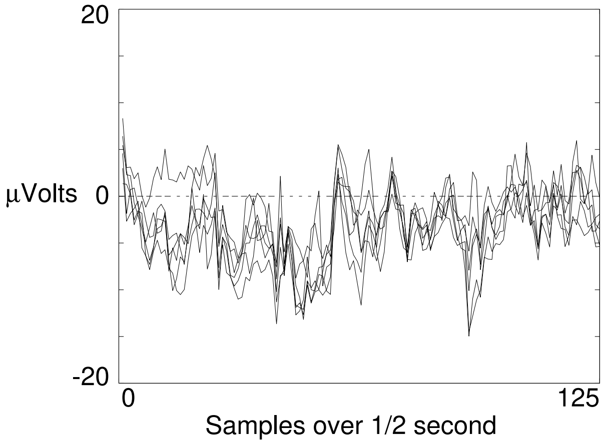
Figure 10: The fraction of averaged windows classified correctly versus the number of consecutive windows averaged over.

Figure 11: Two 20-0 networks trained on different partitions of the data. In each 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.

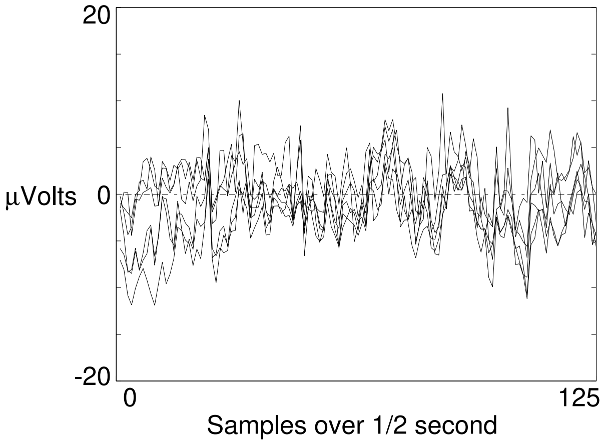
Figure 12: Range of normalized eigenvalues of covariance matrix of hidden layer outputs for test set over 90 repetitions. Values are normalized by dividing by the maximum eigenvalue for each repetition. The horizontal line is at a value of 0.01, below which eigenvalues are considered to be insignificant.

Figure 13: Results of k-means clustering for three and give clusters.

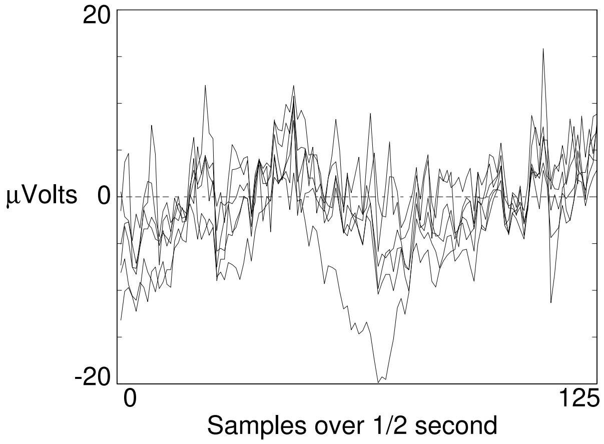

A1
A2
C3
C4
P3
P4
O1
O2



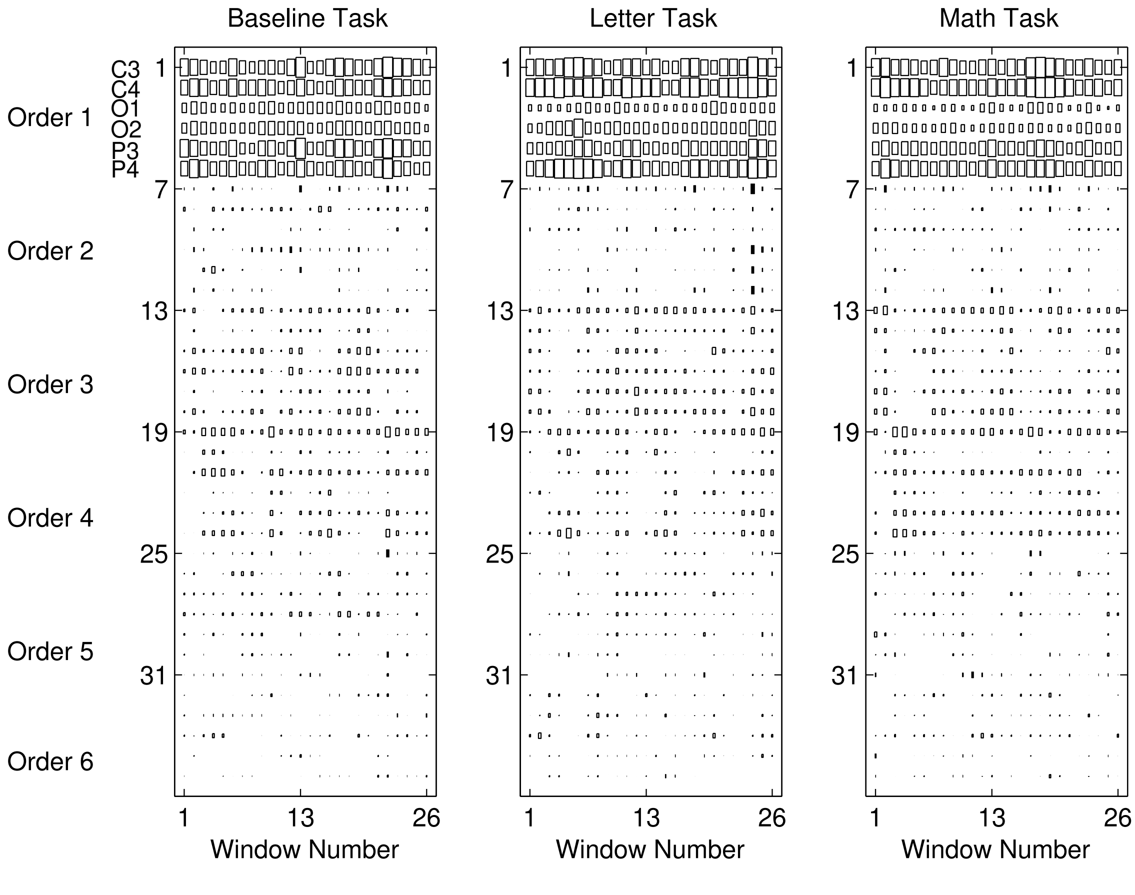
Baseline Task



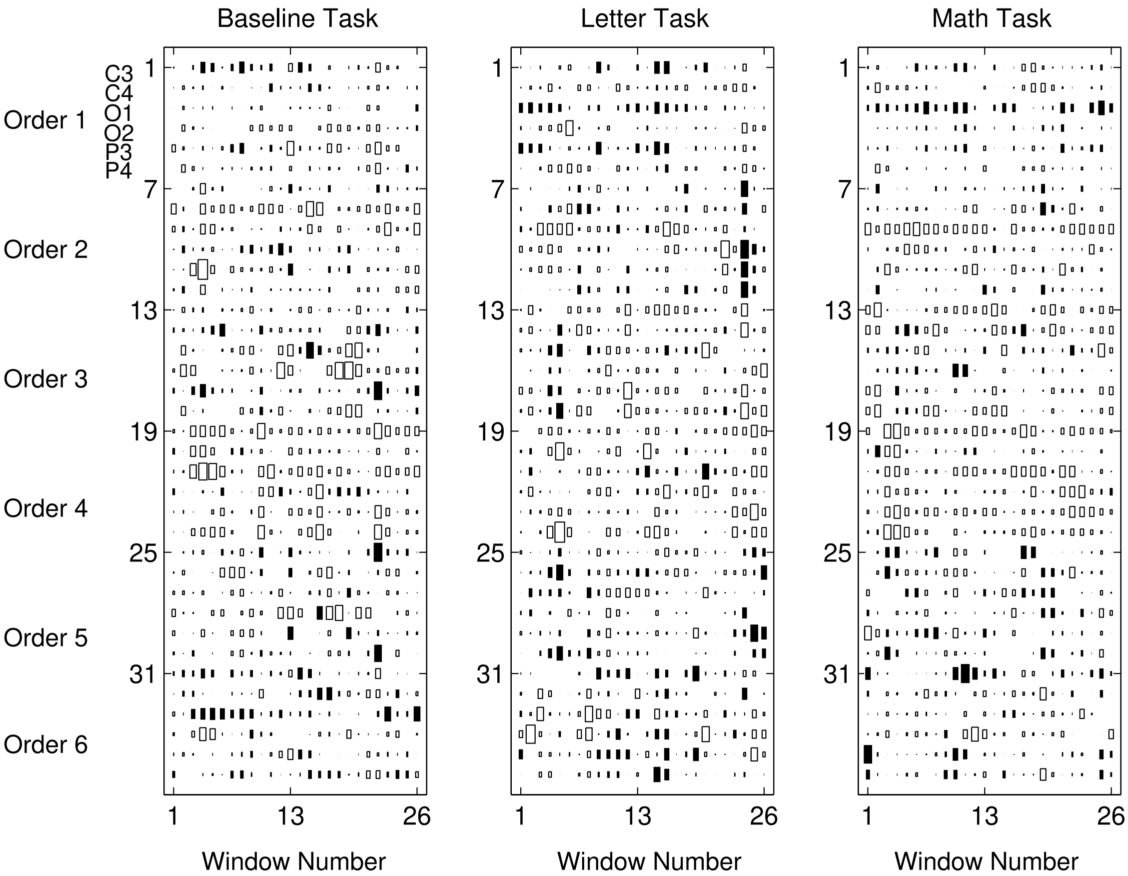
Letter Task



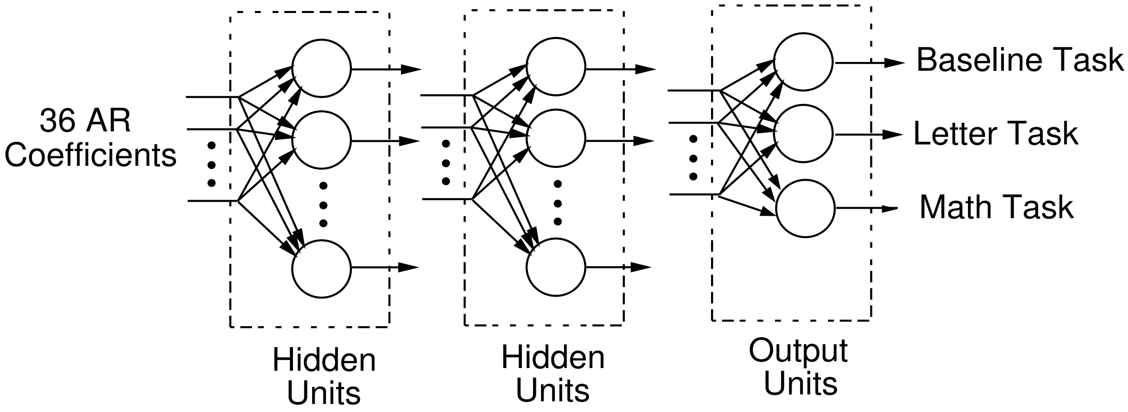
Math Task

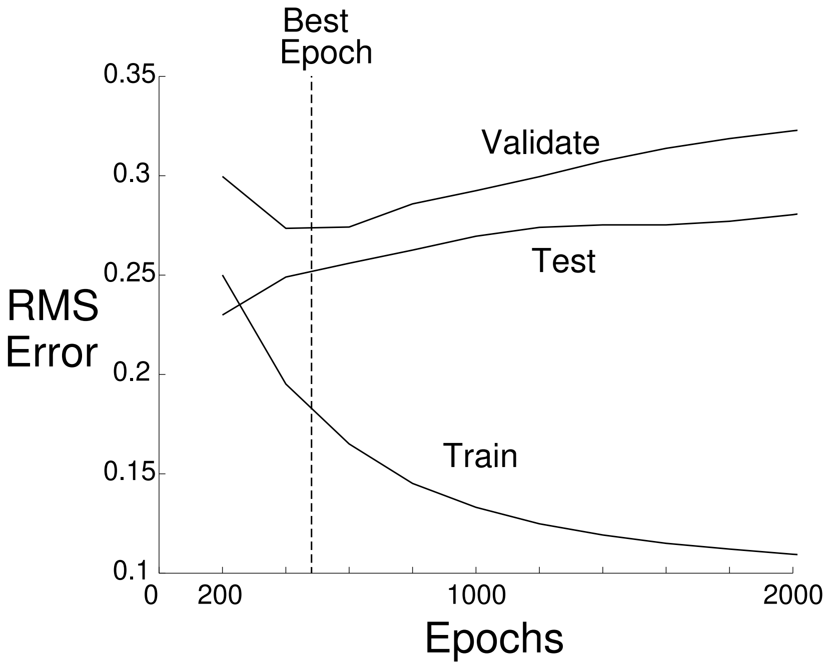


Baseline Task
Letter Task
Math Task
C3
C4
O1
O2
P3
P4
Order 1
Order 2
Order 3
Order 4
Order 5
Order 6
1
7
13
19
25
31
1
13
26
Window Number

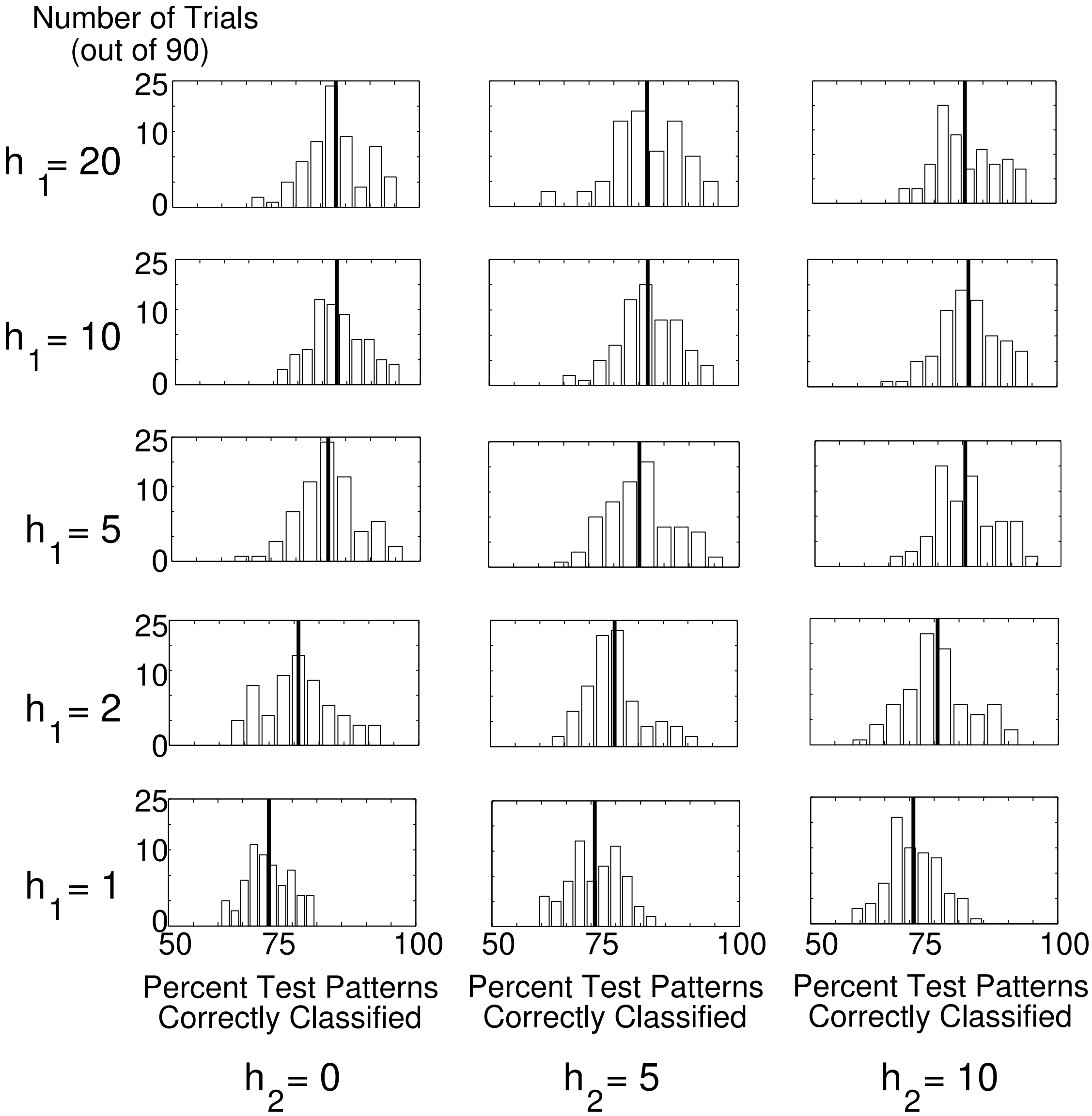


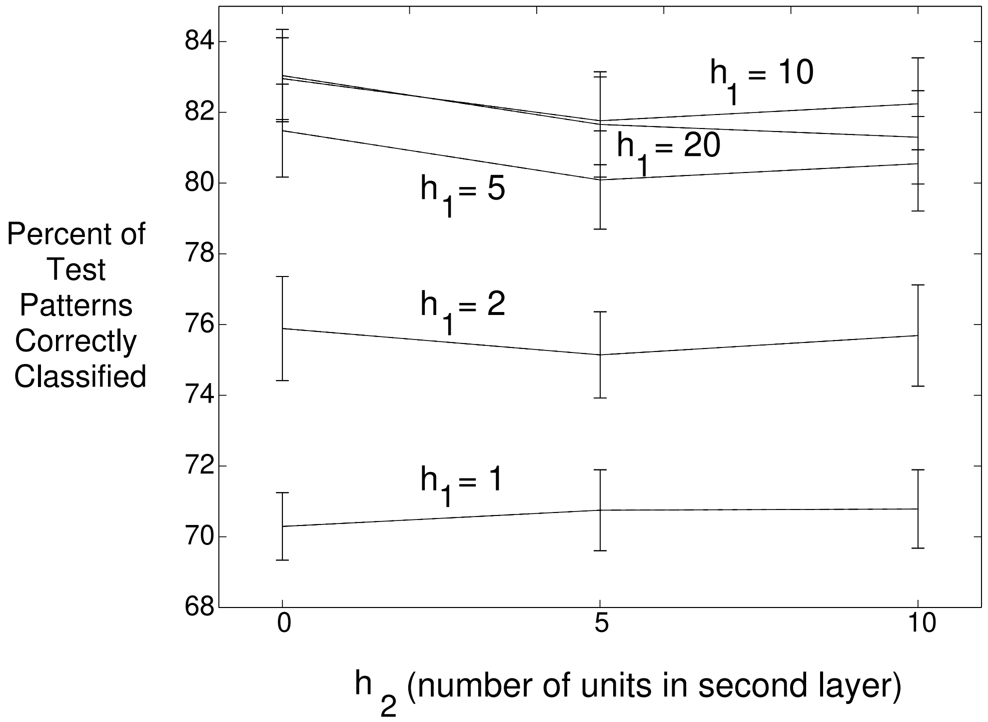
Baseline Task
Letter Task
Math Task
Order 1
Order 2
Order 3
Order 4
Order 5
Order 6
C3
C4
O1
O2
P3
P4
1
7
13
19
25
31
1
13
26
Window Number


36 AR
Coefficients
Hidden
Units
Hidden
Units
Output
Units
Baseline Task
Letter Task
Math Task

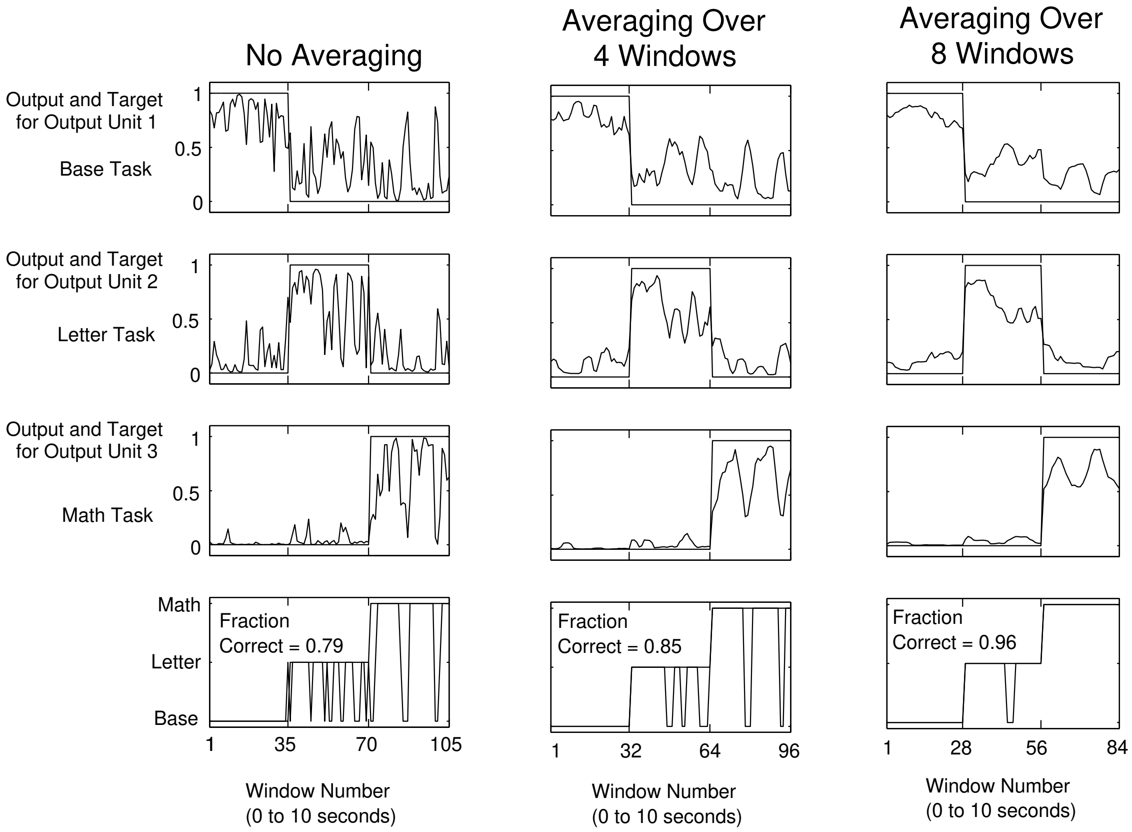


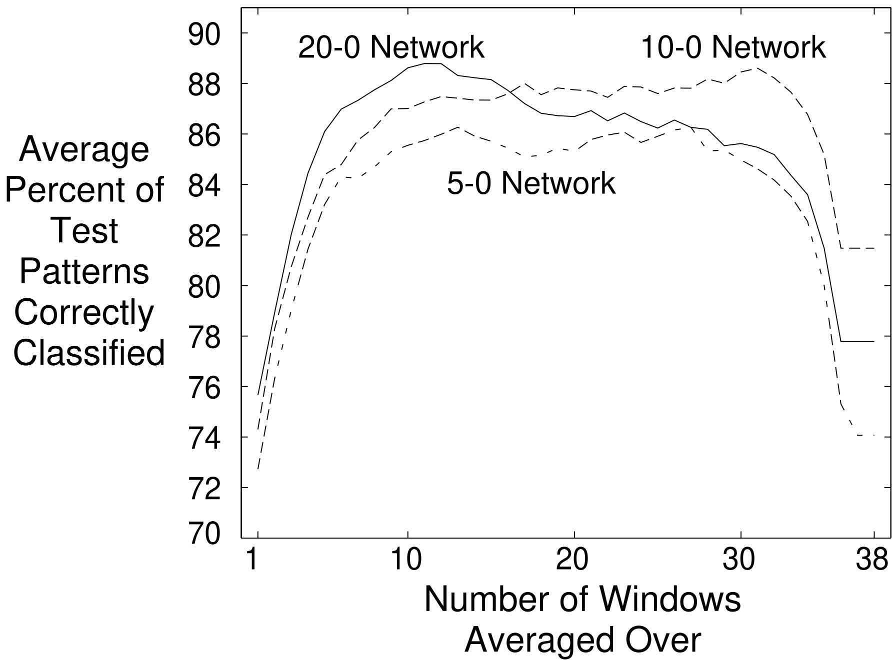
Best
Epoch
0.35
0.3
0.25
0.2
0.15
0.1
RMS
Error
Validate
Test
Train
0
200
1000
2000
Epochs


Number of Trials
(out of 90)
25
10
0
$h_1 = 20$
$h_1 = 10$
$h_1 = 5$
$h_1 = 2$
$h_1 = 1$
50
75
100
Percent Test Patterns
Correctly Classified
$h_2 = 0$
$h_2 = 5$
$h_2 = 10$

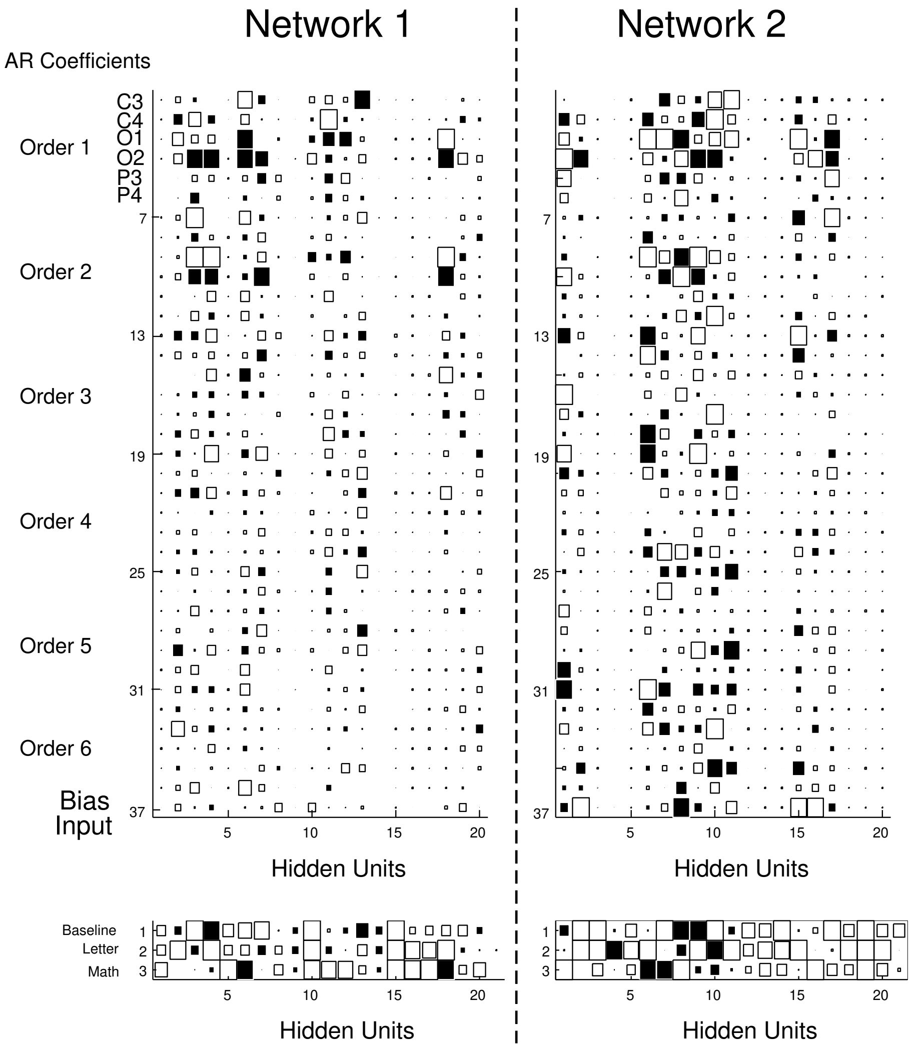


Percent of
Test
Patterns
Correctly
Classified
84
82
80
78
76
74
72
70
68
0
5
10
h₁ = 10
h₁ = 20
h₁ = 5
h₁ = 2
h₁ = 1
h 2 (number of units in second layer)

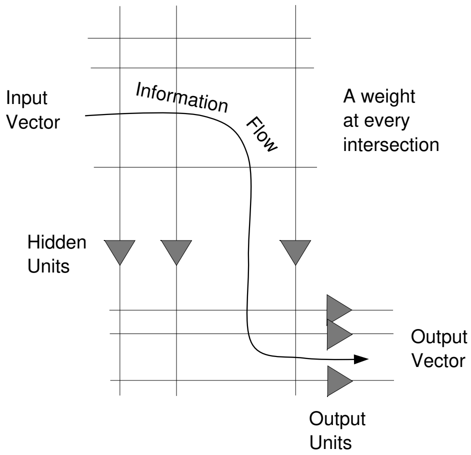

No Averaging
Averaging Over
4 Windows
Averaging Over
8 Windows
Output and Target
for Output Unit 1
Base Task
Output and Target
for Output Unit 2
Letter Task
Output and Target
for Output Unit 3
Math Task
1
0.5
0
Math
Letter
Base
Fraction
Correct = 0.79
Fraction
Correct = 0.85
Fraction
Correct = 0.96
1
35
70
105
1
32
64
96
1
28
56
84
Window Number
(0 to 10 seconds)
Window Number
(0 to 10 seconds)
Window Number
(0 to 10 seconds)

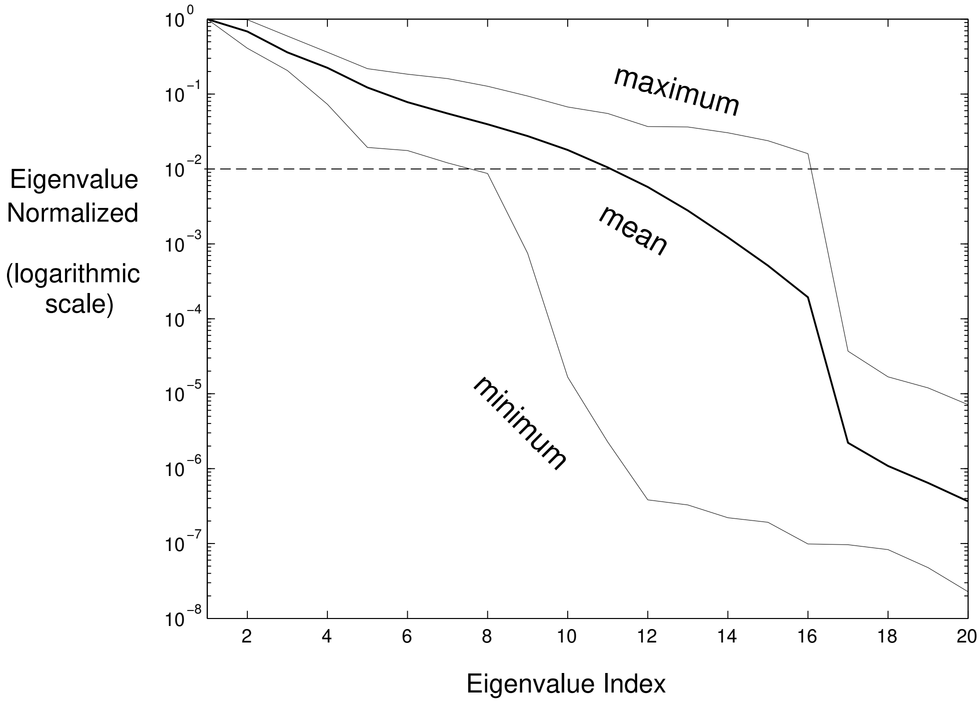


20-0 Network
10-0 Network
5-0 Network
Average Percent of Test Patterns Correctly Classified
90
88
86
84
82
80
78
76
74
72
70
1
10
20
30
38
Number of Windows Averaged Over

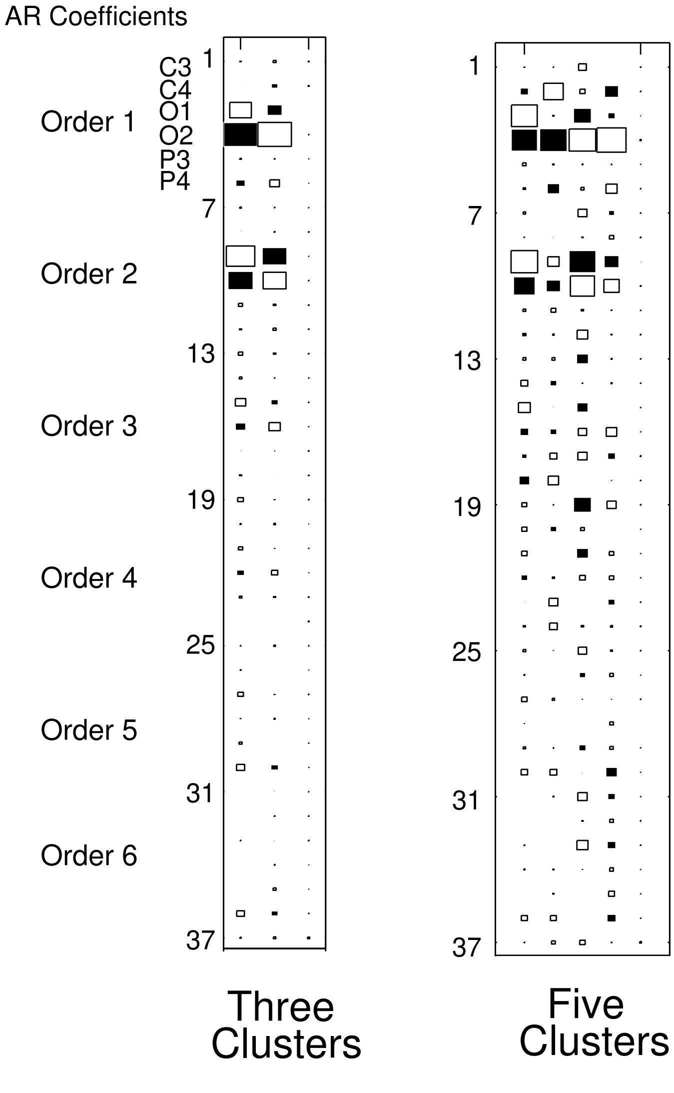

Network 1
Network 2
AR Coefficients
C3
C4
O1
O2
P3
P4
Order 1
Order 2
Order 3
Order 4
Order 5
Order 6
Bias
Input
7
13
19
25
31
37
5
10
15
20
Hidden Units
Baseline
Letter
Math
1
2
3
Hidden Units

Key:


Input
Vector
Information
Flow
A weight
at every
intersection
Hidden
Units
Output
Vector
Output
Units



maximum
mean
minimum
Eigenvalue Normalized
(logarithmic scale)
Eigenvalue Index
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
2
4
6
8
10
12
14
16
18
20


AR Coefficients
C3
C4
O1
O2
P3
P4
Order 1
Order 2
Order 3
Order 4
Order 5
Order 6
1
7
13
19
25
31
37
Three Clusters
Five Clusters