Nonlinear Modeling and Neural Network Analysis for EEG Brain-computer Interface

OURE Final Report

Devin Cornell

Dr. Donald Wunsch

Electrical Engineering

April 1, 2015

Abstract

This research explores some of the most recent advances in Electroencephalography-based Brain-computer interface algorithms, and provides some benchmark data with which to compare algorithms given the same data constraints. The three algorithms with focus in this research are filter bank common spatial patterns, wavelet-based feature extractors, and convolutional neural networks. Of these methods tested, only the FBCSP algorithm provides satisfactory results, but it is hypothesized that future work regarding the other two techniques could provide better results, either separately or in conjunction with the FBCSP algorithm.

Contents

Introduction

Background Information

Electroencephalography is the recording of electrical potential measured at the scalp relative to some reference point [1]. Changes in electrical potential on the scalp are due to sums of many small voltage changes across individual neurons in the brain, creating a link between mental tasks and the measured electrical potentials. Some applications of practical EEG use include seizure detection [2], Brain-computer interfaces (BCI) [3], and measurement of consciousness [4]. EEG has the features of high time-frequency resolution, low cost implementations, and low spatial resolution relative to other brain imaging techniques such as functional magnetic resonance imaging (fMRI), computed tomography (CT), or positron emission tomography (PET) [5].

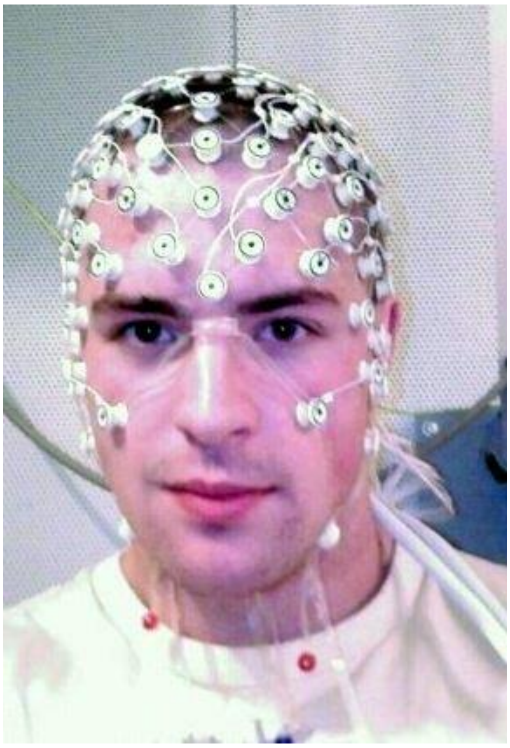

Figure 1: Figure thanks to Douglas Myers, public domain. This example has many EEG channels, however only a 3 channel system was used in this research.

The goal of BCI is to interpret these electrical potentials as mental tasks for the control of a system. Many practical applications have been developed in the last decade including prosthetic assistance [6], video gaming [7], art creation [8], and the exploration of virtual environments [9]. The most common mental task used for control purposes is called motor imagery [10]. Motor imagery is a process where a subject imagines the moving of a physical body part without actually moving it. This can be utilized by EEG because of the way that sensorimotor functionality is physically separated in the motor cortex of the brain.

Although it is possible to detect different types of motor imagery in the brain for the purpose of a control interface, it is a difficult task where some of the most advanced methods are leading to average classification rates around 80 % [11]. It is often a difficult task for subjects to maintain focus on motor imagery for sustained periods of time, and even the exact inter-subject technique for generating mental imagery differs enough to cause classification problems across subjects. Despite all of these challenges, motor imagery-based BCI has been demonstrated for use in practical applications and with subject-specific training data and subject experience this could be a powerful way for subjects to interact with the environment.

Labeled Data

The data set focused on in this research is the BCI IV 2b data set [12] originally published in [9] and used for the BCI IV competition. This competition was created for the sake of encouraging researchers to come up with new solutions to some of the most challenging problems in EEG. This set contains 5 sessions of 120 trials each for 9 subjects. This resulted in 600 trials for each test subject. At the time of the competition only 3 of the five sessions had known labels, but now the competition is over and labels for all 5 sessions have been provided [12]. The last two sessions will still be used for evaluation of classifier models due to the availability of comparative data from literature.

The data consists of three bipolar EEG channels from the scalp and three electrooculogram (EOG) channel measurements, all sampled at 250 Hz and band-passed between 05 Hz and 100 Hz [12]. The EOG signals are voltage levels near the eye that can detect the electrical field emitted from the eye. This electrical field can have a significant effect on the voltages measured on the scalp, and are recorded so that the effects of EOG may be characterized in the preprocessing phase [13]. Prior to each session, there are three EOG analysis sessions where the subjects were asked to physically adjust their eyes in different ways. In the preprocessing phase, this information is used to generate a linear model for how the electrical field affects the EEG voltage measurements. This linear model is used to cancel the effects of the EOG before feature extraction is performed [13]. During the first two sessions the patients did not receive feedback as to whether or not they had performed the motor imagery correctly, but in the last two sessions they did receive this information.

Training Data Extraction

In both cases, the period of time used for feature extraction and classification appeared 0.5 seconds after the cue began and was recorded for 2 seconds as per [11]. Two seconds for each of the three channels for each trial was collected as the training set for the feature extraction algorithms. This figure, from [11], shows the “train_time_segment” that was used for all feature extractors used in this research.

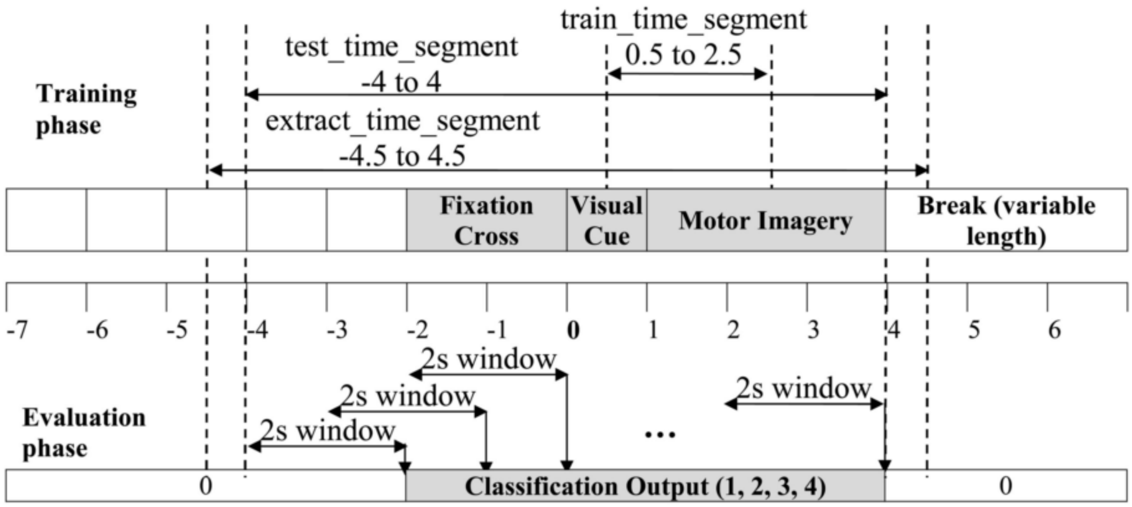

Figure 2: given in [11] to show the time segment extracted for feature generation. This extraction time was also used for the purpose of this research.

Evaluation Metric

The evaluation metric for the supervised learning classification rates is based on the kappa statistic found in [14]. The kappa statistic is useful because it takes into account the actual vs expected classification rate, where expected classification rate is based on the number of possible classes and a random classifier. For a two class system such as the one found in the BCI IV 2b dataset, the expected probability is 0.5 because there are two possible classes. For a two class system, a conversion table from kappa statistic to classification percentages has been provided.

Table 1: Kappa value to percentage classification lookup table for two class systems. This is just to give relative reference points for various kappa statistics.

Kappa	Percent
0.7	85%
0.5	75%
0.25	62.5%
0.1	55%

This equation for the kappa statistic was originally given by [14].

Equation 1: kappa statistic, given in [14].

$$K = \frac{P(A) - P(E)}{1 - P(E)}$$

Kappa statistics are provided for all classifier data instead of actual training percentages due to their ability to be less misleading [14]. A Kappa Statistic close to zero indicates that the classifier is no better than random chance at choosing the correct motor imagery class.

EEG BCI Analysis Model

There are several stages in the process of EEG BCI interpretation that can be considered when choosing a BCI system.

- performance of motor imagery by subject
- data collection apparatus
- preprocessing and EOG removal
- feature extraction
- classification of motor imagery

In this research, both feature extraction and classification techniques will be explored to determine effectiveness.

Research Tools

Due to the algorithmic complexity of BCI motor imagery classification, Matlab was selected to be the primary work environment for this research project. In addition, several other external libraries were utilized.

- Biosig Toolbox
- ConvNet Library Using NVidia CUDA

The creation and availability of these tools made this research possible.

BioSig Toolbox

The BioSig toolbox is an open source library for biomedical signal processing. It includes tools for processing several different signals including EEG by providing tools for artifact removal and updates to Matlab that allow for a simpler way of processing signals with NaN values. BioSig is important to this project because the BCI IV 2b dataset is stored in the GDF file format, a file format used by BioSig specifically for storing biological signals. Biosig is also used for EOG preprocessing of all feature extractors and contains several classifiers which were used for comparison results sporadically throughout the research project.

ConvNet Library Using NVidia CUDA

The ConvNet library was used for the implementation of the Convolutional Neural Network. This code was compiled by Sergey Demyanov to be a flexible tool for invariant backpropagation across convolutional neural networks (CNN). The tool is comprised of three different implementations of the algorithm: Matlab, C++ compiled CPU version, and C++ compiled GPU version using NVidia CUDA. The Matlab code was created by Rasmus Berg Palm from the DeepLearningToolbox [15], the C++ CPU version came from [16], and the CUDA version came from [16] and the implementation of cuda-convnet2.

While the CNN was originally utilizing the Matlab version for ease of debugging and verification, eventually it was important to switch to the GPU version because of an order of magnitude differences in speed. For a data set size and CNN architecture that would cause the Matlab version to take up to 12 hours, the GPU version would complete in less than 10 minutes. Because deep learning tools such as the CNN are becoming so popular today, it is important that more developers and researchers begin to utilize high-performance computing tools like CUDA, especially given the cost overhead compared to expensive computer clusters or supercomputers. Without a GPU implementation of the CNN, it may have been impossible to test so many different network configurations.

Methodology

There are a number of different ways that this data can be analyzed. In this case, classifiers were trained only on subject-specific data. While this does not provide optimal classification rates across all subjects, it does provide a good indicator of classifier performance for each patient and thus a good metric for comparing feature extraction and classification techniques for BCI.

This structure shows the general format of the data processing.

1. For each patient
    a. Extract cued motor imagery portions of signal
    b. Remove artifacts from EOG
    c. Generate features and train classifier across all training data
    d. Evaluate combined test data to determine correct percentages
    e. Generate Kappa value to indicate effectiveness of technique

Employed Algorithms

Four types of algorithms were explored in this research to understand the current popular techniques in BCI.

- Preprocessing and artifact removal
- Classification
- Feature extraction
- Combined feature extraction and classification

Preprocessing and Artifact Removal

Two techniques were used in an effort to remove artifacts and eliminate redundant information from the EEG channels.

EOG Artifact Removal

For EOG analysis, the EOG preprocessing technique provided by the BioSig toolbox was used. This technique uses the EOG channels to provide a linear transform that removes the effects of EOG in the EEG channels. This technique was used for all trials and the evaluation of all classifiers.

(EOG reference in dataset document)

Classification

After feature extraction is performed, the motor imagery type of a given time segment can be determined with a trained classifier. Two types of classifiers were explored in this research.

- Multi-Layer Perceptron (MLP)
- Support Vector Machine (SVM) with nonlinear kernels

Multi-Layer Perceptron

The multi-layer perceptron is the simplest and most common type of neural network, usually employed as part of a more complicated neural network such as the recurrent neural network or convolutional neural network. For applications in classification, it can create nonlinear classification boundaries using an iterative learning algorithm called backpropagation [17].

This figure, from [17], shows how first a set of inputs is propagated through each layer of a network, to finally output a class estimation, and then the wrong classes are selected to generate propagation of error backward through the network to the layer weights. Then, the layer weights can be changed to generate a more accurate estimation the next time the input is forward-propagated.

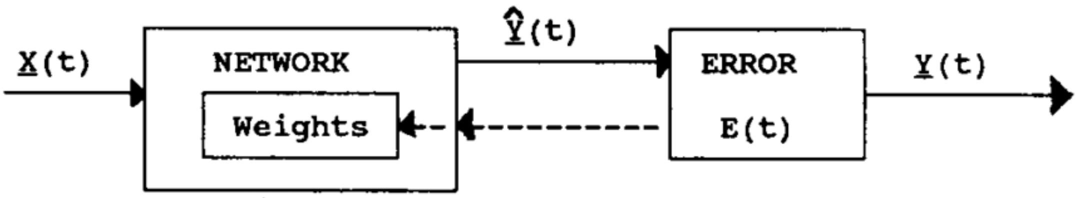

Figure 3: Given in [17] to show how backpropagation works.

For this project, a custom Multi-Layer perceptron was generated that can classify a set of arbitrary input features. This classifier was used for all data recorded in this research project due to the ability to handle nonlinear classification boundaries [17].

Feature Extraction

Before classification can occur, the raw EEG signals needs to be converted to a series of features which can allow a classifier to distinguish the motor imagery type. The following feature extraction algorithms were explored in this research.

- Common Spatial Patterns (CSP)
- Filter Bank Common Spatial Patterns (FBCSP)
- Autoregressive moving-average model (ARMA model)
- Wavelet Transforms

Common Spatial Patterns

Common Spatial Patterns is a statistical analysis technique that creates a spatial filter using differences in covariance matrices between two classes [18]. This technique is useful because it allows a BCI system to filter out parts of a signal based on location, or where on the scalp a signal component is originating. Although EEG is considered to have low spatial resolution, it can still be enough to determine the classification using these filters.

It has been shown by [18] that detection of the Event-Related Synchronization and Desynchronization is possible using spatial filters. This accomplished by creating a spatial filter shown here as W_b and transforming the EEG data $E_{b,i}$ to get a spatially filtered signal Z_b [11].

Equation 2: Spatial filter, given in [11]

$$\mathbf{Z}_{b,i} = \mathbf{W}_b^T \mathbf{E}_{b,i},$$

This W_b can be generated from the Eigen decomposition given in this equation also from [11], where $\sum_{b,1}$ and $\sum_{b,2}$ are the covariance matrices between channels for the two classes corresponding to the two types of mental imagery.

Equation 3: Eigen decomposition of class-specific covariance matrices, given in [11]

$$\Sigma_{b,1}\mathbf{W}_b = \left(\Sigma_{b,1} + \Sigma_{b,2}\right)\mathbf{W}_b\mathbf{D}_b,$$

Note that this Eigen decomposition is essentially the decomposition of the ratio of the covariance matrix of each class, which leads the CSP algorithm to find a transformation that maximizes the spatial differences between the two classes. Then [11] describes that feature generation can be given by the following equation, where $v_{b,i}$ is the feature vector and W_b is the CSP matrix generated from solving the previous equation.

Equation 4: Feature generation from CSP matrices, given in [11].

$$\mathbf{v}_{b,i} = \log \frac{\operatorname{diag}\left(\bar{\mathbf{W}}_b^T \mathbf{E}_{b,i} \mathbf{E}_{b,i}^T \bar{\mathbf{W}}_b\right)}{\operatorname{tr}\left[\bar{\mathbf{W}}_b^T \mathbf{E}_{b,i} \mathbf{E}_{b,i}^T \bar{\mathbf{W}}_b\right]},$$

This feature generation is performed by combining all of the trials from each mental imagery class and then generating the transformation that maximizes 2^{nd}-order differences between channels. This feature generation gives the relative likelihood of $E_{b,i}$ of belonging to a given class based on this statistic. [11]

Note that covariance is a 2^{nd}-order statistic, and therefore provides a comparison of the signals based on the assumption that they are Gaussian in nature. In future work, it may be of use to generate spatial filters based on higher order statistics as well as the 2^{nd}-order statistics. This concept will be extended in the next session by first separating signals based on frequency.

Due to the increased effectiveness of splitting into frequency bands before generating the CSP matrix, only the FBCSP algorithm was implemented for this experiment. [11]

Filter Bank Common Spatial Patterns

Filter Bank Common Spatial Patterns (FBCSP) combines the spatial filter of CSP with selected band pass frequency filters in order to determine origins of particular frequencies across the scalp. This works by creating a separate spatial filter for each frequency band in question, and then generating features for each band.

The subscript b used in each of the figures from the section on CSP are can be assumed to indicate that a CSP matrix is generated for each frequency band and then features are also generated for each frequency band. There is a large inter-patient variance in the specific frequencies which are relevant, and so the classification of these features have been shown to be more effective when combined with the Mutual information-based best individual feature algorithm (MIBIF) or Mutual information-based rough set theory algorithm (MIRSR) algorithms given in [11], or the extension given in [19].

This figure from [11] is helpful in understanding the operation of the FBCSP algorithm.

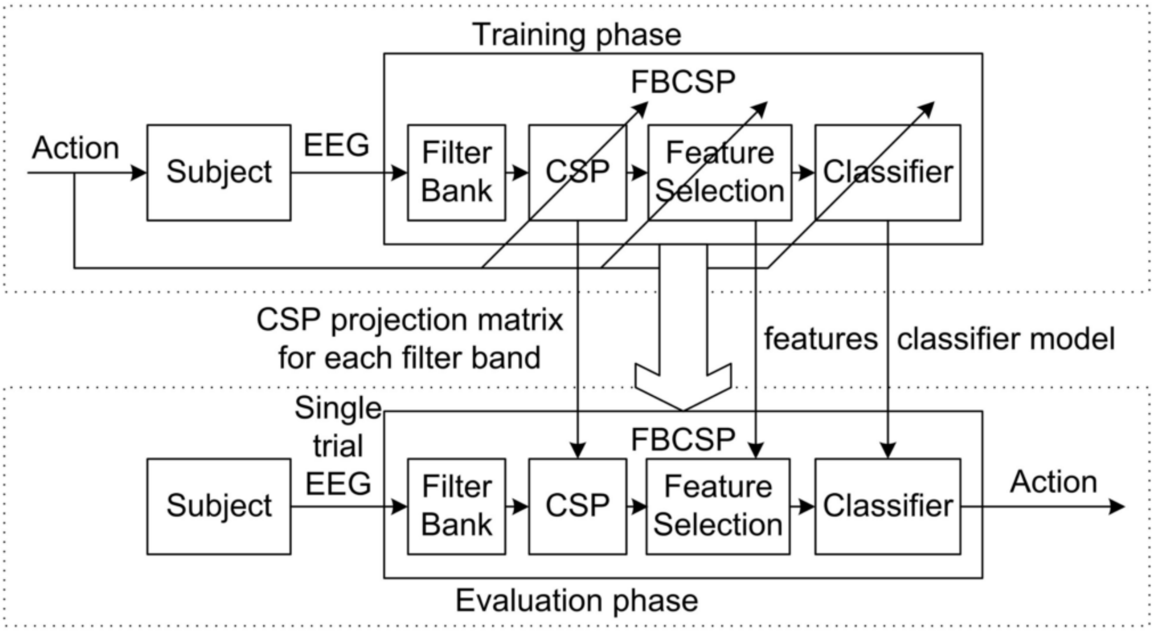

Figure 4: FBSCP diagram given in [11].

For experimentation purposes, each FBCSP was created based only on the training data from the given patient. While testing Kappa values could certainly be increased by choosing selected training sessions as done in [11], this more accurately shows the ability of the FBCSP algorithm to generate classifier models from individual subject data. Note that a multi-layer perceptron (MLP) classifier with 50 backpropagation iterations was used here to generate kappa statistics.

The data collected here was based on the same spectral filters as found in [11], which is from 4 to 40 Hz in bands of 4 Hz.

Table 2: Classification rates recorded in FBCSP example.

'Subject'	'Training Time (sec)'	'Classification Time (sec)'	'Train K'	'Test K'
'B0101T'	1.39	0.30	0.38	0.08
'B0201T'	1.05	0.29	0.27	0.04
'B0301T'	1.00	0.29	0.32	0.12
'B0401T'	1.10	0.30	0.90	0.46
'B0501T'	1.11	0.30	0.72	0.34
'B0601T'	1.09	0.30	0.54	0.36
'B0701T'	1.04	0.29	0.50	0.24
'B0801T'	1.07	0.32	0.37	0.41

'B0901T'	1.03	0.29	0.49	0.23
'Averages'	**1.10**	**0.30**	**0.50**	**0.25**

Note from the kappa statistic table that the kappa value of 0.25 corresponds to about 62.5 % classifications. Although this is a very low percentage, also note that it is performed without and feature selection algorithms as mentioned above, and has only been trained on the same patient's training data, which sometimes becomes as low as 0.27 in the case of 'B0201T'.

Also note the comparison with data from [11] which uses training data from only selected trials in addition to the MIRSR algorithm mentioned above. This data was obtained from Table 4 of [11], also using the BCI IV 2b dataset.

Table 3: Classification Kappa values recorded using the MIRSR algorithm on BCI IV 2b data.

Subject	Kappa
'B0101T'	0.400
'B0201T'	0.207
'B0301T'	0.219
'B0401T'	0.950
'B0501T'	0.856
'B0601T'	0.613
'B0701T'	0.550
'B0801T'	0.850
'B0901T'	0.744
'Averages'	**0.599**

Wavelet Transform with Channel Covariance

The wavelet transform covariance feature extraction technique was a custom algorithm that generates features based on differences in amplitude and phase information given by convolving the EEG signal with complex wavelet functions. While the performance of this algorithm is fairly low, it was attempted primarily as a precursor to use of a wavelet neural network, which may be explored in the future as an adaptive feature extraction technique.

The selected type of wavelet used for this classifier was the Morlet Wavelet, the most basic of analytic wavelets that can give information about the amplitude and phase of a specific frequency for a given period of time. Note that this particular technique was not explored fully, and there may be much room for improvement by extending the wavelet transform to two dimensions, trying different wavelets, or creating an adaptive system where the wavelets can be trained by a classifier through use of a wavelet neural network.

This is an example of the real part of a 2 dimensional Morlet wavelet as given in [20].

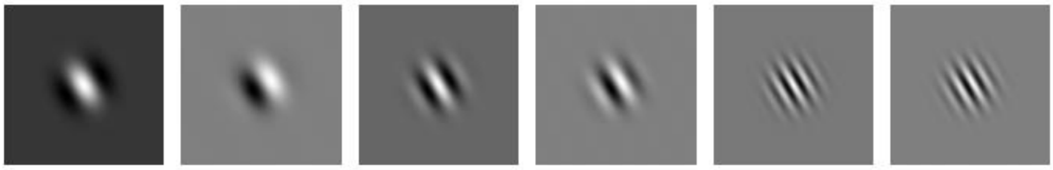

Figure 5: Real part of Morlet wavelet function in two dimensions provided from [20].

Features extracted are elements of the amplitude and phase covariance matrices between channels of a training sample that has been convolved channel-wise with a one dimensional complex Morlet wavelet at frequencies from 4 to 40 Hz in 2 Hz increments. When it is combined with a multi-layer perceptron classifier, the following classification results were obtained.

Table 4: Data collected using an MLP with the wavelet transform channel covariance features.

'Subject'	'Training Time (sec)'	'Classification Time (sec)'	'Train K'	'Test K'
'B0101T'	18.007	15.869	0.395	0.031
'B0201T'	17.373	16.058	0.225	0.050
'B0301T'	17.868	15.845	0.190	-0.050
'B0401T'	18.560	16.686	0.695	0.281
'B0501T'	18.669	16.796	0.286	-0.019
'B0601T'	18.283	15.972	0.195	-0.031
'B0701T'	17.616	15.560	0.430	0.194
'B0801T'	19.345	17.640	0.341	0.338
'B0901T'	17.362	15.891	0.365	0.063
'Averages'	**18.120**	**16.257**	**0.347**	**0.095**

Note that while most evaluation kappa values are insignificant and indicate no predictability, two of the patients, B0401T and B0801T both have significant predictive abilities. It is hypothesized that further experimentation of wavelet types and frequencies could lead to higher classifications for all patients if the correct training sessions are selected. As-is it could also be used as a potential boosting classifier to get increased classification rates when combined with other classifiers or feature extractors.

Combined Feature Extraction and Classification

While both feature extraction and classification models are needed in order to interpret user intentions in a BCI system, there are some techniques which combine the two for adaptive feature extraction.

Convolutional Neural Network

Show the differences in training times, classification times, and testing percentages that are seen by adjusting different parameters of the network.

Table 5: Data collected from a CNN composed of 16 3x3 convolution layers on the input, 32 2x2 convolution layers, and then a 256 perceptron layer.

'Subject'	'Training Time (sec)'	'Classification Time (sec)'	'Train K'	'Test K'
'B0101T'	90.519	0.411	0.550	0.069
'B0201T'	89.333	0.397	0.685	-0.007
'B0301T'	90.567	0.404	0.765	0.044
'B0401T'	91.748	0.409	0.786	0.163
'B0501T'	91.985	0.407	0.705	0.156
'B0601T'	89.936	0.406	0.855	0.081
'B0701T'	90.171	0.401	0.680	0.050
'B0801T'	91.939	0.412	0.764	0.238
'B0901T'	90.685	0.423	0.610	0.056
'Averages'	**90.765**	**0.408**	**0.711**	**0.094**

Conclusions

Based on the recorded events for the three experimental classifiers, only the known FBCSP algorithm obtains reasonable classification rates. Due to the complex dynamic nature of EEG data, it is very difficult to score highly with the evaluation data. Even some of the highest performing classifiers such as [11] and [19] are not performing over 90%. Based on the demonstration of some of the raw feature extraction techniques in a very constrained learning environment combined with the current state-of-the-art, it is determined that there is still much progress to be made in the way of EEG BCI analysis algorithms.

While the experiment performed here using the FBCSP algorithm did not perform as well as FBCSP combined with MIRSR, it has been shown that even trained only on the current patient, for most patients it is possible to obtain decent classification rates. It may be possible in the future to use genetic algorithms to select different sessions for inclusion in the learning set that could train a classifier.

Even though the wavelet classifier was not fully adaptive, nor did it explore different types of wavelets, it still performed better than the CNN. It may be possible to obtain higher results by combining analytic wavelet transforms with CSP instead of band-pass filters to create a wavelet-based Common Spatial Patterns.

While convolutional neural networks have become very popular over the last five years, they are not suitable for this application using the raw EEG data. It is hypothesized that by converting the raw EEG signals into a higher dimensional space using Fourier Transform or other types of transform before using the CNN, it may be possible to obtain higher classification rates. This has been demonstrated by use of the Fourier transform by [21], but there are other transforms to try as well.

Acknowledgements

Special thanks to the facilitators of the BCI competition, all of the providers of the software libraries given above, and all of the people who have been doing work on EEG BCI.

References

1. Niedermeyer E. and da Silva F.L. (2004). Electroencephalography: Basic Principles, Clinical Applications, and Related Fields. Lippincot Williams & Wilkins. ISBN 0-7817-5126-8.
2. Omerhodzic, I., Avdakovic, S., Nuhanovic, a, Dizdarevic, K., & Transform, a W. (2010). Energy Distribution of EEG Signals : EEG Signal Wavelet-Neural Network Classifier. Neurocomputing, 2, 210–215.
3. Coyle, D. (2009). Neural network based auto association and time-series prediction for biosignal processing in brain-computer interfaces. IEEE Computational Intelligence Magazine, 4(4), 47-59.
4. Höhne, J., Holz, E., Staiger-Sälzer, P., Müller, K.-R., Kübler, A., & Tangermann, M. (2014). Motor imagery for severely motor-impaired patients: evidence for brain-computer interfacing as superior control solution. PloS One, 9(8), e104854. doi:10.1371/journal.pone.0104854
5. McFarland, D. J., McCane, L. M., David, S. V., & Wolpaw, J. R. (1997). Spatial filter selection for EEG-based communication. Electroencephalography and Clinical Neurophysiology, 103(3), 386–394. doi:10.1016/S0013-4694(97)00022-2
6. Tangermann, M., Müller, K. R., Aertsen, A., Birbaumer, N., Braun, C., Brunner, C., … Blankertz, B. (2012). Review of the BCI competition IV. Frontiers in Neuroscience, 6(JULY), 1–31. doi:10.3389/fnins.2012.00055
7. Mohamed, E. (2014). Enhancing EEG Signals in Brain Computer Interface Using Wavelet Transform. International Journal of Information and Electronics Engineering, 4(3), 234–238. doi:10.7763/IJIEE.2014.V4.440
8. Muldner, K., & Burleson, W. (2013). Utilizing sensor data to model students’ creativity in a digital environment. Computers in Human Behavior, 42, 127–137. doi:10.1016/j.chb.2013.10.060
9. Leeb, R., Lee, F., Keinrath, C., Scherer, R., Bischof, H., & Pfurtscheller, G. (2007). Brain-computer communication: Motivation, aim, and impact of exploring a virtual apartment. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 15(4), 473–482. doi:10.1109/TNSRE.2007.906956
10. Tangermann, M., Müller, K. R., Aertsen, A., Birbaumer, N., Braun, C., Brunner, C., … Blankertz, B. (2012). Review of the BCI competition IV. Frontiers in Neuroscience, 6(JULY), 1–31. doi:10.3389/fnins.2012.00055
11. Ang, K. K., Chin, Z. Y., Wang, C., Guan, C., & Zhang, H. (2012). Filter Bank Common Spatial Pattern Algorithm on BCI Competition IV Datasets 2a and 2b. Frontiers in Neuroscience, 6(March), 39. doi:10.3389/fnins.2012.00039
12. Brunner, C., & Leeb, R. (2008). BCI Competition 2008 – Graz data set A. Knowledge Creation Diffusion Utilization, 1–6. doi:10.1109/TBME.2004.827081
13. Fatourechi, M., Bashashati, A., Ward, R. K., & Birch, G. E. (2007). EMG and EOG artifacts in brain computer interface systems: A survey. Clinical Neurophysiology, 118(3), 480-494. Retrieved from www.scopus.com

14. Carletta, J. (1996). Assessing agreement on classification tasks: the kappa statistic, (2), 9. doi:10.1.1.48.4108
15. Palm, Rasmus Berg. "Rasmusbergpalm/DeepLearnToolbox." Rasmusbergpalm/DeepLearnToolbox. GitHub, 1 Apr. 2015. Web. 1 Apr. 2015. <https://github.com/rasmusbergpalm/DeepLearnToolbox>.
16. Krizhevsky, Alex. "Cuda-convnet2." Cuda-convnet2. Google Code, 1 Apr. 2015. Web. 1 Apr. 2015. <https://code.google.com/p/cuda-convnet2/>.
17. Werbos, P. J. (1990). Backpropagation Through Time : What It Does and How to Do It, 78(October), 1550–1560.
18. McFarland, D. J., McCane, L. M., David, S. V., & Wolpaw, J. R. (1997). Spatial filter selection for EEG-based communication. Electroencephalography and Clinical Neurophysiology, 103(3), 386–394. doi:10.1016/S0013-4694(97)00022-2
19. Meng, J., Yao, L., Sheng, X., Zhang, D., & Zhu, X. (2015). Simultaneously optimizing spatial spectral features based on mutual information for EEG classification. IEEE Transactions on Biomedical Engineering, 62(1), 227-240.
20. Bernardino, a, & Santos-Victor, J. (2005). A real-time Gabor primal sketch for visual attention. Pattern Recognition and Image Analysis, Pt 1, Proceedings, 3522, 335–342.
21. Cecotti, H., & Graeser, A. (2008). Convolutional Neural Network with embedded Fourier Transform for EEG classification. 2008 19th International Conference on Pattern Recognition, 1–4. doi:10.1109/ICPR.2008.4761638