

Classifying the Brain's Motor Activity via Deep Learning

Tania Morimoto & Sean Sketch

Motivation

Over 50 million Americans suffer from mobility or dexterity impairments. Over the past few decades, research in engineering and neuroscience has resulted in brain-computer interfaces (BCIs) that show promise to return independence to this movement-impaired population. Generally speaking, BCIs aim to determine their user's intention and convert this intention to a control signal for some external device. Although it may be possible to “decode” the activity from any region of the brain, most research has focused on that produced by the motor cortex. The control signals developed from decoded motor activity have been used to move computer cursors and drive robotic arms. Algorithms to more quickly and accurately decode motor activity are an expanding area of research.

Background

BCIs generally record neural activity in one of three ways: (1) intracortically, via an implanted electrode array, (2) intracranially, via electrocorticography (ECoG), or (3) from the surface of the scalp, via electroencephalography (EEG). Unsurprisingly, invasive BCIs - intracortical and ECoG - are rare. There are probably fewer than ten patients in the United States with implanted arrays. EEG-based BCIs, on the other hand, can be used without an invasive procedure or doctor's supervision. However, accessibility comes at a cost. Compared to recordings from intracortical or ECoG arrays, EEG signals have low spatial resolution and are easily contaminated by non-neural signals, such as movements of the face and head. These characteristics make it difficult for conventional decoding algorithms to reliably determine user intent [1].

For motor activity, these conventional algorithms are based on the brain's mu and beta rhythms, electrical oscillations between 8 and 36 Hz that arise from large populations of neurons in the primary motor cortex. The nature of these rhythms can be monitored by EEG, and there is evidence that both motor movements (e.g., opening and closing the hand) and motor imagery (e.g., imagining opening and closing the hand) affect their amplitude. As seen in Fig. 1, this amplitude modulation (relative to a resting state) is obvious when raw EEG signals are converted from the time domain into the frequency domain. It occurs on different EEG channels (recording locations - see Fig. 2) at different frequencies for different types of motor imagery (e.g., left versus right hand movement). By examining plots similar to the one in Fig. 1, it is possible to manually select the most relevant channel-frequency pairs for each type of motor imagery. These become the “features” for the classification algorithm. BCI2000, an open-source platform for BCI research, uses such features [2]; it will serve as the standard of comparison in the Results below.

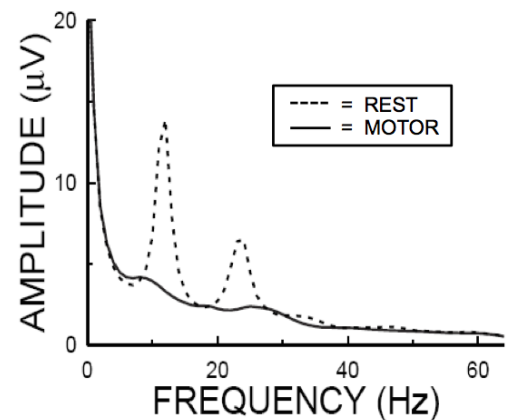


Figure 1 – EEG amplitude is modulated between 8 and 36 Hz (mu/beta band) during motor tasks [2]

Although this conventional approach to feature selection and classification for motor EEG is ubiquitous, EEG's low signal to noise ratio makes it such that BCI users must be extensively trained before their features are clear enough to manually extract. In this report, we propose an alternative method for feature extraction from EEG. Just as deep networks were able to learn phonemes from speech data [3], such networks could extract basic “neural activity units” as features from EEG recordings. Such “neural activity units” could serve as features for enhanced BCI classification as well as improve our understanding of the brain's processing.

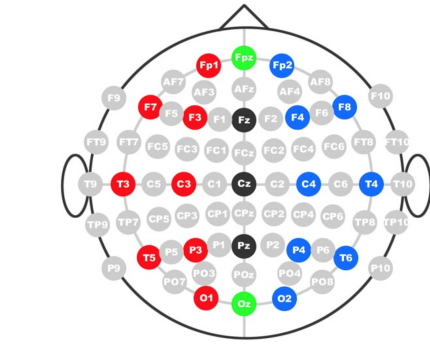


Figure 3 displays a schematic of the autoencoder architecture and EEG preprocessing. As noted in the schematic, the resulting features are the inputs to the network that maximally activate each of the hidden neurons. In other words, they are the characteristics of EEG signals that each neuron is “tuned” to detect. The feature for a given hidden neuron is visualized by normalizing a vector of the weights that connect that neuron to each element of the input.

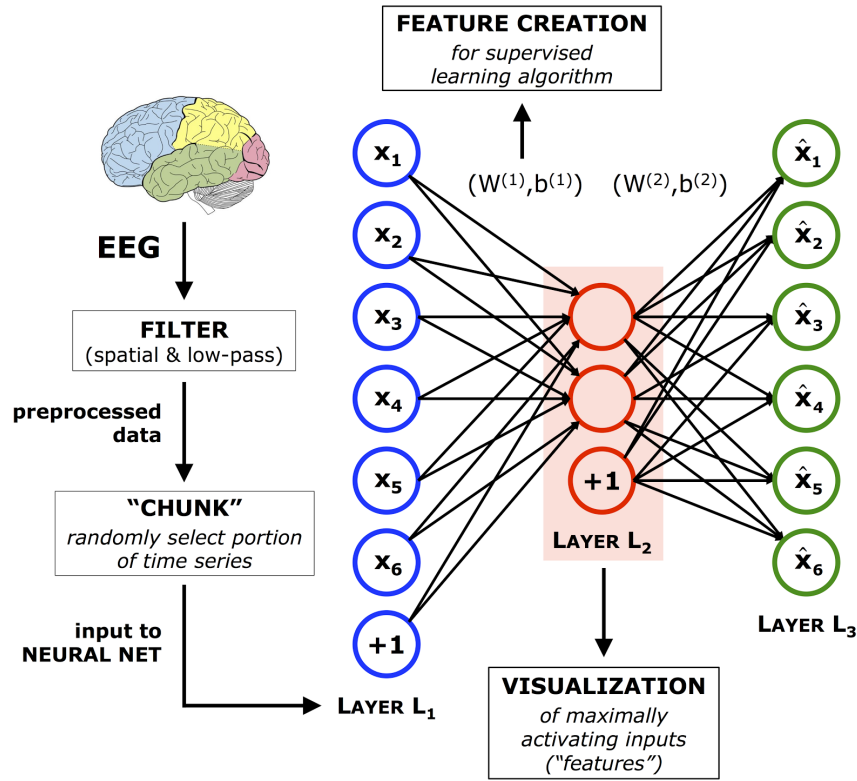


Figure 3 – Method for preprocessing EEG data and extracting features using a single hidden-layer autoencoder neural net

Classification Learning Algorithms: We implemented two supervised learning models - binary logistic regression (BLR) and a support vector machine (SVM) - due to their success in EEG classification throughout the neural engineering literature. Training and testing data were first filtered and “chunked” using the same preprocessing steps described above. Then, in a forward pass through the neural network, the weights and biases learned by the autoencoder transformed the preprocessed time series into the compressed feature space (the output of the network’s hidden layer L_2). These L_2 outputs became the inputs for BLR and the SVM. The label associated with each input was 0 (left) or 1 (right) for BLR and -1 (left) or +1 (right) for the SVM.

Results

Feature Extraction: After trial-and-error optimization over the autoencoder’s number of hidden neurons, “chunk” time, and regularization weight, we implemented an autoencoder network with 8 hidden neurons, a “chunk” time of 0.9 seconds, and a regularization weight of 0.1. This network was trained on data from channels above the primary motor cortex (FC1, FC3, FC5 on the right and FC2, FC4, FC6 on the left), and we visualized the learned feature for each hidden neuron, split up channel by channel. As displayed in Fig. 4, the features for each channel converged to waveforms with increasing iterations of gradient descent (i.e., longer training of the network).

In addition, electrodes recording from opposite hemispheres - for example, FC3 on the left and FC4 on the right - produced features with opposite phase. This was true regardless of the order in which the channel data was input to the network. This indicates that the autoencoder is extracting basic physiological information from convoluted EEG signals.

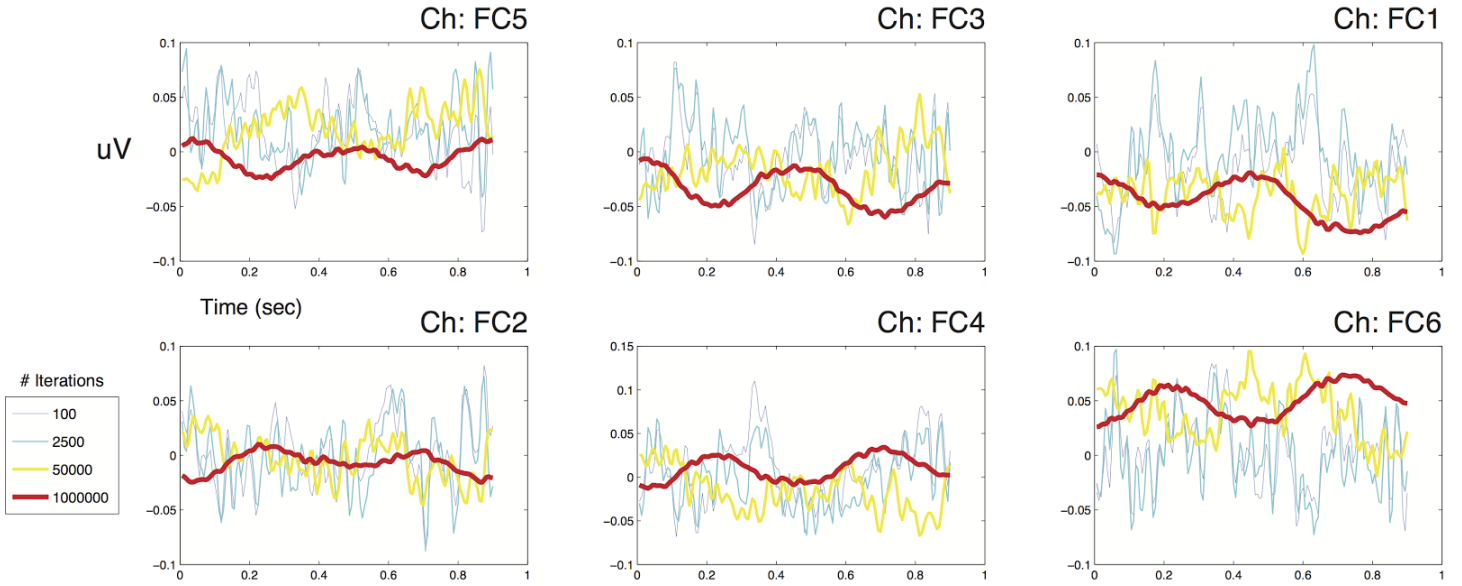


Figure 4 – Features converge to waveforms with more training of the autoencoder network

* This set of channel-wise features was derived from a single hidden neuron.

Classification: Both binary logistic regression and an L2-regularized, L2-loss SVM were implemented without substantial parameter tuning. Fig. 5 shows the training and testing error when the SVM was implemented using data from the six motor electrodes (FC1, FC3, FC5, FC2, FC4, FC6). As expected, the training error increased while the testing error decreased as the number of training examples became larger. Other parameters, including the “chunk” time of the input data and the number of iterations performed during feature selection, remained constant.

Table 2 compares the testing error of BLR, SVM, and BCI2000 classification algorithms. The inputs to BLR and the SVM were outputs from the autoencoder’s compressed feature space, as explained in the Methods above. The inputs to BCI2000 were raw EEG signals, which were classified using the conventional algorithms explained in the Background section. Although manual feature selection and BCI2000 classification outperforms the autoencoder’s features and BLR/SVM, there are benefits to our method. Most notably, the conventional method is limited by the nature of the brain’s mu and beta rhythms. There is likely additional information in EEG signals not captured by such a narrow analysis. Therefore, the performance of our method should increase with the addition of more electrodes (true of neural networks and deep learning in general), whereas conventional classification should remain the same, due to its dependence on recording over the motor cortex. In fact, this improvement is evident when comparing errors in Table 2 and Fig. 5, trained on two and six electrodes respectively. Finally, there is room for substantial optimization in selecting parameters for both the autoencoder and supervised learning algorithms.

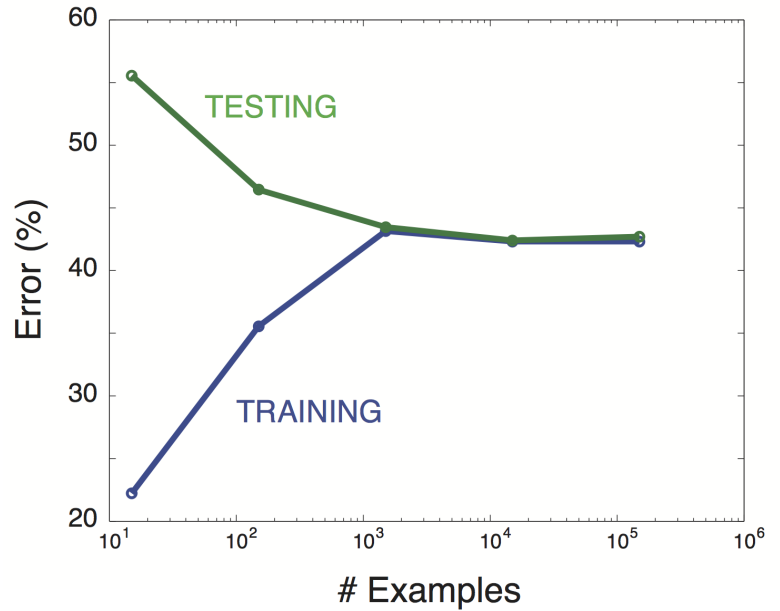


Figure 5 – Training error increases and testing error decreases as more examples are used to train the SVM

<i>Classification Method</i>	<i>Number of Training Examples</i>	<i>Training Error</i>	<i>Number of Testing Examples</i>	<i>Testing Error</i>
Binary Logistic Regression	2000	48.0%	2000	51.6%
SVM		48.3%		51.2%
BCI2000	N/A	N/A	N/A <i>* classifying in real time with 0.5s window</i>	18.0%

Table 2 – Comparison of classification error

Future Work

Future work on the project can be organized into three categories: (1) optimization, (2) application, and (3) extension. Optimization pertains to both autoencoder feature learning and the supervised classification algorithms. Although we performed a crude optimization for several autoencoder parameters (using nested “for” loops and reasonable parameter ranges), there are significantly more efficient methods, potentially using cross-validation. Specifically, we are interested in finding the optimal number of hidden neurons (aka. features), regularization weight, and time duration of the signals used for feature selection.

As noted above, the EEG BCI setup in the CHARM Lab allows for real-time cursor-movement experiments. Currently, however, it can only decode the EEG using manually selected features and BCI2000’s classification algorithms. This online control task will be useful in verifying the performance of our algorithms and deep-learned features. Given that the user can modulate his or her brain activity in reaction to the cursor’s movement, our algorithms and features will likely perform better than shown in the figure and table above. In addition to this application, using our method to learn features from intracortical or ECoG recordings might reveal more fundamental truths about the brain’s processing.

Finally, the features learned by our autoencoder network can be extended to other classification tasks. For example, if the same features were able to identify the current user of a system (from a known set of users), there would be less need for recalibration, which currently limits the practicality of BCIs.

Acknowledgements

This work was supported by Stanford University and the Collaborative Haptics and Robotics in Medicine (CHARM) Lab. The authors wish to thank Professor Andrew Ng and the course assistants for *CS 229: Machine Learning* for their technical support, as well as Jim Notwell for providing explanations and resources relevant to deep learning.

References

1. C. Guger, G. Edlinger, W. Harkam, I. Niedermayer, and G. Pfurtscheller, “How many people are able to operate an EEG-based brain-computer interface (BCI)?,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 11, no. 2, pp. 145–147, Jun. 2003.
2. G. Schalk, J. Mellinger: “A Practical Guide to Brain-Computer Interfacing with BCI2000”, Springer, 2010.
3. H. Lee, P. Pham, Y. Largman, and A. Ng, “Unsupervised feature learning for audio classification using convolutional deep belief networks,” in *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, Eds. Cambridge, MA: MIT Press, 2009, pp. 1096–1104.
4. G. Schalk, D. J. McFarland, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw, “BCI2000: a general-purpose brain-computer interface (BCI) system,” *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, pp. 1034–1043, Jun. 2004.
5. A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C. K. Peng, and H. E. Stanley, “PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals,” *Circulation*, vol. 101, no. 23, pp. E215–220, Jun. 2000.
6. Trans Cranial Technologies, *10/20 System Positioning*, Trans Cranial Technologies, 2012.