EEG Classification for Motor Imagery Tasks

ECE C247 Final Project

Alon Krauthammer alonk@ucla.edu
Mark Kubiak markkubiak@ucla.edu
Chris Munoz cmunozcortes@ucla.edu
Eashan Samak esamak@ucla.edu

Abstract

This project aims to perform electroencephalography (EEG) classification on the BCI competition IV motor imagery dataset 2a. We explored three different neural network architectures to accomplish this task: a shallow convolutional neural network (CNN), a spatial CNN (SCNN) and an SCNN with an LSTM layer with attention. In addition to exploring different models, we explored various pre-processing and data augmentation techniques: input standardization by mean centering, cropping, dimensionality reduction via PCA, and μ and β band filtering. In terms of accuracy, the worst performing model was the shallow CNN (our "default") network, followed by the SCNN. Our SCNN+LSTM with attention mechanism reached the best test accuracy with 76%, and our shallow CNN came in second with a test accuracy of 73%. For the latter model, dimensionality reduction with PCA improved performance by roughly 3-4%, when the dimensions were reduced to 16-18 channels.

1. Introduction

Decoding of raw EEG signals for motor imagery tasks can be realized through a variety of methods. These methods can be classified into five main categories: 1. linear classifiers (e.g. support vector machine, SVM), 2. nonlinear Bayesian classifiers (e.g. hidden Markov models HMM), 3. nearest neighbor classifiers (e.g. k-nearest neighbors, KNN), 4. neural networks, (e.g. multi-layer perception, MLP), and 5. combination of different methods using boosting or voting. [4]

Generally, these methods require pre-processing of the data to overcome low signal-to-noise ratios (SNR) typical of these recordings [2]. A variety of pre-processing techniques exist to extract useful features from the data. Popular techniques include cropping, trial averaging, normalization, band-pass filtering, and spatial transforms such as common spatial patterns (CSP). One shortcoming of this approach is that the expert knowledge these techniques require can introduce bias and may reinforce underlying assumptions about the data. In contrast, neural networks have the ability to learn representations of the raw data without any pre-processing steps and can consider potentially unknown correlations in the data [2].

In this project, we implemented several neural network architectures to examine their performance on EEG classification, and attempted several pre-processing techniques to verify whether performance could be improved if the models were fed handcrafted features as opposed to raw EEG data. Our ultimate goal was to analyze the performance of three different models on the task of subject independent classification (i.e. across all subjects on the dataset), and to analyze the performance of at least one model on the task of subject-dependent classification.

The first model implemented was a shallow convolutional neural network based on the work by Schirrmeister *et al*. in [3]. There were three main reasons for implementing this model: first, it provided a strong performance baseline for comparison with subsequent models without requiring any manual feature engineering; and second, it could be augmented with other kinds of layers to enable us to explore more complex architectures. The final reason was that it served as a baseline for quantifying the effects of any pre-processing steps, which proved to be very significant. The only modification to this architecture, the inclusion of dropout with a value of $p = 0.8$ prior to the dense layer at the output (see additional *Model* section for architecture details) had the greatest impact in reducing overfitting.

The third model we implemented was a spatial convolutional neural network (SCNN) designed by Kostas *et al*. [2]. This architecture includes multiple layers of spatial convolutions and temporal convolutions. These two types of layers are implemented separately so that spatial features are extracted independently from temporal features. The idea behind this is mimicking a typical feature-based pipeline consisting of spatial channel mixing and band-pass filtering.

The last model we examined was an augmented version of the SCNN described above that added an LSTM layer and attention mechanism after the spatial and temporal convolutional layers. The goal of the LSTM layer with atten-

tion mechanism is to enable the LSTM to develop a sequential processing capability with flexibility to consider any combination of samples that may be appropriate, rather than processing the sequence temporally in a sample by sample fashion [2].

2. Results

All models were trained using the "Four class motor imagery (001-2014)" dataset from the BCI Competition IV (dataset 2a) [1]. The data contains 22 channels corresponding to twenty two electrodes that were placed on the surface of the subjects' scalp. The signals were sampled at 250 Hz, band-pass filtered between 0.5 Hz and 100 Hz, and notch filtered at 50 Hz for line noise suppression. Two sessions were recorded on different days for each subject. Each session consisted of 6 runs of 48 trials (12 for each one of the four classes), giving a total of 288 trials per subject on each session. Specifically, the dataset provided for this project included data for 2558 trials which we assume corresponds to the data from one of the two sessions minus trials with bad data.

In the first part of our analysis, we evaluated the performance of each model on the task of classification across all subjects. To this end, we split the training dataset with 80% used for training and the remaining 20% for validation.

The best test accuracies for our models were as follows: 1. shallow CNN: 73.1%; 2. SCNN: 68.8%; and 3. SCNN+LSTM: 76.5%. Please refers to the *Methods* section attached to the end of this report for the validation accuracy of each model.

Considering that EEG data is in general subject dependent [4], these results are very reasonable and right in line with the classification accuracy reported in [2] and [3] when attempting to classify the data across all subjects on this dataset. Similarly, our results using the shallow CNN to perform subject specific classification are in line with the performance reported by other neural network models on this dataset [4].

3. Discussion

3.1. Pre-processing and Data Augmentation

We attempted several pre-processing steps and data augmentation techniques to try to improve the baseline performance of our shallow CNN model (60%). These steps included input data standardization, cropping in time, dimensionality reduction, and band-pass filtering in the μ and β bands.

3.1.1 Input Standardization

The first notable result was that standardization of the inputs seemed to have a regularizing effect on the given shallow CNN model, which was initially overfitting quite badly. The standardization that was attempted was as follows: for each example, calculate the mean and variance for each of the 22 channels. Then, mean-center each channel and scale to unit variance. This was performed as a preprocessing step rather than a regularization step, so these adjustments were explicitly calculated for each of the training, validation, and test examples. This had a clear regularizing effect, reducing overfitting and allowing for meaningful learning to occur for an additional 10 training epochs, improving performance by roughly 2-3%.

3.1.2 Cropping

The next pre-processing step that was found to be helpful was cropping the examples in time. With the full 1000 samples representing 4 seconds of data, we tried using only the first 3 seconds, as well as the last 3 seconds. It was clear from the performance of these two cropped models that the first second of data was very important, and indeed the final 1-2 seconds of each recording were having a slight negative impact on model performance. The best performance came from using only the first two seconds, or 500 samples, of each recording. This also reduced the size of the fully-connected layers, which may have further reduced overfitting and thereby been responsible for the marginally increased performance.

3.1.3 Dimensionality reduction

Additionally, we sought to pre-process the data with a PCA, reducing the number of channels and distilling the useful information for better model performance. An iterative search was performed on the standardized, cropped dataset, reducing the number of channels from 22 to n, where n was varied between 6 and 21. This was performed in a nested fashion with a search over values of p for dropout between 0.7 and 0.9. Models were trained for 150 epochs. It was found that PCA modestly improved model performance in general when reducing to between 12 and 20 channels.

3.1.4 μ and β Band Filtering

BCI research has shown that Event-Related Synchronization (ERS) and Event-Related Desynchronization (ERD) are highly correlated with motion and have high activity in the μ band (8Hz to 13Hz) and β band (13Hz to 30Hz) [5]. Convolutional filter sizes at the first layer were chosen to capture features of these frequencies, but a data augmentation strategy to present the μ and β bands was attempted. This strategy copied the 22 EEG channels, one set with a 6th-order Butterworth filter around the μ frequency and another set with a 6th-order Butterworth filter around the β frequency. The unfiltered set and two filtered sets were

concatenated to generate a (66×1000) sized sample. This was tried against three separate architectures - a deep CNN, shallow CNN, and CNN+LSTM. All three models experienced either the same performance with this augmentation or a degradation in performance. This indicates that either these frequencies were unimportant or the model was able to learn filters to distinguish features in these frequencies. Future work may want to change the extra features to a measurement of power in these bands with time.

3.1.5 Data Exploration and Alternate Tasks

The most significant finding during data exploration was of class imbalance in the test set - it was found that, while the training set was fairly uniform, the test set was biased towards the 'Right Hand' class. We believe that this class imbalance could have caused the dropoff in performance between validation and test sets for the model described above.

In addition to performing classification across all subjects in the dataset, we used this model to perform subject-specific classification. First, we found that model performance varied greatly between subjects, with subjects 2 and 6 having the lowest accuracy. When removing subjects 2 and 6 from the overall dataset to train a model, this improved performance by 4% on average. Second, standardization, which improved performance of the overall model, actually hindered performance on the per-subject models. We hypothesize that this is because the models were learning consistent differences in amplitudes per action for a given subject, but these differences were not always preserved from subject to subject.

A model trained on subject classification only achieved 99.8% test accuracy, and similar validation accuracy after only a handful of training epochs. This obviated any need to pursue a multi-task model on this dataset.

3.2. Model Performance

A significant part of our efforts were directed towards improving the performance of our shallow CNN model, which reached a baseline test accuracy of 60% on the task of subject-independent classification. Using the preprocessing steps described earlier, including PCA reducing the inputs to 16 dimensions, cropping out the final 500 samples of each example, and the per-channel standardization, allowed us to improve greatly upon this baseline. Best test and validation accuracy was generally found for PCA to 16-18 channels and dropout of 0.85, but the one-off overall best test accuracy of 73.14% was achieved by a model reducing to 20 channels. Overall, the contribution of PCA to test performance was roughly 3-4%. More typical test accuracy for this model was 71-72%.

Besides the single-dropout approach described previously, we added another dropout layer after the first fully-connected layer with a value of $p = 0.6$, but this did not yield any measurable gains. Changing activation function hindered performance and was not pursued further.

The second model that we implemented, the spatial convolutional neural network (SCNN), reached lower accuracy than the shallow CNN on the task of subject-independent classification; however, we did not spend a significant amount of time tuning hyperparameters and we only trained it with raw EEG data and no data augmentation techniques whatsoever. Considering this, the performance of this model was still reasonable, correctly classifying over 65% of the test dataset. There were several hyperparameters we would have tuned had we had more compute power available and more time. Some of them would have been the number of filters on the spatial convolutional layers, the number of filters on the temporal convolutional layers, the dropout factor, and the learning rate.

The SCNN+LSTM with attention mechanism was the best performing model on the task of subject independent classification. This result can be explained by two architectural decisions. First, the inclusion of separate spatial convolutional layers and temporal convolutional layers, which allows the model to learn useful spatial features independently from temporal features. Second, the inclusion of the attention mechanism, which allows the LSTM layer to learn what parts of the input sequence are more relevant to the task at hand. Kostas *et al*. [2] report a mean accuracy of 73.3% with a standard deviation of 13.6%, so our accuracy of 75% falls within the range reported by authors of the model. Amongst the hyperparameters we tuned for this model were dropout factor, choice of activation function, learning rate, and batch size.

References

[1] C. Brunner, R. Leeb, G. Müller-Putz, A. Schlögl, and G. Pfurtscheller. Bci competition 2008–graz data set a. *Institute for Knowledge Discovery (Laboratory of Brain-Computer Interfaces), Graz University of Technology*, 16:1–6, 2008.

[2] D. Kostas, E. W. Pang, and F. Rudzicz. Machine learning for meg during speech tasks. *Scientific reports*, 9(1):1–13, 2019.

[3] R. T. Schirrmeister, J. T. Springenberg, L. D. J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, and T. Ball. Deep learning with convolutional neural networks for brain mapping and decoding of movement-related information from the human EEG. *CoRR*, abs/1703.05051, 2017.

[4] P. Wang, A. Jiang, X. Liu, J. Shang, and L. Zhang. Lstm-based eeg classification in motor imagery tasks. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(11):2086–2095, 2018.

[5] Y. Yang, O. Kyrgyzov, J. Wiart, and I. Bloch. Subject-specific channel selection for classification of motor imagery electroencephalographic data. In *2013 IEEE International Con-*

ference on Acoustics, Speech and Signal Processing, pages 1277–1280. IEEE, 2013.

ECE C247 Final Project
Methods

Alon Krauthammer, Mark Kubiak, Chris Munoz, Eashan Samak

March 15, 2021

1 Detailed Results

Table 1 shows the validation accuracy and the test accuracy for each model we evaluated. In addition to the task of classification across all subjects, we trained our shallow CNN for the task of subject-specific classification. Table 2 shows the accuracy performance for subject-dependent classification.

Model	Validation Accuracy	Test Accuracy
Shallow CNN w/ PCA	0.768	0.731
SCNN	0.704	0.654
SCNN+LSTM	0.758	0.765

Table 1: Best accuracy results over all subjects in BCI IV dataset 2a.

Subject	Validation Accuracy	Test Accuracy
1	0.822	0.700
2	0.600	0.560
3	0.898	0.840
4	0.681	0.700
5	0.721	0.766
6	0.581	0.510
7	0.848	0.740
8	0.878	0.760
9	0.829	0.872

Table 2: Best accuracy results for each subject with Shallow CNN w/ PCA model.

2 Model Architectures

Layer	Output shape	Num. of parameters
Reshape	$(N, 16, 500, 1)$	0
Conv2D	$(N, 16, 476, 40)$	1040
Permute	$(N, 476, 16, 40)$	0
Reshape	$(N, 476, 640)$	0
Dense	$(N, 476, 40)$	0
Activation	$(N, 476, 40)$	35240
Average pooling 1D	$(N, 27, 40)$	0
Activation	$(N, 27, 40)$	0
Flatten	$(N, 1080)$	0
Dropout	$(N, 1080)$	0
Dense	$(N, 4)$	4324

Table 3: Architecture summary for Shallow CNN w/ PCA model.

Layer	Output shape	Num. of parameters
Input layer	$(N, 1000, 22)$	0
Reshape	$(N, 1000, 22, 1)$	0
Conv2D	$(N, 1000, 12, 100)$	1100
Conv2D	$(N, 1000, 1, 100)$	120000
Batch norm	$(N, 1000, 1, 100)$	400
Spatial dropout 2D	$(N, 1000, 1, 100)$	0
Conv2D	$(N, 977, 1, 40)$	96000
Conv2D	$(N, 954, 1, 40)$	38400
Conv2D	$(N, 931, 1, 40)$	38400
Conv2D	$(N, 908, 1, 40)$	38400
Batch norm	$(N, 908, 1, 40)$	160
Average pooling 2D	$(N, 45, 1, 40)$	0
Spatial dropout 2D	$(N, 45, 1, 40)$	0
Flatten	$(N, 1800)$	0
Dense	$(N, 4)$	7204

Table 4: Architecture summary for SCNN model.

Layer	Output shape	Num. of parameters
Input layer	$(N, 1000, 22)$	0
Reshape	$(N, 1000, 22, 1)$	0
Conv2D	$(N, 1000, 12, 100)$	1100
Conv2D	$(N, 1000, 1, 100)$	120000
Batch norm	$(N, 1000, 1, 100)$	400
Spatial dropout 2D	$(N, 1000, 1, 100)$	0
Conv2D	$(N, 977, 1, 40)$	96000
Conv2D	$(N, 954, 1, 40)$	38400
Conv2D	$(N, 931, 1, 40)$	38400
Conv2D	$(N, 908, 1, 40)$	38400
Batch norm	$(N, 908, 1, 40)$	160
Average pooling 2D	$(N, 45, 1, 40)$	0
Spatial dropout 2D	$(N, 45, 1, 40)$	0
Reshape	$(N, 45, 40)$	0
Birectional LSTM	$(N, 45, 152)$	95080
Flatten	$(N, 6840)$	0
Dense	$(N, 4)$	27364

Table 5: Architecture summary for SCNN+LSTM model.