Data Representation for Motor Imagery Classification

by

Andrew Festa

BSc Computer Science and Electrical Engineering

Rochester Institute of Technology

2020

THESIS

Presented to the Faculty of the Department of Computer Science

B. Thomas Golisano College of Computing and Information Sciences

Rochester Institute of Technology

In Partial Fulfillment of the Requirements for the Degree of

Master of Science in Computer Science

Rochester Institute of Technology

April, 2020

Data Representation for Motor Imagery Classification

Approved by supervising committee

Jeremy Brown Date
Supervisor

Ifeoma Nwogu Date
Reader

Philip White Date
Observer

Hans-Peter Bischof Date
MSc Program Director

To all the researchers out there who just want to have fun and explore the possibilities.

And if something good comes of it, then all the better.

Acknowledgments

As with any extensive effort, this work could not have been possible without the help and support of many others. I could not have done it by myself, and I make no claims about having the aptitude to do so.

To my advisor, Professor Jeremy Brown, who not only served to provide insight and feedback for this research, but also guided my learning over the past several years I have been at RIT for my Bachelor's and Master's degrees. He continually dealt with my overly stubborn nature and yet, against all better judgement, still agreed to act as my advisor for my thesis research.

For the other members of my committee, Professor Ifeoma Nwogu and Professor Philip White. Beyond your technical teachings, you helped me to improve my writing skills – both in and out of the classroom. And like Professor Brown, you were willing to humor me when I was being obstinate. Or when I was just looking to chat after class or in the hallways. The little details often add up to more than expected.

And finally to my other mentors – Jackie Corricelli, Craig Paradis, Jeffrey Bierline, and Napoleon Paxton – and my parents. You helped shape my perspective and knowledge of the world and encouraged me to view failure just as importantly as success. Without you, I would not have made it to where I am today. You have only yourselves to blame for what you have unleashed on the world.

Abstract

Data Representation for Motor Imagery Classification

Andrew Festa, MSc

Rochester Institute of Technology

April, 2020

Supervisor: Jeremy Brown

While much progress has been made towards the advancement of brain-controlled interfaces (BCI), there remains an information gap between the various domains involved in progressing this area of research. Thus, this research seeks to address this gap through creation of a method of representing brainwave signals in a manner that is intuitive and easy to interpret for both neuroscientists and computer scientists. This method of data representation was evaluated on the ability of the model to accurately classify motor imagery events in a timely manner.

The proposed data representation of electroencephalographic signals in the form of signal images was found to be able to perform adequately in the task of motor-imagery. However, the amount of time to record enough samples was on the scale of a fifth of a second following the onset of an input from the user. This time delay represents the minimum window size needed to classify the event, meaning that to reduce this delay would require a fundamental shift in the data that is acted upon to perform classification or to generate the signal images. Furthermore, the system performed better than expected, even in the face of random data, suggesting that the system may be relying on some external factor or undesired artifact present in the data in order to perform its task.

The strength of this approach came from its ability to be understood, visually examined, and altered in near-real-time in order to explore the data captured during a recording session. This was done after data had been recorded and involved altering sets of configuration parameters that affect the computations that go into generating a signal image. Namely, this included the window size, the function used to interpolate between two adjacent data points, and the amount of overlap of the windows. Effectively, this allows a researcher to playback the signal in an intuitive manner, watching for large shifts or edges in the images in order to detect large changes in the underlying data stream. Thus, while this approach may be unsuited for the task of classification, it would be an effective tool for conducting exploratory data analysis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recent advances in the hardware required for small-scale and non-intrusive methods of measuring brain activity offer an unprecedented level of potential for the development of brain-computer interfaces (BCI). Where this type of technology used to be accessible only to the professional medical community [2, 34], non-professionals are now able to approach this domain as a viable method of control. NeuroSky and Emotiv both provide cost-effective boards for recording electroencephalograms (EEG) for developers to use for experiments along with a thriving community for novices and experts alike. OpenBCI takes this a step further by open-sourcing both the software and the hardware for their boards, the Ganglion and the Cyton.

Despite the explosive growth of the field since the early 90's [39,45], drawing meaning from the understanding of the brain remains a difficult challenge of two parts. The first is the challenge of the forward problem which is an attempt to discern the expected outputs from the brain given an initial set of environmental stimuli. The other is the inverse problem which attempts to map the brainwaves back to the stimuli which gave rise to those patterns. The open-source community tends to focus on the inverse problem in an effort to build BCI and other control systems. The forward problem is generally left to large research labs with extensive resources. But to create a truly effective BCI, both the forward and inverse problems must be addressed in tandem to ensure that the system operates based on the theoretical truths of the functionality of the brain.

1.1 Problem Statement

The potential for BCI is often characterized by its potential to offer a non-invasive and unique means of communication and control in that it requires no muscle movement by a user [34, 3]. Development of such a system inherently requires an understanding of theoretical neuroscience as well as data science techniques [4], requiring teams to be proficient in numerous knowledge domains running the gamut from neuroscience to computer science to information theory. This necessitates communication between these various domains, communication which inevitably becomes challenging and begins to break down across domain boundaries.

One solution teams often take to combat this issue of the breakdown of communication is to limit the areas of expertise required to solve a problem. That is, teams of neuroscience will improve upon the capabilities of BCI systems by advancing the theoretical neuroscience, while teams of data scientists will attempt to improve classification and recognition of the event using data analysis techniques [41]. Effectively, this means that the domains remain divided and sharing information is wrought with the potential for misunderstanding due to differing experiences and language use.

An arguably more robust solution is to form a team with a diverse skill-set. Unfortunately, a team can only grow so large before becoming unmanageable, and there will still inevitably exist a gap in expertise areas due to the complex nature of such a system.

1.2 Motivation

Based on the aforementioned concern, it would prove fruitful to provide both researchers and non-professionals a common means of understanding between the distinct theoretical aspects involved in advancing the capabilities of BCI systems [20]. That is, provide a method for the subject matter experts, in this case, the neuroscientists, to communicate and provide the information and data to the computer scientists in such a manner that the approaches are able to leverage the current state-of-the-art methods for data analysis and classification.

With regards to the computer science side of the equation, neural networks are often cited as being black boxes in the sense that it can be difficult to reason about the how or why they make certain decisions, though progress is being made to this end particularly in the domain of computer vision. If the knowledge representation were able to be more closely aligned to the progress currently being made by the field of computer vision, it would offer the ability for the neuroscientists to leverage the research currently being undertaken by this field of computer science [31]. This could allow for the inner-workings of the brain to be more directly explored, potentially offering further insights into the forward and inverse problems previously discussed.

Providing these advancements to brain-computer interfaces offers a myriad of potential real-world benefits. With respect to humanitarian efforts, this technology could be used to increase the quality of life in numerous populations, including the elderly and the infirm. BCI's require a low-level of physical strength and capability and are ideally situated for individuals who are physically limited [47]. They could allow those bound to wheel-chairs to navigate the world without the use of their hands [25, 37], or they could restore physical autonomy to those with paraplegia.

From a more science fiction perspective, they could be used to build fully-immersive virtual reality systems. The immediate use that comes to mind of such an environment is likely for gaming, but it could also offer an unprecedented method for professional and developmental training. For example, such a VR system could allow for surgeons to practice their fine-motor skills and hone their techniques without requiring expensive, or even locally-situated, equipment; They could practice in their own homes on their own time. Soldiers and athletes could run through simulations without concerns as to injury or the confines of space. BCI's offer the potential to experience and interface with reality in a completely novel setting.

1.3 Thesis Statement

The work for this thesis will attempt to address the issue of bringing together solutions for the forward and inverse problems in EEG by offering a novel method of representing the multivariate electrical signals captured by EEG systems. In this way, developers and researchers outside of large, heavily funded medical institutions will be able to better advance their models due to a better understanding of the theoretical operation of the brain. The proposed method of presenting this data is similar to the form taken by spectrograms and will leverage the current understanding of the mind by building upon a common method of representing EEG signals: EEG montages. Neuroscientists, computer scientists, and hobbyists will all be able to more readily interpret the complex EEG signals, facilitating the advancement of research both towards understanding the function of the brain as well as how to use that understanding to develop better and more robust brain-computer interfaces.

1.4 Objectives

There are several aspects involved in building out a BCI system. Namely, it is required that data be able to be accurately recorded, labeled, and stored for training and future analysis. Additionally, the data must be able to be analyzed in near-real-time when acting as a control system. Developing such a solution requires both a hardware and software component. The hardware component will be the Ganglion board developed by OpenBCI as it is among the cheaper options of consumer-level EEG systems while still providing multiple measurement channels as well as access to the raw data read by the electrodes. The software component – primarily Python based – will allow for leveraging of community-developed machine learning and AI libraries, such as Keras and Scikit-learn. These tools will be used in the development of a method of representing the signals generated by the Ganglion.

My primary focus in conducting this research work is to present a novel method for representation of EEG data that is understandable for subject matter experts in separate domains. To this end, I propose a method of using signal values, collected from an EEG system, to construct images that are able to be used as training inputs for a classifier. The data representation of a set of EEG samples will take the form of an image, where each image generated correlates to a series of time samples of the data channels measured by the Ganglion board. Typically, these signals would be fed directly into a machine learning algorithm, such as a recurrent neural network.

BCI systems are intended to offer a method of communication and control to an outside environment [5], and any method of data representation of EEG should be able to be used towards this task. This desired characteristic will be tested by using the signal images of the EEG signals to train a classifier that is able to distinguish between a resting state and two motor imagery (MI) event-related potentials: moving left and moving right. From a high-level, the data representation must fulfill the following requirements:

1. Holds true to the theoretical principles of the underlying data
2. Easily interpretable by both subject matter experts and AI engineers
3. Offers reasonable time-delay following onset of an event

1.4.1 Data Acquisition

The data used to build out and validate the proposed system and method of data representation will be pulled from existing datasets as well as manually generated through use of an EEG headset. The use of existing datasets allows for a direct comparison against prior research [46] as well as facilitates development of the system due to the fact that many of these datasets have been pre-cleaned or offer a baseline method for cleaning. If there is a statistically significant disparity between the performance of the system on the pre-existing datasets in comparison to the manually collected data, this could signify that the data collected from the EEG headset may have been improperly collected or cleaned. The manually collected data is a required asset due to the necessity of having a method to collect and analyze, in real-time, the signals from a user's brain.

1.4.2 Theoretical Truth

The meat of the proposed approach comes in the form of actually creating the method of representing the data, and it is the first requirement which necessitates that the representation must not be in a form that leverages some undesired characteristic that may happen to be present in the dataset. For instance, when a user blinks their eye, a noticeable artifact is produced in the FpZ location. However, this artifact is not actually indicative of the desired event. The data representation must not attempt to use such a feature when attempting to classify the different classes of events.

1.4.3 Cross-Domain Understanding

The second requirement captures the essence of representing data in this manner; it is meant to bridge the gap between the areas of expertise offered by the neuroscientist domain experts and the skill-set of an artificial intelligence engineer. By requiring that the data be understandable to both sets of research experts, it should allow for a more direct line of communication about how and why the model may be performing as it is. This simplifies development of accurate and robust models and ensures the team is able to communicate more effectively.

1.4.4 Event Classification

The final requirement ensures that the machine learning model is actually able to use the data in this form in order to accurately classify between the desired types of events. If this were not the case, the proposed method of data representation would serve little purpose beyond being another method for representing the information, such as is done by current EEG montages. Furthermore, it stipulates that not only does it perform adequately, but it must be able to keep pace with current, state-of-the-art models. It is likely that certain parameters will have to be identified and tuned when used to create the signal images [43]. For instance, the effect of a stimulus is not expected to be present for longer than one second. There would be little to no reason to include such a long range of time in identifying a particular event. In fact, it is likely that the period of interest is much shorter than half a second. However, the actual length of time is unknown and is an area which must be explored in order to determine the optimal length of time, and thus the number of samples per image, for motor imagery event detection.

1.5 Experiment Phases

In order to address these stages of building out a BCI system, the research work was broken into three main phases: data acquisition, data preprocessing, and event classification. The first phase is where the actual raw data for the rest of the project was collected and stored for analysis. For manually recorded data, a low-cost board from OpenBCI was used, and the data collected was compared against a reference dataset that used the BCI2000 system for data acquisition. All the EEG data was stored as CSV and as HDF5 (Hierarchical Data Format version 5) files. CSV is easier to visually inspect whereas HDF5 allows for faster iteration of experiments due to its performance benefits.

The second phase of the research is where the measured data was transformed into the proposed data representation. The signals were broken into windows of different time-lengths: 0.2, 0.4, 0.6, 0.8 and 1 second intervals, and it was these windows which were used to build out the signal images.

The final phase acted on the signal images in order to build a model to classify different input events. The signal images were fed to the classifier for training and prediction in different combinations in order to evaluate how the system performed when presented with variable time-lengths following the onset of an event. Additionally, this phase evaluated the system in order to determine the efficacy of the proposed data representation for the task of motor-imagery classification.

Chapter 2

Background

Building out a brain-controlled interface requires knowledge spanning multiple domains and often requires a trade-off between theoretical truth and practical application. It is often preferred to create a control system that can be more rapidly and accurately implemented using a clever trick rather than relying on computational neuroscience theory behind brain-waves and the functionality of the brain. While this allows for progress to be made towards the goal of a BCI, it limits itself in its ability to expand to new applications. As previously stated, developing a true BCI requires tackling both the forward and inverse problems in tandem: learning what actions give rise to what brain patterns and learning how to correlate particular patterns to a given action by a user.

2.1 Prior Work

Much of the prior work in this field has been a two pronged approach: neuroscientists working to extract meaningful features from the generated signals and computer scientists working to use those features to perform data analysis and classification. That is, where neuroscience domain experts work on solving the forward problem, computer science experts often focus tackling the inverse problem.

The features used by neuroscience experts often rely on common digital processing techniques which are applied to methods of representing the signals, e.g. montages. For the work done by data scientists, the signals are often viewed as multivariate time series. Common classification techniques involve models well-situated to dealing with this type of data, such as recurrent neural networks [21, 42] or logistic regression models [26]. The EEG signal can be seen as holding information in a spatio-temporal form, and neuroscientists are able to perform feature extraction based on this. These features are non-linear in nature and can be effectively fed to either a typical convolutional neural network [30], recurrent neural network [10], or a recurrent convolutional neural network [53] for classification.

When approaching the problem of classification of motor imagery events, subject matter experts often perform manual feature extraction [46,50] based on the theoretical underpinnings of the neuroscience involved. These features often take the form of Hjorth Parameters [52], common-spatial patterns, Fourier or Laplacian transforms, or wavelet transforms, which are then fed into machine learning classifiers (SVMs, logistic regression models, artificial neural networks [49]). Evaluation of these models commonly reports on the accuracy of the models, despite the fact that the datasets are typically heavily unbalanced and skewed towards resting states or non-events [27]. Another method of evaluation is a transient analysis of the system with regards to the timing involved in recognizing and classifying an event [16]. In this method of evaluating the performance of classifying event related potentials, a plot shows the start of an event as boundary marks, similar to the plot shown in figure 2.1. The plot additionally shows where the system made predictions about the onset and end of an event. This then allows

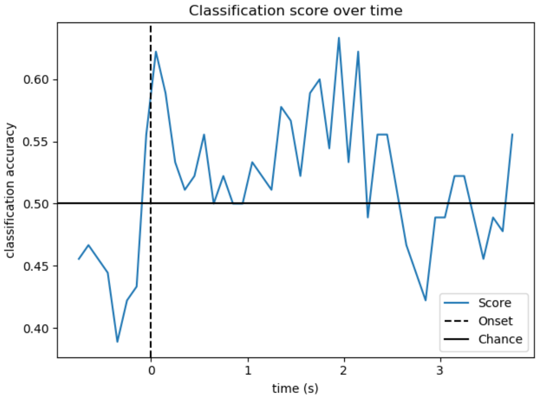


Figure 2.1: MNE Common-Spatial Pattern Accuracy

the researchers to visually inspect the ability of the model to accurately identify and respond to events along with how long it took for a particular event to be recognized, assuming that the event was successfully recognized by the classifier.

A third approach for solving this task is to employ reinforcement learning (RL) algorithms [9]. These have been shown to be able to not only learn to recognize event-related potentials, but also tackle the problem of error-related potentials where the system attempts to detect when a user thinks they have made a mistake [29]. Not only has this approach been implemented in augmented reality simulations [13], but also in the physical realm by training robots [24] to recognize and respond to EEG signals. Unfortunately, RL is not as widely understood or studied as other forms of machine learning and suffers particularly when faced with sparsity in the reward space, as is the case with EEG data for motor imagery classification.

2.2 Common Challenges

As this research focuses on the efficacy of the data representation for the task of MI classification and on the cross-domain understanding of the representation, it is desired that a system be in place that is capable of performing recording sessions of brain-wave activity. The Ganglion board was selected for this purpose. However, this board presents several particular challenges when using the released GUI tool meant to collect and display the readings from the channels on the board. The tool is configured to display the values streamed from the board along with the Fourier transforms, as provided by the tool itself. This information is not directly accessible except for visual analysis. To get it into a form usable for use outside of the OpenBCI environment, the data must be exported in some way or otherwise made accessible as a data stream. This presents a challenge in the fact that the developed library for interfacing with the data stream is not particularly stable or robust with regards to its ability to provide access to the raw data collected by the board. Furthermore, its inability to interface with a Windows environment due to the underlying Bluetooth library is a known limitation. Conversely, certain other libraries, such as NVIDIA's graphics acceleration library (CUDA), can be more easily configured on Windows. Selecting one platform for development requires sacrificing usability of one library for usability of another. In isolation, this is not a particularly challenging issue to solve, but it points towards a general trend in the libraries and code released by the community. Much of it tends to be developed to work in a lab environment. Configuring it to work in a particular setup requires additional legwork that has little to do with actual BCI or EEG research.

There is a common trope in machine learning that there is no substitute for clean data. This holds particularly true for EEG tasks due to the low signal-to-noise ratio (SNR) inherent in the data itself, regardless of the method used for collecting the data. The data can be poisoned by experimental setup, such as electronics or ambient noise in the test environment, or by use of dry versus wet electrodes. This is further confounded upon due to variances in the test subject themselves. EEG data is meant to capture the electrical potentials which occur when a user performs some action which requires use of the brain and is inherently dependent on the actual mental state of the user, including hunger, fatigue, and general contentment. Altering any of these states has the potential to be reflected in the EEG recordings when performing the same task. One solution is to continuously build out a larger and larger dataset which represents the user in various states of mind. This quickly becomes intractable due to the exponential growth of adding a single feature to the set of monitored states. Adding a single feature to monitor (hunger, sleep level, happiness, stress, etc) increases the number of potential mental states by a factor proportional to the number of discrete states of the new feature.

2.3 Intuition

Teams composed of different areas of expertise will inevitably face an issue of terminology. One domain will use terms and phrases which are either not common-place, or may even be contradictory, to similar terms used by the other domain. Using common phraseology can go a long way in establishing an effective method of communication between the various domain experts. Some teams attempt to solve this problem by having an ever-growing compendium of common terms. Just as adding new variable to track quickly becomes intractable, so too does this becomes likewise unmanageable. And it still offers the potential for confusion if there happens to be any missing terms or there are simply too many to learn. On the other hand, images and graphics tend to be more universally interpreted. The human mind perceives images in a fairly ubiquitous manner as the mind tends to search for particular features in an image, e.g. corners, edges, and color splotches. Leveraging this characteristic of how humans perceive visual information would allow expert of different domains to point more directly to certain properties or artifacts present in the image. In fact, it is partially this insight which plays into why convolutional neural networks perform well on the task of image recognition.

Intellect is often seen as an ability to recognize patterns and is the basis for intelligence quotient (IQ) quizzes. Performing such a task, however, is not strictly dependent on ones ability to recognize a pattern, but also on how an individual views the information in their mind. Some individuals view mathematical concepts as shapes which fit together in various ways. Viewing numbers in this way can prove beneficial in performing mental arithmetic operations or to extend complex mathematical concepts to new domains. This insight suggests that representing information in a visual manner will prove advantageous for machine learning models to learn to classify complex motor-imagery events. Rather than relying on expert and human oversight in order to guide the learning capabilities of a model, the model would be able to learn shapes and forms inherent in the data in order to derive potentially new insights about how to distinguish between the different events.

Chapter 3

Machine Learning

Machine learning is currently the predominant form of implementation for artificial intelligence. Industry is using it to transform all forms of solutions, from network security to digital marketing. In some cases, particularly in domains where there exists a known solution or where the cost of any inaccuracy is very high, this shift is unwarranted and overkill. Still, the trend exists that is making data to be the new gold, and companies are favoring the ability to process and understand that data.

Specifically with regards to neuroscience, several different techniques have been applied in order to distinguish between different events as well as different mental states [51] since the mid-80s. Despite the variability in the specifics as well as due to the nature of the problem itself, the classifiers are typically derived from a set of parent classification techniques, and the techniques employed are often representative of the background of the team of the researchers. For instance, many neuroscience focused teams employ techniques which allow them to leverage their theoretical understanding of the mind. This requires significant effort in performing feature extraction and is better situated to being fed to models such as logistic regression or support vector machines. Computer science minded teams tend towards classifiers following artificial neural network architectures due to the ability of these models to perform automatic feature extraction to an extent, requiring less of a deep-dive into the theoretical underpinnings of the neuroscience involved in forming the decision boundaries.

3.1 Typical Workflow

The key goal in any machine learning project is to answer a question. Does this image contain a cat? What language is being spoken in this audio clip? What is the user thinking about? To this end, it is important to clearly define this question and the environment in which the model will be placed. It is too often the case that project managers, clients, or even researchers have trouble clearly articulating these details, leading to ambiguity in architectural design and definition of what constitutes a minimal viable product.

The actual process for implementing a machine learning initiative varies between industry and academia. Even in each of those groups, the process is not consistent. Software engineering has established life-cycles that are meant to standardize and streamline the development of code. Despite efforts, an analogous methodology does not exist for data science. Still, there exists certain activities that must be achieved in order to accomplish the end goal of creating a machine learning model, steps which deal heavily with pipe-lining the data from the acquisition stage to feeding it to a model, be it for inference or prediction.

3.1.1 Data Acquisition

One of the defining characteristics of machine learning is that it operates, and learns, based on data. Once a project has been defined and scoped out, it is important to make sure that the data required for solving the problem either exists or is attainable in some way. This may be that the community, or some other entity, has released a dataset or such a dataset can be created by some means. Often, groups or researchers will release datasets in an attempt to spur research into a particular domain. For instance, the MNIST dataset is a large dataset of handwritten digits that can be credited with advancements in the field of image processing.

An important consideration when sourcing the data is how does the data actually correlate to the question at hand and does the data actually contain an answer to the question of interest. For instance, if you have a set of recipes for chocolate chip cookies, would it be reasonable to expect your model to be able to answer what the weather will be tomorrow? Now consider an open WiFi network at the nearest Starbucks. Is there a correlation between the location of the shop and the color of the cars in the parking lot? The intuitive answer would be ”no”, the two likely don’t influence each other. But maybe the Starbucks is located in the northeast United States, where cars are more likely to be dark, compared with the southwest, where they are more likely to be lighter colored. While a simple example, it illustrates that as the question becomes harder, sometimes it can be difficult to know how to construct a dataset and if the dataset actually contains an answer.

A final point is with regards to how much data to collect. Some problems inherently require more data and some models require more data. Support vector machines, for example, generally require less data than neural networks to get working results. However, neural networks will generally scale better with more data than support vector machines. Models type aside, the more data there is to work with, the more clearly defined the decision boundaries can be. There is an important caveat though. This increased amount of data also creates potential for certain classes to become intermingled. More data is better, in general, since then the model is able to learn a more representative answer, but it can also make it harder to find an answer if the answer is more convoluted given the larger dataset.

3.1.2 Preprocessing

In any real-world situation, it is almost never the case that the data will be perfectly formatted and clear of any errors. Maybe a sensor went on the fritz for a few seconds or certain rows of the dataset are missing values. The general term for dealing with these errors is data cleaning, and it aims to make the data uniform in structure, if not in distribution.

The actual process of cleaning the data is based on the question of interest. Specifically with EEG data, it is important to identify and mark bad channels, i.e. those channels where the electrode may have a poor connection or may be picking up on a high level of noise. Additionally, it should be formatted in such a way as to allow for simple iteration through all the data. Ideally, in a way that is intuitive. It wouldn't make much sense to format the rows of an EEG recording based on a sorting of the values of a single electrode.

3.1.3 Exploratory Data Analysis

This step is closely tied to the previous step of data preprocessing. In truth, the two are more of an iterative process. The researcher does some processing, explores the effect on the data, adjusts and does some more processing, sees how the new changes come into play, and so on, ad infinitum. The actual goal is to get a feel for what the data looks like and how it reacts under different conditions. Do the data points follow a Gaussian distribution? Is each data point independent of all the others? Does the presence of a particular data point influence another? In general, there are no set questions to ask when exploring the data. It is more of an art and often requires some level of underlying knowledge regarding the domain.

3.1.4 Model Building and Tuning

It is at this point that the data is actually fed to a classifier for learning and evaluating how the model performs based on a set of hyper-parameters. For the initial pass, these parameters will likely not be the optimal values and will require fine-tuning in order to improve the performance of the model. In the case of a decision tree, these parameters might be the depth, minimum node purity, split function, and class weights. This also requires a definition for what constitutes the best model. In the case of initially detecting cancer, maybe it is desired to allow for a high level of false positives in order to ensure that low level of false negatives. Or the model is to be deployed in an embedded system where memory constraints are an important concern. Reducing the depth of the decision tree might decrease performance, but will save on time and space complexity. Again, there is no magic formula for divining these optimal parameters.

3.1.5 Deployment

Once a model has been built that performs at an acceptable level, the final step is to deploy it such that it can be used to answer the question expected of it. Effectively, this means that some interface must be provided that allows for new data to be fed to the model and allows the model to give back its answer. This can take many forms from a command-line interface where data is entered manually as arguments to a program to a web-based agent that provides a rich user-experience and provides metrics about queries and answers over time. The way in which the answer is used is based on the use-case for the machine learning project, and the method of deployment reflects that critical decision of how to provide a user access to the trained model.

3.2 Structured Data

In a traditional approach for business intelligence, data has been analyzed by looking at excel reports, combing through tables, and generating graphs. This type of data is referred to as structured data; it has a consistent and defined structure, such as that of a database schema. The data is able to be viewed as sets of features and can often be moved around in the form of comma-separated values where each row represents a single data point. This is the form of data that machine learning has traditionally operated on as it follows the form of analysis typically done by humans. It allows experts to transfer their knowledge of a domain directly to an artificial intelligence agent and allows for the results from the model to be readily interpreted. If the model makes a prediction contrary to the prediction a subject matter expert might make, the subject matter expert can trace through the features the model is examining in order to pinpoint where and why the model might be making the error.

These types of models offer particular benefit to subject matter experts due to their ease of interpretability. As they are often based on maximizing or minimizing a measure of variance or information gain, a wide number of techniques can be applied to aid in understanding the method by which this model is able to perform classification which is a large benefit over certain forms of black box models, such as neural networks which require more expertise to reliably interpret. Additionally, this type of model explicitly operates on extracted features, allowing those same experts to understand the actual information the model is using to perform its predictions.

3.2.1 Logistic Regression

Logistic regression relies primarily on a statistical analysis of the classes involved in a logistic model. It observes the samples, or data points, and attempts to build a function which mathematically represents the samples. In the case of analysis of EEG data for motor imagery events, each sample is a measurement of the electrode values at a particular point in time [49]. Each event is described by various features, such as entropy or spatial-temporal spectral energy. Each sample can be considered as a set of parameters, and the regression model fits the best hyper-plane to describe those parameters.

3.2.2 Support Vector Machine

Just as with logistic regression, support vector machines operate on a discrete set of features. This does not mean that each feature must be discrete; it means that the information presented to the model can be used to form a hyper-plane which differentiates between the various classes present in the dataset. It offers a similar benefit to subject matter experts in that they are easily understood with respect to how and why they are making their predictions. Furthermore, they offer a benefit over neural networks in that they typically require less information in order to perform adequately for a large number of tasks. This is particularly useful in the task of motor imagery classification as it can cost a significant amount of resources to preform a single EEG recording session.

3.3 Unstructured Data

Where structured data can be thought of as tables and laid out information, unstructured data is essentially data in any open-ended form. The classic example is images, as the information content in an image comes from how it is interpreted by a human. To a computer, there is no standard or set structure to an image. An image can take many forms, both in an abstract and technical sense.

This offers the potential benefit of not relying on expert input to shape the data or to provide guidance as to where the model should look for how to build its decision boundaries. The model is able to build its own intuition regarding the problem and is capable of providing new insight into the search space for the domain itself. However, this same flexibility makes interpreting the actions and results of such models extremely difficult, giving them the reputation for acting as black boxes. Where statistical techniques can be applied to distributions of features, a generalizable technique does not exist for models that operate on unstructured data.

3.3.1 Convolutional Neural Network

Convolutional neural networks (CNNs) have been shown to perform extremely well for the tasks of computer vision and pattern recognition. This is primarily due to the fact that images and patterns can be interpreted as a discrete signal. Convolution is able to match particular signal forms as well as recognize arbitrarily complex forms in signals. Furthermore, they offer the benefit of operating directly on images rather than requiring manual feature extraction. In addition to training being more automated than models such as logistic regression and support vector machines, these models have the benefit that they are able to recognize latent variables which may be hidden or unknown to human researchers. Unfortunately, this can also obscure reasoning behind why a model makes a particular prediction.

3.3.2 Recurrent Neural Network

Recurrent neural networks (RNNs) are typically an intuitive choice when dealing with time-related information, such as the stock market or audio signals. This is largely due to the fact that they are specifically designed to retain past information for a length of time. For example, in a long short-term memory (LSTM) architecture, a typical cell is comprised of an input gate, an output gate, and a forget gate. These three gates work in conjunction to retain information for an arbitrary length of time based on activation states of the cell. LSTMs specifically were designed to deal with the issue of vanishing and exploding gradients, and other RNN architectures have been specially crafted in order to perform better in domains [15].

RNNs have the unfortunate characteristic that they are considered difficult to train in the sense that it can take a lot of information in order to properly learn the time-dependent functions which model the decision boundaries. They can be considered more sensitive to spurious or uncleaned data, which is often a defining characteristic of EEG signals. It is expected that this classifier model would preform worse when attempting to apply transfer learning; the trained model will likely not be as successful at generalizing to new users even if it is able to generalize to different tasks for a particular user.

Chapter 4

Neuroscience

At its core, neuroscience studies the nervous system in order to discern and understand the workings of the mind. In part, it views the mind as a complex neural circuit where the firing or suppression of neurons gives rise to consciousness and thought processes. Since its inception, dating back to ancient Egypt, the field has grown to encompass all forms of higher-order processing and perception, including visual perception, language, and motor-execution. Modern neuroscience is divided into many sub-fields, but the ones of interest for this research effort are primarily the fields of neural circuits and computational neuroscience.

4.1 Typical Workflow

Just as data science has a typical workflow that most projects follow, most neuroscience efforts dealing with brainwave classification follow a general set of steps. The details of each step and other minutiae will likely differ from project to project, but the core guidelines remain constant. The MEG/EEG Analysis and Visualization open-source library (MNE) [19] describes this workflow as one of their *cookbook* tutorials aimed towards new researchers and practitioners [1]. In this chapter, they break down the workflow into three major sections:

1. Preprocessing
2. Epoching and evoked data
3. Source localization

By and large, this is a rather good breakdown of the basic steps to take when performing an MEG/EEG project. However, this workflow has the implicit assumption that data has already been collected and is available for use. The package itself even has convenience functions for downloading and formatting the data as is expected by the library for further processing. Beyond this, several of their steps can be generalized to some degree, and over the course of this research project, I found a better characterization of the steps to be as follows:

1. Data Sourcing
2. Preprocessing
3. Data Transformations
4. Source Localization

In many ways, this closely resembles the workflow described previously in section 3.1, as it is to be expected as the alterations to the MNE workflow are largely based on the experience gained while performing a classification task using MEG/EEG data. The differences between the two further suggest a slight variance in the methods employed by neuroscience experts versus computer science practitioners. The machine learning workflow attempts to rely on the merit of the process itself and remove those steps which may offer variability or may require subject domain experts. The MNE workflow explicitly reinforces the importance of such experts when conducting research efforts.

4.2 Motor-Imagery Classification

The actual problem explored during the course of my research is an attempt to map user brainwaves into a desired hand movement, as described is 1.4. This task is referred to as motor-imagery which is the state of mind in which a user is simulating a given action. It has been found to have close parallels to motor execution, a task where a user actually performs the action in question [6,48]. This suggests that there is a shared representation in the brain for both of these mental tasks [23]. Though this interaction is far from simple, this connection is believed to be due to that both actions can give rise to motor cortical excitability [35]. Still, this similarity means that the data recorded from one task should carry over from one task to the other, and a recording method for one task should be just as suitable for the other task. Electroencephalography is one such recording method often used for BCI systems and is the one I used for this research.

4.2.1 Electroencephalography

Electroencephalography has roots as far back as 1875 when Richard Canton presented his research on the electrical firings present in small mammals [22,8]. Canton is further credited with being among the first to record human brainwaves [36], although it is the work of Adolf Beck which first gave rise to the concept of brain waves [8]. Together, these works served as a base of research for many other pioneers in the field, including Vladimir Vladimirovich Pravdich-Neminsky, who published on evoked potentials in 1912 [40].

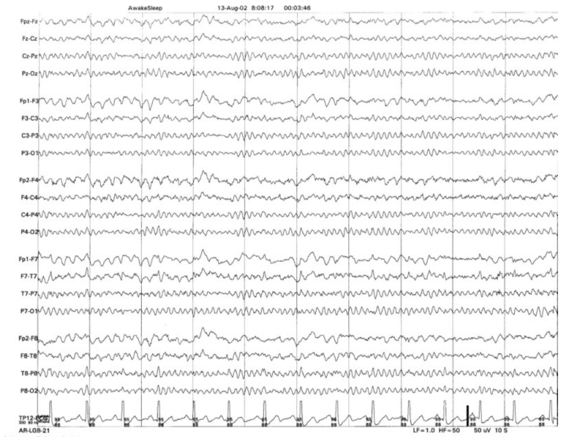
Motor-imagery, the task explored in this research, is closely related to these types of signals as motor-imagery falls under a set of signals known as event-related potentials. Although often used synonymously, the relevant difference here is that evoked-potentials occur when a user is presented with a stimulus, event-related potentials occur when a user performs some task or [38, 28].

The actual process by which these brainwaves are recorded is by attaching small electrodes to the scalp and measuring the electrical potential between that electrode and a reference electrode, commonly a floating ground [11]. This works on the premise that the brain is a network of neurons that communicate amongst one another through synapses, which act to inhibit or excite activity [7, 14]. Although the electrical signal from a single neuron is minuscule and hard to measure without a direct connection, when hundreds or thousands or neurons fire in parallel, an electric field is generated that can be propagate through tissue and bone. It is this potential that can be measured by the electrode connected to the scalp.

In order to improve upon the spatial resolution obtained, many such electrodes are connected around the entirety of the head or localized around a specific region. These placements often follow a known standard, referred to as the 10/20 positioning system. In this context, the 10/20 refers to the 10% or 20% interelectrode distance [7].

4.2.2 Potential Issues and Limitations

One issue with EEG is with regards to a user's ability to manipulate these various states of mind. Various studies have shown that not everyone is equally capable of manipulating their brain patterns in such a way as to be able to effectively use a BCI system or manipulate their brainwaves. It is estimated that between 10% to 15% of the population are incapable of using effecting their brainwaves enough that they would be able to use an EEG-based BCI system [33, 25]. As such, researchers often manually examine the trials and simply disregard those trials which prove to be noisy or where the electrode does not appear to be collecting good and clean data. This tends to be a challenging tasks involving a high level of expert insight and training and is costly both with regards to



© American Epilepsy Society

Figure 4.1: Eye Flutter Artifact [7]

time as well as personnel as this expert must dedicate their knowledge towards cleaning the dataset. Unfortunately, this task cannot be crowd-sourced in the same manner as many other data preparation tasks can be, eg. sorting and labeling images of animals, due to this high level of expertise required in understanding the various ways in which EEG signals can be represented; It is required that experts interpret these montages in order to distinguish between key features and undesired noise and artifacts.

For example, consider the the transverse bipolar montage depicted in figure 4.1, which depicts eye flutter by a user. This artifact occurs when the user moves their eyes, inducing a changing potential in the frontopolar leads. It would not be reasonable to expect this to be common knowledge which could be easily identified by a non-expert, though it has been found that similar types of tasks can be gamified with rather decent results, as was done by MIT in identifying protein folding structures. However, this still requires validation by a subject matter expert in order to verify that the data has been properly cleaned and prepared.

There are several issues with this approach. Arguably the largest is that it focuses work on only a single type of signal. From a theoretical perspective, the region of the brain that is most active when a user is performing some action is dependent on the actual action or activity. Figure 4.2 shows various regions of the brain. Consider the occipital lobe. This region is responsible for processing visual information, while the

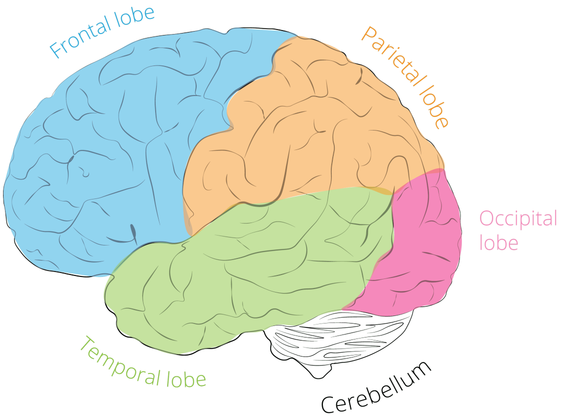


Figure 4.2: Regions of the Brain [14]

temporal lobe is active in matters related to speech and natural language [14]. While researchers and developers have been attempting to solve problems such as correlating signals to language, approaches have often fallen back on clever manipulations of measurements taken from the prefrontal lobe rather than measuring and analyzing signals from the portions of the brain that are primarily responsible for these actions. While certain levels of success can be attained in this manner [17], it limits the possibilities by which humans may be able to interact with computer systems through thought alone.

Chapter 5

Data Acquisition

There are a myriad of challenges to overcome when performing the recording sessions necessary for gathering data for BCI systems. For the non-neuroscientist, there exists a further issue in that it can be difficult to verify if a recording session gathered adequate clean data and how to inspect the signals for abnormalities and artifacts.

5.1 Physio Dataset

In order to reduce uncertainty as to whether errors may be due to the data representation or the recorded data, an external dataset was used to first build out and evaluate the proposed method of data representation. This dataset, called the *EEG Motor Movement/Imagery Dataset*, is provided for research use by PhysioNet [44, 18]. Not only is this dataset expertly collected and meticulously verified, but it is also used by several solutions for evaluation of BCI systems making it a perfect candidate not only for verifying the efficacy of the proposed method for classification of motor imagery events, but also for comparison against current techniques used for performing the same task.

5.1.1 Recording Session Protocol

This dataset was built using the BCI2000 system, which is a mid to high level system compared to the Ganglion, the OpenBCI board used for manual data collection. The BCI2000 system has 64 channels, compared to the Ganglion's 4 channels, and is more aimed towards neuroscientist researchers rather than the open-source and non-professional communities. The actual data recorded is comprised of 6 different trial types for 109 different subjects, and each trial is either 2 or 3 minutes in length, depending on the trial type. The actual trial types, along with Physio's description of each trial, are shown in table 5.1. Each of the baseline trials are 2 minutes in length, while the remaining trials are 3 minutes in length.

5.1.2 Subtleties and Nuances

When dealing with this dataset, there are a few small details that bear mentioning as they influence design decisions.

1. Each subject performed each baseline trial once and each non-baseline trial 3 times, resulting in a grand total of 1526 recorded trials. However, not all of these trials were used in the development and evaluation of the proposed data representation.

Trial Type	Description
Baseline	Baseline recording of brain activity when the user's eyes are open
Baseline	Baseline recording of brain activity when the user's eyes are closed
Motor execution	A target appears on either the left or the right side of the screen. The subject opens and closes the corresponding fist until the target disappears. Then the subject relaxes.
Motor imagery	A target appears on either the left or the right side of the screen. The subject imagines opening and closing the corresponding fist until the target disappears. Then the subject relaxes.
Motor execution	A target appears on either the top or the bottom of the screen. The subject opens and closes either both fists (if the target is on top) or both feet (if the target is on the bottom) until the target disappears. Then the subject relaxes.
Motor imagery	A target appears on either the top or the bottom of the screen. The subject imagines opening and closing either both fists (if the target is on top) or both feet (if the target is on the bottom) until the target disappears. Then the subject relaxes.

Table 5.1: PhysioNet Trial Types

The only trials used were the ones where the subject was instructed to imagine moving either their right or left hand. The baseline trials were purposefully ignored in order to try and limit the ability of the system to differentiate between rest and normal brain activity versus the desired events.

2. The BCI2000 system samples each electrode at 160 Hz. This value is used to correlate sample index to time as well as used in computations of window lengths, which is explained in greater detail in 6.1.

3. A FIR band-pass filter was applied to the signals to only pass frequencies between 5 Hz to 50 Hz. This frequency range was selected as it allows for limiting the signals to the frequencies of interest and is one of the ranges provided by OpenBCI filters, which was used when manually recording data.

4. The events are annotated as the three strings shown in table 5.2 and are labeled based on the presentation of the target. Theoretically, it could be the case that the subject simply did not adhere to the instructed protocol, in which case EEG activity would still be recorded but would other wise be mislabeled. This could be controlled for in the case of motor execution trials as an action could be visually

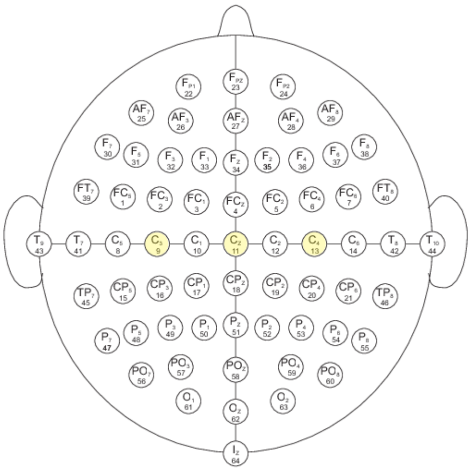


Figure 5.1: Physio Electrode Positions

confirmed by the tester. However, in the case of motor imagery trials, there appears to be an implicit level of trust that the subject is performing the correct action at the correct times.

Event	Description
T0	Rest
T1	Onset of motion of the left fist
T2	Onset of motion of the right fist

Table 5.2: PhysioNet Events

5. The data is provided as EDF files, which is a common format for providing this type of data. This is relevant as this data format is supported by the MNE library, but the relevant information is extracted using this library and then transformed as part of the preprocessing steps.

6. The electrodes were placed on the subject according to the international 10/10 system, as shown in figure 5.1. However, due to the theoretical considerations outlined previously in section 4, only the $C3$, Cz, and $C4$ channels were used when building the system. However, the reference implementation provided by MNE uses all 64 channels for evaluating the features extracted.

Figure 5.2: Ganglion Board

5.2 Manually Recorded Dataset

Part of the desired outcome for this research was to explore the ability to use the proposed data format for development of a BCI system. As such, there has to be at least some capability to record and analyze EEG activity in real-time. This was done using the Ganglion board from OpenBCI, shown in figure 5.2. The user was connected to the board using passive electrodes and presented with a random stimulus on a timer. They were then expected to imagine moving either their right or left hand, based on the stimulus presented. Additionally, there was a "rest" prompt, to which they were expected to just relax and not imagine any action.

5.2.1 Ganglion and Headset Configuration

One of the earlier challenges to overcome was actually getting the EEG data into a coding environment in a real-time manner. OpenBCI provides a GUI for visualization of the recorded signals, and they have a Python library that is meant to allow developers to interface with the board using Python. Unfortunately, it relies on a Bluetooth library that is only compatible with Linux and Mac. While not an insurmountable issue, it was desired that the system be able to work cross-platform.

Rejected Approaches

The first two approaches were to try and communicate directly with the board or with the Electron hub that the OpenBCI GUI uses to communicate with the board, thereby bypassing the GUI entirely. The first solution, direct communication with the board, was rejected simply due to the fact that the Simblee board, which is the actual breakout board used by the Ganglion, was discontinued, making it difficult to find compatible drivers for directly interfacing with the board.

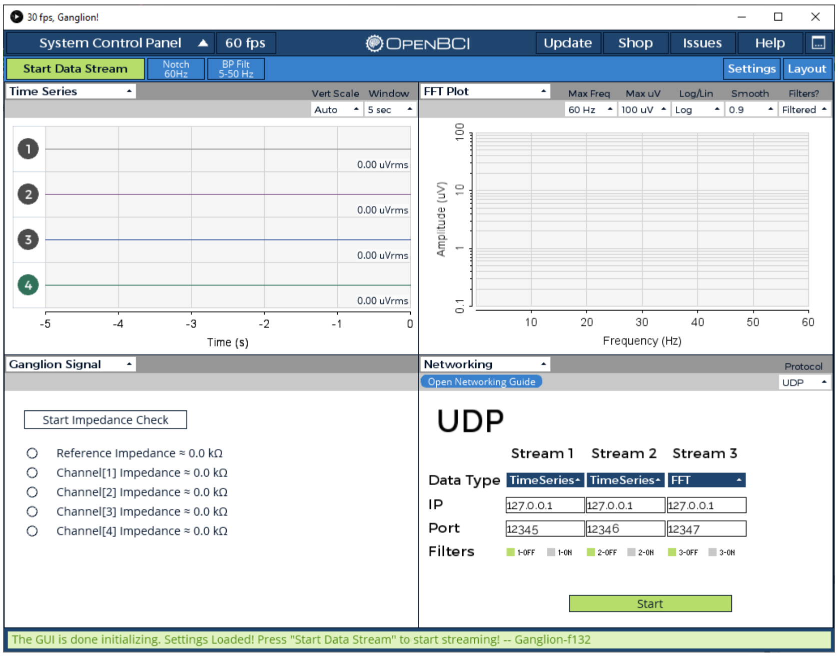
The second option – communicating with the Electron hub – was found to be a suitable solution and was pursued with a fair degree of success. This solution was able to successfully establish a connection with the board and record the desired data from the board. The reason for moving away from this solution was due to the complexity when presented with another solution that was simpler and more stable.

Final Approach

The final method of acquiring data in a real-time manner was by having the OpenBCI GUI establish the connection to the board and then having the Python environment communicate with the OpenBCI GUI over UDP using the Networking widget depicted in the bottom right of figure 5.3. This provides the benefit of a simple solution that is suitable for the desired purpose as well as allowing for use of OpenBCI's filtering capabilities.

The filtering capabilities of the GUI can be seen towards the top left of figure 5.3. The first is a notch filter that is meant to filter out noise from mains lines, which operate at 60 Hz in the USA. The second filter is a band-pass filter that allows for several different ranges.

The last benefit of using the GUI was in ease of checking the impedance values for each electrode using the widget shown in the bottom left of figure 5.3. As discussed



Vert Scale: Auto
Window: 5 sec

Max Freq: 60 Hz
Max uV: 100 uV
Log/Lin: Log
Smooth: 0.9
Filters?: Filtered

○ Reference Impedance ≈ 0.0 kΩ
○ Channel[1] Impedance ≈ 0.0 kΩ
○ Channel[2] Impedance ≈ 0.0 kΩ
○ Channel[3] Impedance ≈ 0.0 kΩ
○ Channel[4] Impedance ≈ 0.0 kΩ

Protocol: UDP

	Stream 1	Stream 2	Stream 3
Data Type	TimeSeries	TimeSeries	FFT
IP	127.0.0.1	127.0.0.1	127.0.0.1
Port	12345	12346	12347
Filters	🟩 1-OFF ▨ 1-ON	🟩 2-OFF ▨ 2-ON	🟩 3-OFF ▨ 3-ON

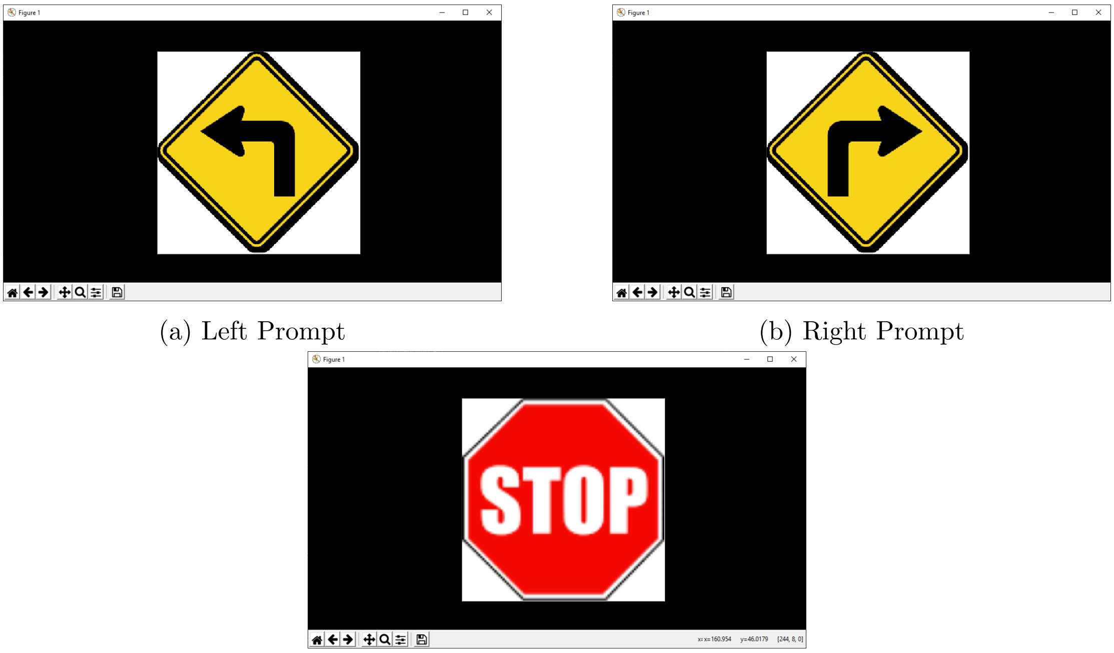
The GUI is done initializing. Settings Loaded! Press "Start Data Stream" to start streaming! -- Ganglion-f132

Figure 5.3: OpenBCI GUI

in section 4.2.1, we want these impedance values to be as low as possible, ideally at or below 5 kΩ, in order to help ensure that the signals of interest flow to ground through the electrodes rather than over the surface of the user's head.

5.2.2 Recording Session Protocol

As the Physio dataset is the reference dataset, the method for manually collecting data seeks to emulate that process as much as possible. Specifically, there are several key decisions that this protocol copies from the creators of the Physio dataset. There are three prompts that are presented to the subject: 'stop', 'left', and 'right', shown in figure 5.4. The subject was instructed to continue to imagine moving the hand that corresponds to the given prompt in the case of the latter two prompts. In the case of the 'stop' prompt, they were to relax and not imagine moving either hand. Both the EEG samples and the events were recorded, and a single sample (which is comprised of three values – one from each electrode) is marked as the current event. An important thing to note is that this protocol follows the decision to record when the stimulus was



(a) Left Prompt

(b) Right Prompt

(c) Stop Prompt

Figure 5.4: Stimulus Prompts

presented rather than when the user reacted to the event.

Despite the similarities in the process, there were several changes made to the method of the recording sessions. In the Physio dataset, each event is separated by about 3 seconds. When manually collecting the data, the time between each stimulus was set to an average of 5 seconds. A 20 % jitter was also added to the delay between generating a new prompt. That is, on average, the delay is 5 seconds. But the actual delay for any given delay is $5 \pm 20\%$ seconds, or more exactly, in the range $(4, 6)$ seconds. Furthermore, in the Physio trials, each trial was 120 to 180 seconds in length. However, when performing the manual collection trials, each trial was 120 seconds long.

The last distinction is in regards to generating the prompt. In the Physio dataset, the subject was aware of the next prompt in the sequence. When manually collecting the data, the user was not aware of the next prompt, and there was no guarantee whether or not the same prompt might appear multiple times in succession.

5.2.3 Experiment

The testing performed for this phase of the research focused around the ability to cleanly record data and tested various configurations for performing the recording sessions. Two headsets were tested on the basis of ease of setup, quality of recorded data, and physical user comfort. Another consideration could have been how the headset affected a user's state of mind (e.g. anxiety levels); However, this was deemed out of scope and not further explored in the course of this research. For both tests, the headsets supported many more electrode positions than able to be used by the Ganglion board. The Ultracortex "Mark IV" supports up to 16 EEG channels and the electrode cap supports up to 21 channels, while the Ganglion only supports up to 4 channels. For symmetry, only three channels were used: $C3$, CZ, and $C4$, which leaves open the possibility of using another position as an additional reference, such as using FpZ in order to better detect and remove artifacts due to blinking. However, this was also deemed out of scope and not further explored in the course of this research.

The final points of note are with regards to impedance checking and signal filtering. Prior to starting a recording session, the impedance was checked for each electrode. If it was over some reasonable threshold, the headset was adjusted in order to lower the impedance to a more acceptable level. Finally, the notch filter was also set to block noise around 60 Hz, as this is the frequency of the mains lines, and a band-pass filter was set to filter out all frequencies outside the range of 5 Hz to 50 Hz.

Ultracortex "Mark IV"

The first headset tested was a modified version of the Ultracortex "Mark IV" headset from OpenBCI, shown in figure 5.5. The modification consisted of an added belt restraint on the inside of the base of the headset. This aided in securing the headset to the user's head while also dampening the effects of any movement on the electrode positions. The electrodes were the dry, spiky electrodes that come with the headset, and the headset was tightened to the point that it did not shift easily under slight movement by adjusting

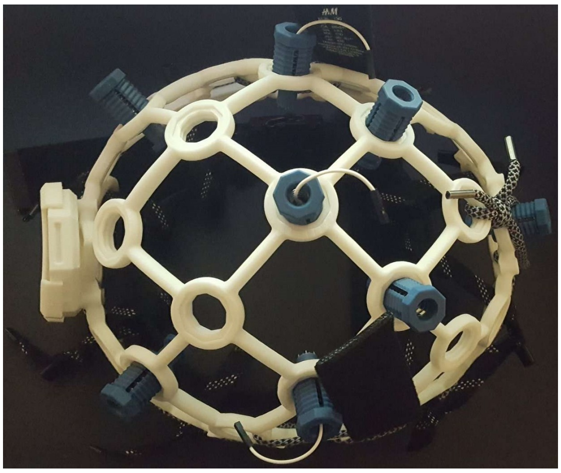


Figure 5.5: Ultracortex Headset

the belt loop on the inside rim. The neck strap applied a level of downward force on the spring-loaded electrodes, ensuring greater contact and lowering the impedance of the dry electrodes. Finally, it was found that applying a small amount of electrode gel further decreased the electrode impedance without many of the common pitfalls and troubles of wet electrodes.

After several tests, the typical impedance of an electrode was found to be around 18 kΩ, though it could be as low as 8 kΩ. It also was not uncommon for an electrode impedance to be in the range of 45 kΩ, with no amount of adjustments seemingly able to reduce this to any degree. Setting up the headset on my own head takes about 2 minutes, with the majority of the time spent making slight adjustments to reduce how much the headset shifts under slight movement. Setting it up on another person takes longer – on the scale of 5 or more minutes. Largely, this is due to not being able to feel how it sits on my head and the adjustments have to be made based on the other person's feedback and how much the headset shifts under slight movement.

Electrode Cap

The electrode cap, shown in figure 5.6, proved to offer a different set of challenges, namely with regards to setting it up on another individual versus myself. Compared

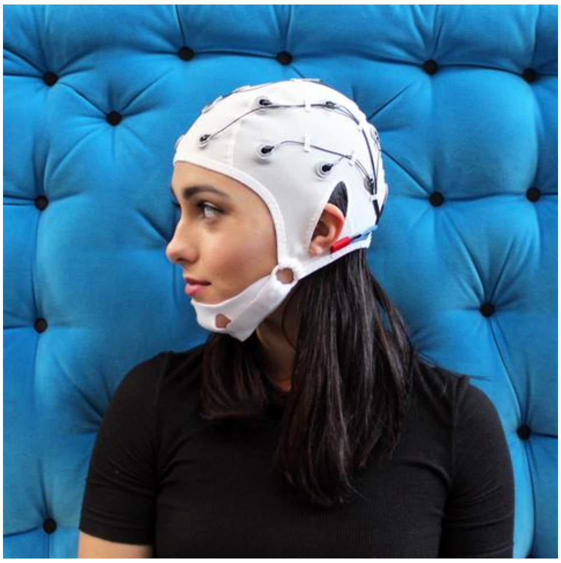


Figure 5.6: OpenBCI Electrode Cap

to the Ultracortex ”Mark IV”, the electrode cap took about 2 to 3 times longer to set up on myself, though on another person, it took considerably less time – about 4 minutes on average. The difficult part of setting it up on myself tended to be locating the correct electrode positions to inject with electrode gel, which was not as much of a problem when setting it up on another due to the fact I could easily see and locate the correct positions. Furthermore, it was much more comfortable to wear and was easier to tighten and adjust such that it sit well on either my head or someone else’s head. The real benefit of the electrode cap came from the impedance values of the electrodes. Not only was the impedance on the scale of $5\,\mathrm{k}\Omega$ to $8\,\mathrm{k}\Omega$, but this value was much easier to achieve consistently without needing much in the way of adjustments.

5.2.4 Recording Results

Compared to the unmodified Ultracortex headset, the electrode cap initially offered more promise with regards to recording clean data. The fact that it was made of a softer material also made it more comfortable to wear for extended recording sessions. However, after modifying the Ultracortex headset, it shifted significantly less when the user moved their head due to the fact that the rigid, plastic structure was not providing the support keeping the headset against the user's head. This alteration also made it much more comfortable to wear versus the base headset. Given its ability to record user data reliably and comfortably, all trials were recorded using the modified Ultracortex headset.

When performing the trials, the Ganglion was connected such that the $C3$ electrode was connected to channel 1, the CZ electrode was connected to channel 2, the $C4$ electrode was connected to channel 3, the reference electrode was connected to the right ear, and the driven-ground electrode was connected to the left ear. In total, 5 trials were recorded using this configuration and were meant to serve as the main set of trials for evaluating the efficacy of the data representation using raw, manually recorded data.

In addition to the standard recording protocol outlined previously in section 5.2.2, two additional sets of trials were recorded. The purpose of these trials were to measure how the system responded to known bad data. Ideally, this would provide further insight into the efficacy of the data representation in terms of learning some characteristic of either the recording method or the data representation. This was briefly commented on as a potential pitfall of the Physio dataset in section 5.1.2 when describing the labels of the events presented to the subject. In the Physio dataset, all of the trials were assumed to be clean and no checks were provided that the subject truly acted as instructed.

The first additional trial was named 'disconnected'. These sets of trials were recorded with the headset completely off the head of the user and with all the electrodes tied together. In this configuration, the impedance is very low (approximately $1\,\mathrm{k\Omega}$ to $1\frac{1}{2}\,\mathrm{k\Omega}$) and any signals measured are due solely to environmental noise. It was easy to generate a large number of these trials as it simply required setting up a script to continually run the same trial configuration multiple times. 3 different sets of trials were recorded in this configuration, the details of which are shown explicitly in table 5.3. Doing this facilitates evaluating not only how the system responds to random noise, but also if there is any change based on how much data is fed to the model during training. Across all 'disconnected' trials, 9 trials were recorded, for a grand total of 18 minutes of recordings using this configuration.

Subject Name	Number of trials
disconnected_01	1
disconnected_03	3
disconnected_05	5
main	5
random	5

Table 5.3: Number of Trials per Configuration

The second trial was named 'random', and these trials were similar to the baseline trials recorded in the Physio dataset. The subject was hooked up to the headset as described for the actual trials, and the recording session proceeded as outlined. However, the user was instructed to not pay attention to the prompts and to just continue on with any normal activity as desired. Each sample was labeled as the stimulus currently presented, meaning the labels were effectively random as the user should not have been aware of which prompt was currently being displayed at any point during the trial; There should be no method for correlating the present stimulus with the activity the user was actually performing. As with the main set of trials, 5 trials were recorded using this configuration.

Chapter 6

Data Representation

When considering the design of the data representation, it is important to keep in mind the actual purpose and use case of a control system. It must be both reactive and accurate, and it would be further desired that it offers some level of variable sensitivity while erring on the side of inactivity. Also, it would not be suitable for most use cases if an erroneous signal was interpreted as an input, thereby causing the system to enter a state of meta-stability as the user attempts to correct for the invalid input and causing further erroneous inputs to the system.

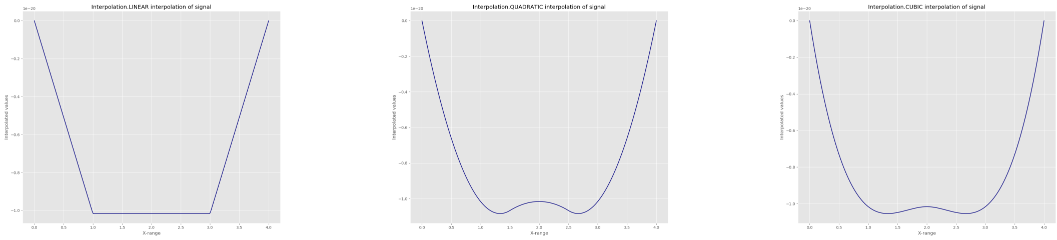
Ignoring the reference channel, there are three signals of interest: $C3$, CZ, and $C4$. These three locations are adjacent to each other over the central sulcus, and at any given point of time, these three signals are 3 values measured at discrete locations on the head. However, brain activity is not discrete. It is characterized by magnetic and electric flows and fields. Further complicating the situation, the fields are influenced not only by these three points, but also all surrounding areas of the brain, and thus, it would be more true to the theoretical operation of the brain if the system were to operate on a representation of the field rather than directly on discrete values.

6.1 Signal Image Construction

The first step in recreating a representation of these fields is to define a sort of spacing between each signal point, which can be viewed as being placed at an x-coordinate along the surface of the head. For demonstration purposes, examine three example points at a single point in time and define the spacing to be 25 points between each signal location. Tie each end of the plot to zero as a ground since these locations are physically similar to the reference and driven-ground electrodes on the head.

The next step is to apply an interpolation function to these five points. Scikit-learn has a built in 1D interpolation function that requires specifying a dimensionality: linear, quadratic, or cubic. Taking these point as an array results in a 1D array of y-values over an x-range, allowing the signal, at a single point in time, to be visualized as a 2D plot, as shown in figure 6.1.

In preparation for the next step, it is good practice to normalize these values to the range $[0, 1)$, and it is at this point that a column for a single time-step has been computed. Concatenating several of these columns together yields a 2D array of single values, quite similar to a method of representing a gray-scale image. While the height of the image is defined by the previously decided spacing, the width is defined by the number of time-step columns concatenated together.



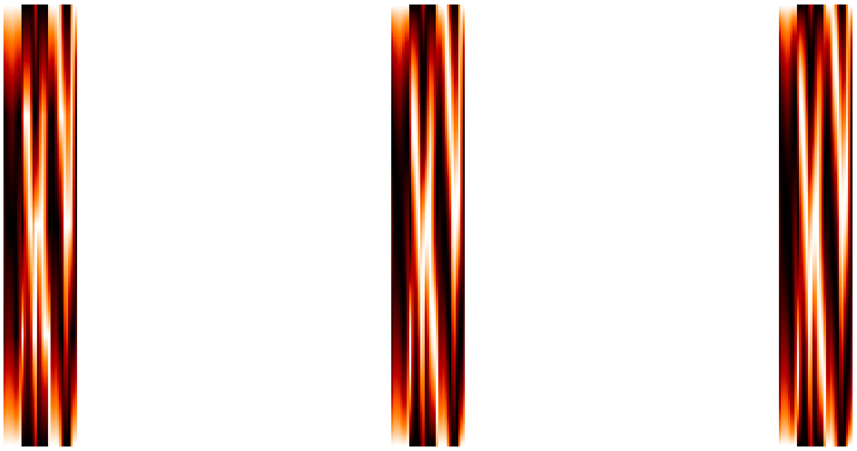
(a) Linear (b) Quadratic (c) Cubic

Figure 6.1: Signal plots after applying interpolation

6.2 Visual Interpretation

While this data representation is ready to be used for the desired purpose, one more modification will be made in order to make the image more intuitively understandable. This is to apply a color-map to the grayscale image in order to better interpret not only signal frequency, but intensity as well. For this purpose, we will use Matplotlib's $gist_heat$ color-map, as it is sequential and the color range feels natural with regards to variable intensity. Using the first 32 time steps of subject 'S001' of the physio dataset for trial 4 gives us the images shown in figure 6.2. Note that this is a 'rest' event.

A key point to keep in mind is that the width of an image directly correlates to a length in time. Modifying the width is comparable to modifying the reactivity of the system as a longer window means it will require more time-steps following an input for the system to build the signal image. It is in this way that we can evaluate the ability of the system to react quickly or slowly to an event. Applying this to our previous example with a width of 32 means that the signal images represents a time length of 0.2 seconds.



(a) Linear (b) Quadratic (c) Cubic

Figure 6.2: Signal images of start of S001 trial

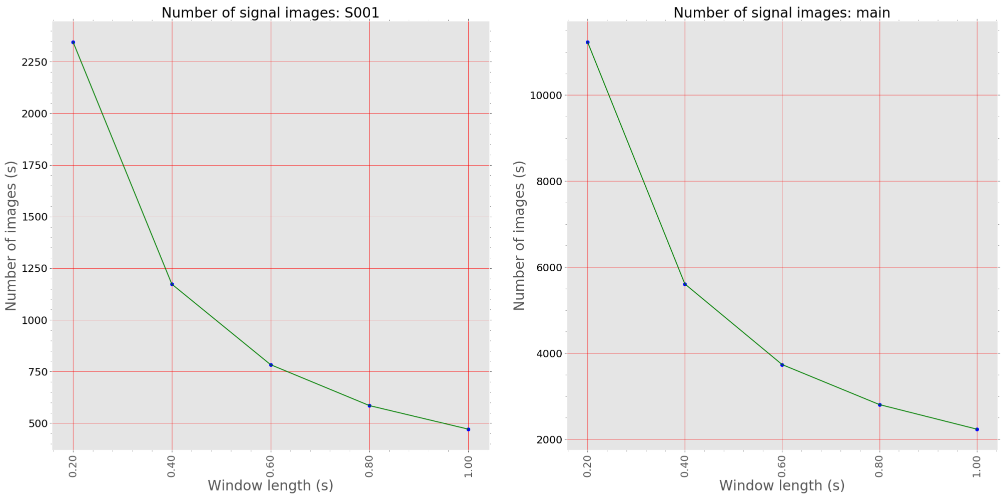
If each column of the image is effectively a time-step, then moving laterally across an image is effectively evaluating the signal in the frequency domain, as was discussed previously in section 2.3. It can be expected a shorter window would prove more difficult for a model to correctly classify due to not having enough samples while a longer window may present a challenge as it may contain multiple events or parts of signals that the classifier attributes to certain types of events.

6.3 Signal Image Labeling

After an image has been constructed, it is necessary to label each image according to an event: 'rest', 'left' or 'right'. It is best to take a step back and recall that a signal image is comprised of multiple signals where each sample is labeled as a singular event, leaving open the possibility that a window could contain multiple events. While not necessarily an issue, as we could use a multi-label classifier, this is not the way we'd want to handle this case for a control system. It wouldn't make much sense for a window to be both a 'rest' and a 'left' input. Instead, a window is labeled as the most common label that appears in a window. Compare this method of labeling a window with the other option of labeling a window as the same event as the last sample that appears in the window. While the latter option might feasibly offer better reactivity to the onset of an event, the decided upon approach should prove more suited to accurate classification as an image is more representative of the event.

6.4 Dataset Construction

Actually applying the signal image generation algorithm to the signal data requires deciding on two parameters: window length and window overlap. The first, length, is simply the amount of time the signal image represents (or the number of columns in a given image). The second, overlap, is a value, less than 1, that defines how many time-steps are shared between two sequential signal images. In the case of an overlap of 1, the signal images would be capturing the same exact time-steps. An interesting point to note is in the case of a negative overlap. Since a positive overlap can be thought of as a percent overlap, a negative value effectively corresponds to a percent separation: setting the overlap to below 0 adds a buffer in between two signal images.



(a) Subject 'S001'

(b) Main

Figure 6.3: Dataset size

Ideally, a new window would be generated for each time step. It is important to keep in mind the use case being explored: a control system. Not only would such an approach be resource intensive when applied to the actual task, beyond training the model, but it is overkill. Furthermore, by parameterizing overlap, the ideal case can still be achieved simply by increasing the overlap to just below 1, and, as such, allows for more fine-grained control when generating signal images either for the task of generating the training dataset or building the images for real-time prediction.

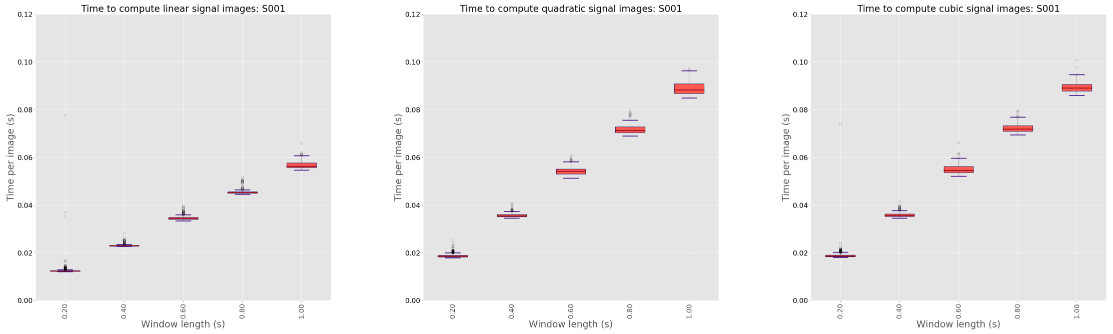
When generating the training datasets for each subject for both the Physio trials and the manually recorded trials, 5 different window lengths were chosen: 0.2 seconds, 0.4 seconds, 0.6 seconds, 0.8 seconds, and 1 second. The overlap for each of these window lengths was chosen to be 20%. Keep in mind that this is a percent overlap, not a direct time, such as is the case for the window length. Thus, this overlap is more pronounced in the case of the 1 second windows rather than for the 0.2 second windows. It is also the overlap parameter which more directly affects the size of the dataset generated for a trial recording.

Consider the actual dataset generated for the Physio trials. Between subjects, each of the trials are more or less about the same length, and, as such, examining subject 'S001' proves to be fairly representative of the dataset generated for each subject. Figure 6.3a shows the number of images generated per window. As three signal images are generated for each window, one for each interpolation type (linear, quadratic, and cubic), the actual number of windows per trial is $\frac{1}{3}$ the amount shown in the figure. However, as this is a consistent factor across all window lengths and all trials, the relation still holds between window length and the size of the generated dataset.

Applied to the recorded dataset, the results are slightly different, though follow the expected pattern. Figures 6.3b shows the number of images generated per window for the 'main' configuration outlined in section 5.2.4. While the actual number of images differs, the plot follows the same general trend of the number of images per window lengths as seen with the Physio dataset.

6.5 Timing Analysis

On top of the delay required to build a signal image that arises due to the need to aggregate the time-steps, it also takes time to perform the computations required to transforms the signal values into a signal image. This can be analyzed by again examining the characteristics of subject 'S001' from the Physio dataset as there are no peculiarities, at this point, that might arise due to different data sources or different subjects. The time per window length for different interpolation types is shown in figure 6.4.



(a) Linear

(b) Quadratic

(c) Cubic

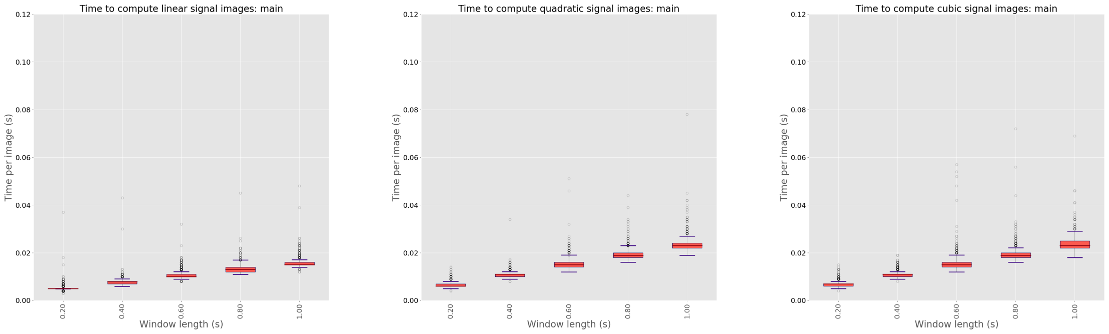
Figure 6.4: Time to generate signal images for subject 'S001'

As can be reasonably expected, the amount of time per image increases with the dimensionality of the interpolation function. Though, while the increase in amount of time seems rather substantial when going from linear to quadratic, this increase is less pronounced when going from quadratic to cubic. While a linear interpolation function would likely be less useful in terms of capturing meaningful data for a classifier to train on, the payoff comes in the form of a fast computation time when performing real-time prediction, but this same benefit does not hold for quadratic. Following the same line of thinking that a higher-order interpolation function would capture more information regarding the underlying signal, there does not appear to be a timing benefit of using a lower complexity interpolation function with regards to quadratic versus cubic.

Applying the same timing analysis to the manual recorded data yields just about a similar trend, shown in figure 6.5. The difference arises due to the different sample rates of the BCI2000 versus the Ganglion board. The BCI2000 system samples the signal at a rate of 160 Hz while the Ganglion reportedly samples the signal at a rate of 200 Hz. This means that the same window length generates a different sized images as the window length of a signal image is defined by time, not number of samples. This holds true for all of the manually recorded trials.

An interesting note is with the distribution of the time to generate the images. For the Physio dataset, the data was recorded in a very clean fashion. No samples were dropped and all windows contained the same number of samples, yielding the same sized image for each window length. The Ganglion has a tendency to drop samples, and sample times were recorded based on wall-clock timings as measured by the system. While there is a general trend that a longer window correlates to an increase in the number of samples, the actual number of samples per window length is not consistent for any given subject or trial.

A final point of note here is with regards to computational power. Table 6.1 shows the specifications of the system used to perform the computations. However, the calculations were performed using Python on a single process with no GPU acceleration: it is important to not draw a greater conclusion from the timings than is truly captured



(a) Linear

(b) Quadratic

(c) Cubic

Figure 6.5: Time to generate signal images for subject 'main'

by the plots shown in figures 6.4 and 6.5. The system itself is more powerful than a typical end-user system, but the implementation was written in such a way as to not effectively use those resources. The timings measured should be taken as a relative comparison on the amount of time taken per interpolation type rather than an absolute value for a total delay. It would not be accurate, for example, to say that a total delay for generating a signal image with a window of 0.2 seconds and a linear interpolation is about 0.24 seconds, despite this being about the sum of the time to accumulate the signal time-steps and to perform the computation for subject 'S001'.

Part	Specification
CPU	2700x 8 core/16 thread 3.7 GHz
Memory	64 Gbs 3000 MHZ DDR4
Storage	2 Tb SSD
GPU 0	1080 TI 11 Gb
GPU 1	1660 6 Gb
GPU 2	2080 8 Gb

Table 6.1: System Specifications

Chapter 7

Event Classification

The last phase in the research was to train a model that is able to quickly and accurately interpret inputs for the BCI system to use as a control input. That is, when fed a signal image that is labeled as either ’rest’, ’right’, or ’left’, it should be able to interpret that image as the appropriate input without a noticeable time delay.

Many different types of models have been used for motor-imagery classification. From logistic regression to recurrent neural networks, these models span the gamut in terms of understandability and interpretability. As discussed in section 3, there has been substantial recent progress in understanding neural networks, particularly in the domain of computer vision where convolutional neural networks (CNNs) currently reign supreme. As the data representation transforms EEG signals into images, the model of choice for the classifier was a CNN. Where typical machine learning research projects dealing with neural networks often explore different model architectures, the architecture for this research was explicitly chosen to be small and simple in order to evaluate the baseline efficacy of the data representation rather than how to construct a model particularly suited to the task of motor-imagery classification. Along with the architecture shown in figure 7.1, each convolutional layer uses a ReLu activation function in order to introduce non-linearity into the model layers. The final, dense layer uses a softmax activation, as this is a 3-class problem of classifying between ’rest’, ’left’, and ’right’.

7.1 Model Architecture

The input to the network is expected to be a 3-channel image that is $224\,x\,224$. Recall that the width of the signal image is not a static size; It is dependent on the number of time steps captured by the signal image. One option to deal with this disparity is to train a different model for each image width and then create a hierarchical model where each model operates on a different time scale. It may seem that this would likely be an effective approach, but that further assumes a single window size contains always the same number of samples across different trials. This may hold true for the Physio dataset, but we've already seen that this assumption is broken with fair regularity for the manually recorded data

The other option is to simply resize each image to a predetermined width and height – in this case, $224x224$. This presents an interesting change to each image as resizing an image to a different width fundamentally changes the information captured by a signal image in two profound ways: directly resizing an image effectively applies a secondary interpolation to a signal image and each time step is effectively scaled in the frequency domain.

The effects of these changes are best seen through example. Consider the images shown in figure 7.2, which show the same signal images from figure 6.2 after having been resized. The additional interpolation can most easily be seen towards the center of the linear image. In the original image, the black cleft is more pronounced, whereas after having been resized, this same cleft has an additional horizontal component that is not present in either the quadratic or cubic resized images despite a similar cleft being present in the originals.

The more interesting change is the second alteration: the effective scaling in the frequency domain. This is of particular importance due to the fact that traditional feature extraction techniques tend to operate on the frequency domain. For example, as discussed in section 4, motor imagery can be detected based on mu-rhythm suppression. After scaling, a model can be expected to interpret each image as being on a comparable

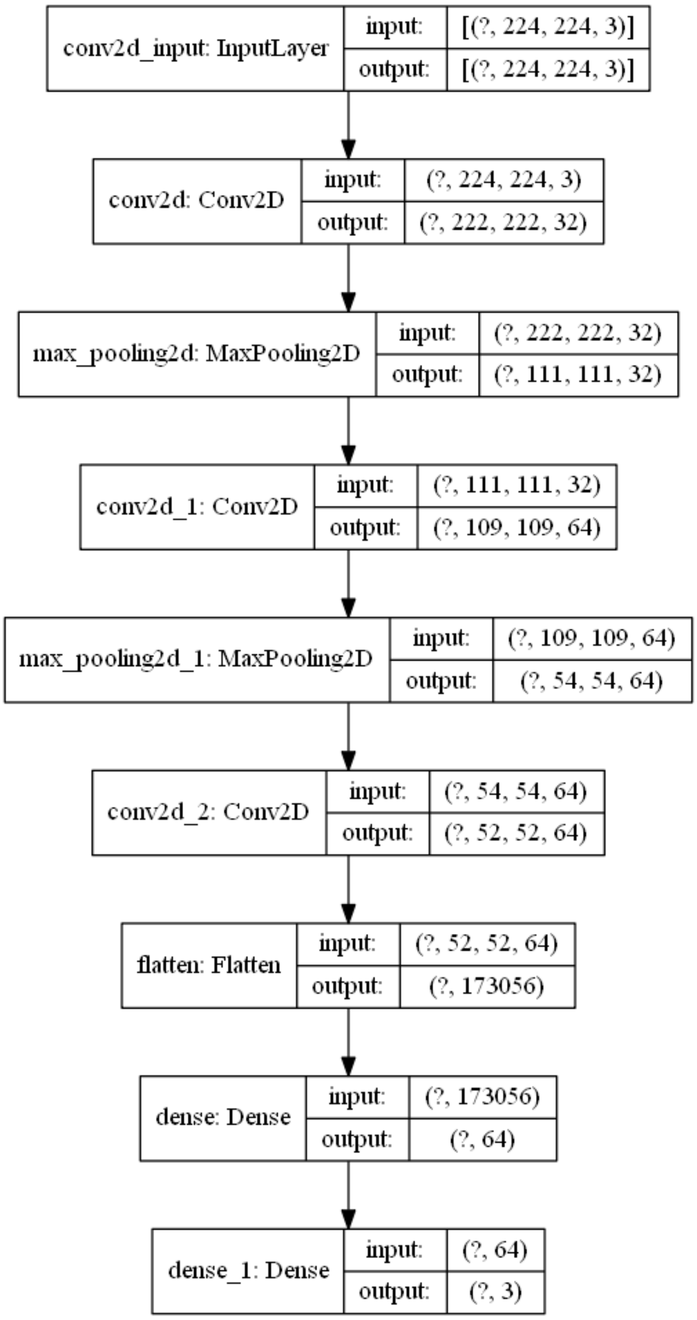
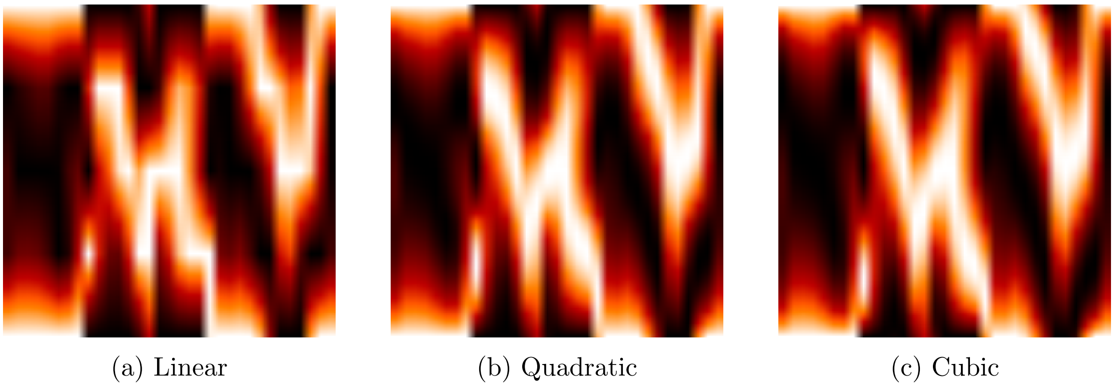


Figure 7.1: Convolutional Neural Network Architecture



(a) Linear (b) Quadratic (c) Cubic

Figure 7.2: Resized Signal Images

time length, even if the original images are of different time lengths. Thus, the frequency represented by each image, which can be visually seen as a horizontal change from black to white, is fundamentally altered after resizing the image.

7.2 Training

As with building the model, when training the model, the training parameters were chosen to prioritize simplicity and interpretability. There are only 5 main parameters with regards to the model that were configured during the training step: maximum number of epochs, learning rate, batch size, optimizer technique, and loss function. A further four parameters were specified that dealt with the data that was fed to the model during training: interpolation types, data source, subject name, and window lengths. In all, the only one that was varied across trials was the window lengths. All others were set to the default values shown in table 7.1. These were found to work passably well, and since they were held constant for each training set, each model was able to be directly compared to each other on the basis of those factors which were altered. Namely, the data representation and the effect of window sizes on the ability of the classifier to accurately predict events.

The three data parameters control the data that is fed to the classifier during training. Together, the data source and subject name uniquely identify an actual set of trials

Parameter	Default value
Maximum number of epochs	200
Learning Rate	$1E-5$
Batch size	16
Optimizer	Adam
Loss function	Sparse categorical accuracy
Interpolation types	Linear, quadratic, and cubic

Table 7.1: Model Training Parameters

performed by an individual, either from the Physio dataset or the manually recorded data. The window lengths specify a list of different sized signal images generated from those trials. This allows for training a model to be specific to an individual as it is generally expected that different people's brainwaves will appear differently even given the same set of events. While the model's ability to perform classification on groups of individuals was slightly explored during the course of the research project, it was deemed out-of-scope and left for future work.

The final considerations are with regards to the data: both the balance of the dataset and how the data was split. From a traditional view of the oddball paradigm, the events of interest occur much less often than the resting state. This inherently introduces a class imbalance that is heavily skewed towards the resting state. However, due to the actual procedure used for performing the recording sessions, coupled with the manner in which the dataset was constructed, this class imbalance is much less pronounced than is commonly expected. Table 7.2 shows the distribution of classes in the dataset for the Physio recordings for subject 'S001', and this distribution holds fairly consistently across all subjects for all trials for the Physio data.

Window Length (s)	T0	T1	T2
0.20	1188	1152	1158
0.40	591	579	579
0.60	396	387	384
0.80	279	306	306
1.00	237	234	228

Table 7.2: Physio Dataset Class Distributions: 'S001'

For the manually recorded data, the dataset is similarly balanced, as can be seen from table 7.3, which lists the class distribution for the 'main' set of trials. This is largely due to the procedural change where there was no guarantee of non-repeated events and no rest event was inserted between any two action events. From this perspective, the 'rest' event is effectively not considered the common stimulus. Rather, it is just another class for the classifier to learn to distinguish.

Window Length (s)	Left	Rest	Right
0.20	3594	3714	3927
0.40	1803	1845	1926
0.60	1197	1242	1296
0.80	897	924	984
1.00	717	738	780

Table 7.3: Manually Recorded Dataset Class Distributions: 'main'

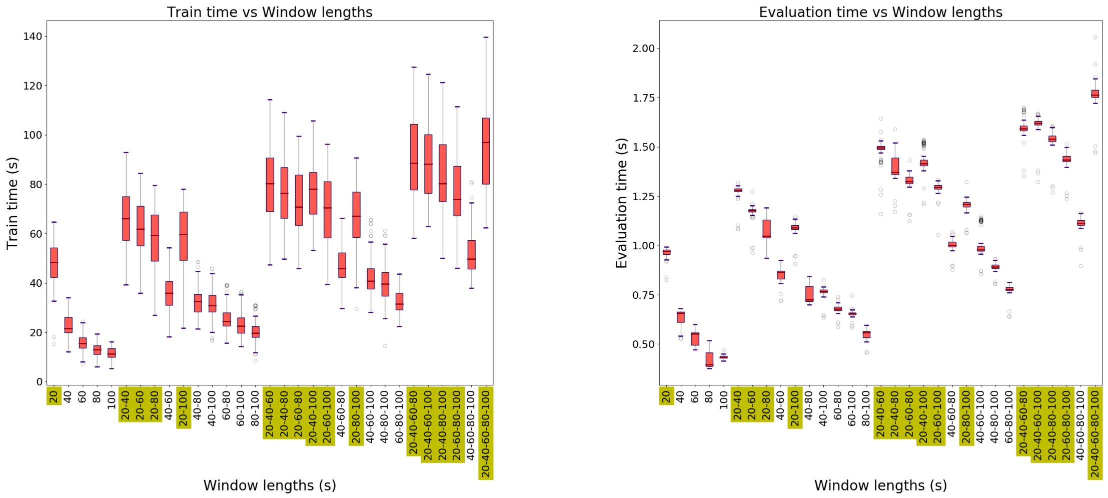
Finally, prior to being fed into the classifier for training, the dataset, built by pulling the images based on the data parameters, was split into a training, validation, and test set at a ratio of 60 : 15 : 25. The training and validation sets were used during training while the test set was held out for evaluating the model's ability to generalize to unseen data. Furthermore, the performance on the validation set was used as the early stopping criteria when training the model. If the validation loss did not improve over the previous three epochs, then the training was stopped and the model weights were restored to the weights of the model with the lowest validation loss.

7.3 Evaluation

Structuring the classification phase of the experiment in this way allows for the system's performance to be measured both in terms of accuracy and timing. The first is done by simply comparing the expected result versus the result of the classification based on the user's input. The timing analysis was done by analyzing the model's ability to accurately predict an event based on different combinations of window lengths and looking for trends across these different combinations.

7.3.1 Effect of Window Lengths

The Physio dataset contains a high number of subjects, allowing for the evaluation to consider not only how the system reacts to a particular individual, but how the process itself generalizes between individuals. Figure 7.3 shows the metrics of interest for gauging how effective the data representation is for creating a motor-imagery classifier. While it is expected, an important trend shows itself in the train time and evaluation time plots. For both of them, the time decreases as the number of images fed during training decreases ('0.20' has the most number of images and '1.00' has the fewest number of images). As different window sizes are mixed and matches, the models have an increasingly hard time learning the decision boundaries between the various classes, as can be seen in that the variance of train time increases, particularly when the spread between window sizes is larger ('0.20' with '1.00' versus '0.80' with '1.00').



(a) Train time

(b) Evaluation time

Figure 7.3: Physio Model Timings

The interesting performance of the model can be seen most clearly in the test accuracy plot, shown in figure 7.4. For all window combinations, merely having the '0.20' window length in the dataset severely impacted the overall ability of the different models to learn that particular dataset configuration. These trials are marked by the yellow background in the x-label. However, this is not a constant factor across all subjects.

For some subjects, the model is still able to perform rather decently when including the shortest window length, and even re-training the model using only '0.20' as the window length, for the same subject, yielded a large variance in the test accuracy. For instance, subject 'S001' generally achieved about 0.77% accuracy when training solely on the '0.20' signal images. Merely rerunning the training script with a different initial weights and random seed could cause the test accuracy to range between 54% to 91%.

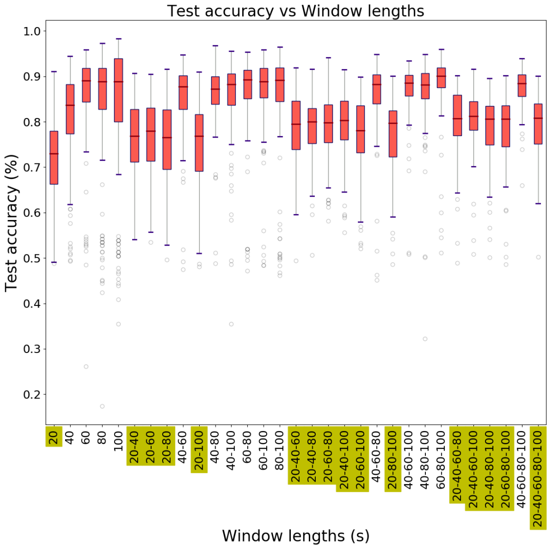


Figure 7.4: Physio Test Accuracy

Keeping in mind that EEG signals are characterized by a particularly low SNR, this large variance in the test accuracy can likely be attributed to a large number of local minima and maxima in the search space. This is somewhat backed by the fact that the batch size was kept rather small, only 16, due to memory constraints.

The true support for this theory of a large number of local minima and maxima is best supported by the models trained on the 'disconnected' subjects. It was expected that the model would perform poorly on these datasets as the data was effectively random, though this was found to not be true for either the 'disconnected' subjects nor the 'random' subject. Table 7.4 shows the accuracy achieved by the models for each window length. In some cases, it even appears that the 'disconnected' subjects outperform the models trained on the Physio dataset subjects, though retraining the models several times shows a similarly high variance in the test accuracy across training sessions.

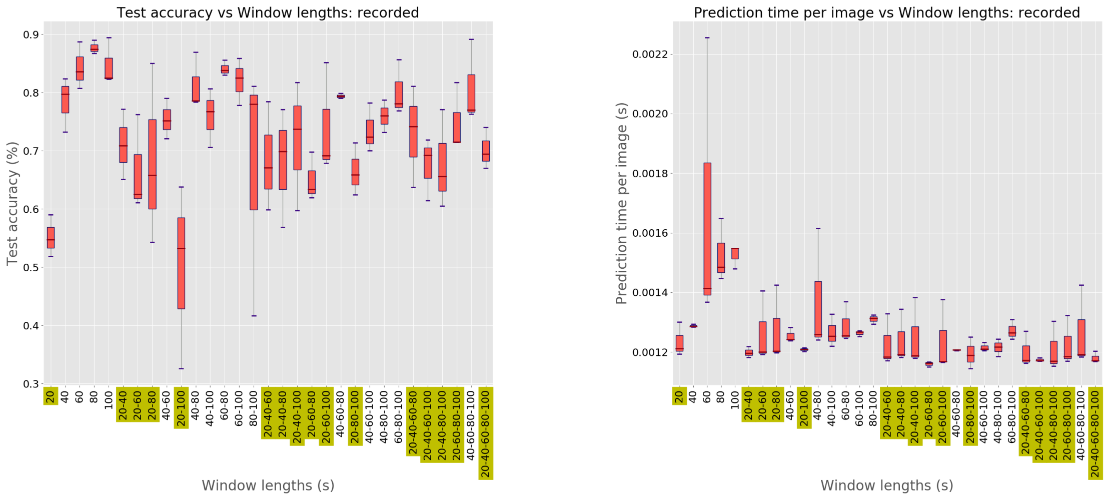
It is likely that there is some latent, environmental noise that occurred consistently when a new prompt was displayed, or the mere act of generating a new stimulus caused a change in the level of computation being performed by the system at a given point in time. This can likely be remedied slightly by amending the recording setup, as the system that was generating the prompts was the same system that was recording the samples streamed from the Ganglion board.

Subject	0.20	0.40	0.60	0.80	1.00
disconnected_01	83.45	92.17	88.77	92.20	88.39
disconnected_03	78.59	82.44	85.56	86.73	91.37
disconnected_05	51.58	82.41	84.15	85.33	82.50
main	59.99	79.76	83.62	87.46	82.29
random	54.75	73.27	80.73	89.03	89.45

Table 7.4: Recorded Accuracy per Window Lengths

It would seem that the most important takeaway from the models' performance on the 'disconnected' datasets is with regards to data acquisition and the effect of ambient noise in the recording environment. Each mean model accuracy, shown in figure 7.5a, was more accurate than random for all window lengths. Typically, it was much better than random performance. While this plot includes the 'main' and 'random' datasets as well as the 'disconnected' dataset, it suggests that accuracy, by itself, is not a good metric to use when evaluating the efficacy of such a system. Noise and random actions are so prevalent in the recorded data that there inevitably exists some local minima or maxima that is able to fairly accurately classify the limited data presented to the model. This effect is even greater when less data is presented to the model during training, as suggested by the fact that the 'disconnected_01' model outperformed the 'disconnected_03' model, which, in turn, outperformed the 'disconnected_05' model.

Despite this, the model trained on the 'main' dataset performed more as expected and showed a slightly reduced variance across training sessions, though nothing of verifiable statistical significance. The more expected behavior is noted in the fact that, in general, the model was able to achieve increasingly accurate performance as the window size



(a) Prediction Timings

(b) Prediction Timings

Figure 7.5: Recorded Model Metrics

grew longer. However, as with the 'disconnected' data, mixing windows with a larger size difference caused the model to take a performance hit, suggesting the importance of frequency in classifying motor-imagery events. Recall that resizing the image caused a scaling in the frequency domain. Thus, if two images, initially of different sizes, are resized to the same width, the model was expected to interpret the images to be on on a similar time scale. Thus, two images of large initial size difference would offer two different representations of the event to the model in terms of frequency, making it harder for the model to correctly learn to distinguish between different events.

When using the 'main' model as a live classifier to predict new inputs from a user, the model maintained a similar level of performance, suggesting that the models trained on the random and disconnected datasets were able to find some feature, noise or otherwise, that happened to arise due to the recording configuration or high SNR.

The final consideration is further along this idea of using the trained model as a live classifier. The classifier is meant to be placed in a BCI system and so the actual time to perform a prediction, shown in figure 7.5b, is of particular concern. In general, the model seems to be able to react quickly enough as a basic input system as the time to classify the signal image is negligible compared to the time to aggregate the necessary number of samples to build the image.

Chapter 8

Conclusion

Overall, the models were able learn to classify the signal images with a reasonable level of success, especially when considering that little to no expert domain knowledge was required to build and train the models. However, it is likely there is some latent feature captured by the data representation that is making the models seem to perform better than they should be performing.

The grand takeaway from this research is that BCI systems offer great promise and significant strides are being made at every level of the community. At the current point in time, especially for low-cost solutions, the technology and understanding of the brain simply is not sufficient or wide-spread enough for use in a real-world and real-time solution. Just as was the case a decade previously in the domain of computer vision, neural networks, and their unstructured approach to classification, do not seem to be the best approach for building a BCI. Instead, feature extraction, guided by experts, remain the most effective solution for such a system. The best use for representing EEG signals as signal images is not for the task of motor-imagery classification. Their strength shows in allowing researchers to visually inspect a set of EEG signals in order to visually locate those features which are best able to distinguish between sets of classes.

8.1 Data Acquisition

Building out a BCI system is already a difficult problem, made more so by the difficulties in performing EEG recording sessions. Additionally, any such recording system has a high burden placed on it, both computationally and with regards to expected performance. Samples are often recorded in excess of 150 Hz, and there is a constant push for higher and higher sampling rates in order to reduce potential aliasing. The Nyquist sampling rate is merely a theoretical lower bound only when considering the pure signal itself. For example, most EEG signals of interest, such as mu-rhythms, have a frequency of about 10 to 15 Hz, meaning we need a sampling rate of at least 20 to 30 Hz. However, as was seen with the Ganglion board, samples can often be dropped for a multitude of reasons or corrupted by environmental noise.

The headsets too are a sticking point in the process. Dry electrodes, which offer the desired potential to not require preparation of the skin as is the case with wet electrodes, are uncomfortable for use in elongated recording sessions. Even wearing the headset for just 20 minutes starts to become uncomfortable and distracting. For a low-cost solution, the open-source community has come a long way in the development of EEG boards, but there is still significant work that must be done before the hardware is at a point where it can reasonable collect, filter, and process samples for use in a BCI and to develop means of collecting data that does not become prohibitive after such a short time-frame.

8.2 Data Representation

The explored method of representing the data, in the form of signal images, was found to be an interesting hybrid between the theoretical neuroscience and the unstructured data preferred by neural networks. The effective strength of the representation seems to be heavily derived from the interpolation techniques which attempt to recreate the electromagnetic field given a set of discrete points. This could be a boon to the community as this way of viewing data could provide insight into solving the forward and inverse problems as well as allowing for faster exploratory data analysis. As a standalone approach for a BCI system and for use as input as a control system, signal images are somewhat lacking in their capabilities. They require a large time delay just to have enough samples to build a signal image, and this does not even account for the time it takes to compute the image once all the signals have been measured. Signal images may be useful in terms of learning the neuroscience behind a BCI system. However, they are not suited to the task of a BCI, or they require significant work before they can effectively be applied to create such a system.

8.2.1 Cross-Domain Understanding

Likely the key lesson derived from this research is not with regards to the performance of the model on the data and how it was fed to a classifier. Looking at the data in this form offered a method of representation that was able to concisely and effectively convey the information that was captured in the signals. From a philosophical perspective, the representation closely resembles a spectogram with the key differentiating factor being the various parameters that go into the actual creation of each signal image. The effects of these parameters can be visualized in real-time by altering the parameters at run-time. This capability offers the ability to create a visualization platform that allows a researcher to quickly search through the parameter space in order to find a configuration which, visually, is able to show a clear distinguish between the various classes captured during the data collection phase.

One caveat to note is that I have a background in domains more closely related to signal processing, including electrical engineering and computer vision. As such, this method of data representation could arguable be playing to my strengths in that they transform the data into a form that I am naturally more inclined to understand. The point here is that, just like any other data representation, one of the most important features is the audiences' understanding of the form in which the data is presented. While this representation seems to be inherently intuitive due to the design decisions which seek to mirror gradients and differences in the signals into their respective pixels in the signal images, further inquiry on the matter should be conducted before any definitive claims can be made.

8.3 Classification

The actual models trained on the signal images performed better than expected. In fact, they performed too well given that this level of performance was able to be achieved even when faced with known bad data. One of the goals of the data representation was that it would hold true to the theoretical underpinnings of the neuroscience involved in motor-imagery classification. The fact that some models were able to achieve over 90% accuracy on random data suggests strongly that there is some undesired characteristic present in either the recorded data or was introduced during the creation of the signal images. Furthermore, in the domains where a BCI control system might be most useful or desired (video games, augmented reality, or system controls), the inherent lag in the system is too great when given other means of control, such as a keyboard and mouse. In video games, a delay of as small as 40 ms is often noticeable while the lowest effective time delay tested in this research was 200 ms. In fact, it is for this reason that old CRT monitors are so sought after for older console games, such as Super Smash Brothers. They have less of a frame delay, allowing for players to min-max their input combinations to a greater degree of finesse.

Chapter 9

Future Work

While the research performed proves promising for this form of data representation for motor imagery classification, the work herein lays the groundwork for future development. Significant work remains in order to viably use it as a means of input for an effective and real-time BCI system. Part of this works deals not with the software and algorithmic approach, but in improving the hardware and the data acquisition capabilities of the system. In general, the next steps can be broken down into two focus areas: hardware progression and algorithmic development. While both are important as to the development of BCI systems, only one step should be taken at a time in order to ensure a solid base for development. The following suggestions for development seek to first address the issue of data acquisition and then enhance the efficacy of the data representation for more specialized approaches.

Release *1.0* in the GitHub repository contains the code that was developed in support of the research for this thesis. The state of the code is such that it is capable mainly of performing the tasks described in the experiments chapters. More specifically with regards to data acquisition, it can automatically download, parse, and format the data from the PhysioNet dataset, and it can interface with the Ganglion board (through the OpenBCI GUI) to record data samples. Unfortunately, this method of communicating with the board does not allow for the full 200 Hz sample rate. The next phase of the research was focused on using the data (from either the PhysioNet dataset or the manually recorded dataset) and building the dataset of signal images to use for training. The final phase uses those images to train a Tensorflow classifier and reports various metrics that characterize the performance of the model.

9.1 Hardware Development

The hardware boards for performing EEG collection have come a long way in the past decade. However, cost is often directly tied to spatial resolution. For the task of motor-imagery, high spatial resolution is not as important a factor, but it is still a desired characteristic of any BCI or EEG system.

9.1.1 Custom EEG Board

Creating a custom board for data acquisition is meant to serve two purposes. The first is an attempt to reduce the cost and improve upon the spatial resolution. The second attempts to increase the control offered by creating the system from scratch. While not a trivial task, it is made somewhat easier by the increase of open-source projects in the space. The circuit schematic for the Ganglion is open-sourced by OpenBCI and has an active community aiding in its development. This is possible as the main business model of OpenBCI seems to not be focused with the intellectual property of the boards, but rather the convenience of having them build and provide the board with minimal oversight required of the hobbyist. Unfortunately, active work on the hardware of the board seems to have stalled.

9.1.2 Active Electrodes

Beyond creating a board that offers at least comparable initial performance to the Ganglion or the Cyton, the electrodes used for data acquisition could theoretically be greatly improved by using active electrodes versus the passive electrodes provided when purchasing a board or headset from OpenBCI. Where passive electrodes essentially act as simple probes measuring the electrical potential and sending this signal to the board for amplification, active electrodes perform a level of amplification at the point of collection. This effectively increases the signal-to-noise ratio of the data as the effect of ambient and environmental noise is less pronounced with respect to the signal after having traveled along the wire of the electrode to the board.

9.2 Algorithmic Development

Where the hardware aspect of development looks to record cleaner and more precise data more directly, algorithmic techniques can be explored which may be able to address issues ranging from data cleaning, exploration, classification, and testing.

9.2.1 Data Cleaning and Noise Isolation

A point that cannot be emphasized enough is there is no substitute for clean data. The age-old adage is once again proven correct: garbage in means garbage out. This holds particularly true in domains where the data is inherently dirty and difficult to work with. While ideally this data would be collected as cleanly as possible, post-processing is possible which is able to isolate the signal from the noise after it has been recorded. The work of this research took several steps to perform this data cleaning, including filtering out specific frequency ranges; However, the issue of identifying artifacts in the data was not addressed.

Normally, this task would be undertaken by a subject matter expert who would mark bad regions of the data stream for removal. Another approach is to consider that most EEG signals are non-Gaussian in nature, meaning that principal component analysis will likely not be an effective tool to separate the signals. Instead, it is possible to use independent component analysis to separate overlapping events, to remove line noise from the data [32], and to automatically detect artifacts that may be present due to muscle movement [12].

9.2.2 Time-sequence Classification

When constructing the images, the time-series nature of the problem was handled by concatenating multiple samples together in order to form a 2D image of a signal over time. A better approach may be to use an architecture that is particularly suited for timeseries based classification, such as a recurrent neural network. While it may not prove more effective to use this other type of architecture for classification of these signals due to the increased difficulties in training such a network, it would provide additional insight into the strengths and weaknesses of the data representation, particularly with respect to the effect of the interpolation function and its ability to recreate the electromagnetic fields at play.

9.2.3 Field Reconstruction

Taking a step back from the end result of a signal image, one of the strengths of the data representation is its ability to interpolate between discrete points in order to partially recreate the electromagnetic field produced by the brain. In the research conducted, only the $C3$, CZ, and $C4$ electrode positions were used. This allowed for the interpolation function to operate on the points as if they were on the same x-y coordinate plane due to the fact that these locations are next to each other laterally on the head. However, increasing the spatial resolution of the system breaks this lateral assumption. Remedying the situation only requires having the interpolation function operate on the points in 3D space rather than 2D space. The signal image then becomes more akin to a classic montage or band-plot representation employed by neuroscience domain experts. Feeding it to a classifier then either requires use of a recurrent convolutional neural network or altering the convolutional filter from a 2D filter to a 3D filter, both of which are readily supported by Tensorflow.

Bibliography

[1] E. Larson D. Engemann D. Strohmeier C. Brodbeck R. Goj M. Jas T. Brooks L. Parkkonen M. Hmlinen A. Gramfort, M. Luessi. The typical m/eeg workflow. https://mne.tools/stable/overview/cookbook.html, Dec 2019.

[2] Sarah N. Abdulkader, Ayman Atia, and Mostafa-Sami M. Mostafa. Brain computer interfacing: Applications and challenges. *Egyptian Informatics Journal*, 16(2):213 – 230, 2015.

[3] N. Al-Ghamdi, G. Al-Hudhud, M. Alzamel, and A. Al-Wabil. Trials and tribulations of bci control applications. In *2013 Science and Information Conference*, pages 212–217, 2013.

[4] Haider Hussein Alwasiti, Ishak Aris, and Adznan Bin Jantan. Brain computer interface design and applications: Challenges and future. In *World Applied Science Journal*, 2010.

[5] Rehab Ashari and Charles Anderson. Eeg subspace analysis and classification using principal angles for brain-computer interfaces. In *2014 IEEE Symposium on Computational Intelligence in Brain Computer Interfaces (CIBCI)*, pages 57–63, 12 2014.

[6] Alyssa Batula, Jesse Mark, Youngmoo Kim, and Hasan Ayaz. Comparison of brain activation during motor imagery and motor movement using fnirs. *Computational Intelligence and Neuroscience*, 2017:1–12, 05 2017.

[7] Hopp JL et al Britton JW, Frey LC. *Electroencephalography (EEG)*. American Epilepsy Society, 2016.

[8] A. Coenen, Edward Fine, and Oksana Zayachkivska. Adolf beck: A forgotten pioneer in electroencephalography. *Journal of the History of the Neurosciences*, 23:1–11, 04 2014.

[9] Anne Collins and Michael Frank. Within- and across-trial dynamics of human eeg reveal cooperative interplay between reinforcement learning and working memory. *Proceedings of the National Academy of Sciences*, 115, 02 2018.

[10] Jahanshah Fathi Miao Sun Fani Deligianni Guang-Zhong Yang Dan-Dan Zhang, Jian-Qing Zheng. Motor imagery classification based on rnns with spatiotemporal-energy feature extraction. In *THE UK-RAS NETWORK CONFERENCE ON ROBOTICS AND AUTONOMOUS SYSTEMS*, 11 2017.

[11] Life Science Data. Fundamentals of eeg recording. https://blog.biopac.com/fundamentals-of-eeg-recording/, Jul 2017.

[12] Irene Dowding, Stefan Haufe, and Michael Tangermann. Automatic classification of artifactual ica-components for artifact removal in eeg signals. *Behavioral and brain functions : BBF*, 7:30, 08 2011.

[13] Rafael Duarte. *Low cost Brain Computer Interface system for AR. Drone Control.* PhD thesis, Federal University of Santa Catarina, 06 2017.

[14] Bryn Farnswort. What is eeg (electroencephalography) and how does it work? https://imotions.com/blog/what-is-eeg/, Jul 2019.

[15] Elliott M. Forney. Electroencephalogram classification by forecasting with recurrent neural networks. Master's thesis, Colorado State University, 2011.

[16] Elliott M. Forney. Measuring vigilant attention: Predictive power of eeg derived measures on reaction time, subjective state and task performance. Master's thesis, Colorado State University, 2011.

[17] Jon Gear. Building a mind-controlled drone. https://gear.dev/building-a-mind-controlled-drone, Jun 2019.

[18] Ary Goldberger, Lus Amaral, Leon Glass, Jeffrey Hausdorff, Plamen Ivanov, Roger Mark, Joseph Mietus, George Moody, Chung-Kang Peng, and H. Stanley. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. *Circulation*, 101:E215–20, 07 2000.

[19] Alexandre Gramfort, Martin Luessi, Eric Larson, Denis Engemann, Daniel Strohmeier, Christian Brodbeck, Roman Goj, Mainak Jas, Teon Brooks, Lauri Parkkonen, and Matti Hmlinen. Meg and eeg data analysis with mne-python. *Frontiers in Neuroscience*, 7:267, 2013.

[20] Alexandre Gramfort, Martin Luessi, Eric Larson, Denis A. Engemann, Daniel Strohmeier, Christian Brodbeck, Lauri Parkkonen, and Matti S. Hmlinen. Mne software for processing meg and eeg data. *NeuroImage*, 86:446 – 460, 2014.

[21] Alex S. Greaves. Classification of eeg with recurrent neural networks. 2015.

[22] L.F. Haas. Hans berger (18731941), richard caton (18421926), and electroencephalography. *Journal of Neurology, Neurosurgery, and Psychiatry*, 74:9, 02 2003.

[23] Paul Holmes and Dave Collins. The pettlep approach to motor imagery: A functional equivalence model for sport psychologists. *J. Appl. Sport Psychol.*, 13, 10 2007.

[24] Iaki Iturrate, Luis Montesano, and Javier Minguez. Robot reinforcement learning using eeg-based reward signals. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4822–4829, 05 2010.

[25] Sim Kok Swee, Desmond Teck Kiang Kho, and Lim Zheng You. Eeg controlled wheelchair. *MATEC Web of Conferences*, 51:02011, 01 2016.

[26] Shiu Kumar, Alok Sharma, Kabir Mamun, and Tatsuhiko Tsunoda. A deep learning approach for motor imagery eeg signal classification. In *2016 3rd Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE)*, pages 34–39, 12 2016.

[27] Min-Ho Lee, O-Yeon Kwon, Yong-Jeong Kim, Hong-Kyung Kim, Young-Eun Lee, John Williamson, Siamac Fazli, and Seong-Whan Lee. Eeg dataset and openbmi toolbox for three bci paradigms: an investigation into bci illiteracy. *GigaScience*, 8(5), 01 2019.

[28] S.J. Luck. *An Introduction to the Event-Related Potential Technique.* A Bradford Book. MIT Press, 2014.

[29] T. Luo, Y. Fan, J. Lv, and C. Zhou. Deep reinforcement learning from error-related potentials via an eeg-based brain-computer interface. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 697–701, Dec 2018.

[30] Xuelin Ma, Shuang Qiu, Changde Du, Jiezhen Xing, and Huiguang He. Improving eeg-based motor imagery classification via spatial and temporal recurrent neural networks. In *Conference proceedings: ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, volume 2018, pages 1903–1906, 07 2018.

[31] Ikhtiyor Majidov and Taegkeun Whangbo. Efficient classification of motor imagery electroencephalography signals using deep learning methods. In *Sensors*, 2019.

[32] Scott Makeig, Anthony Bell, Tzyy-Ping Jung, and Terrence Sejnowski. Independent component analysis of electroencephalographic data. 8, 08 1996.

[33] Dennis J. McFarland and Jonathan R. Wolpaw. Braincomputer interface use is a skill that user and system acquire together. *PLOS Biology*, 16(7):1–4, 07 2018.

[34] D.J. McFarland and J.R. Wolpaw. Eeg-based braincomputer interfaces. *Current Opinion in Biomedical Engineering*, 4:194 – 200, 2017. Synthetic Biology and Biomedical Engineering / Neural Engineering.

[35] Hai-Jiang Meng, Yan-Ling Pi, Ke Liu, Na Cao, Yan-Qiu Wang, Yin Wu, and Jian Zhang. Differences between motor execution and motor imagery of grasping movements in the motor cortical excitatory circuit. *PeerJ*, 6, 08 2018.

[36] David Millet. The origins of eeg. http://www.bri.ucla.edu/nha/ishn/ab24-2002.htm, Jun 2002.

[37] I. A. Mirza, A. Tripathy, S. Chopra, M. D'Sa, K. Rajagopalan, A. D'Souza, and N. Sharma. Mind-controlled wheelchair using an eeg headset and arduino microcontroller. In *2015 International Conference on Technologies for Sustainable Development (ICTSD)*, pages 1–5, 2015.

[38] K.E. Misulis and T. Fakhoury. *Spehlmann's Evoked Potential Primer.* Butterworth-Heinemann, 2001.

[39] NeuroSky. Eeg: The ultimate guide. http://neurosky.com/biosensors/eeg-sensor/ultimate-guide-to-eeg/, April 2020.

[40] E. Niedermeyer and F.H.L. da Silva. *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*. LWW Doody's all reviewed collection. Lippincott Williams & Wilkins, 2005.

[41] S. S. R, J. Rabha, K. Y. Nagarjuna, D. Samanta, P. Mitra, and M. Sarma. Motor imagery eeg signal processing and classification using machine learning approach. In *2017 International Conference on New Trends in Computing Sciences (ICTCS)*, pages 61–66, Oct 2017.

[42] Giulio Ruffini, Ibanez-Soria David, Marta Castellano, Stephen Dunne, and Aureli Soria-Frisch. Eeg-driven rnn classification for prognosis of neurodegeneration in at-risk patients. In *Artificial Neural Network-Based Control Architecture: a Simultaneous Top-down and Bottom-up Approach to Autonomous Robot Navigation*, pages 306–313, 10 2016.

[43] Giulio Ruffini, David Ibañez, Marta Castellano, Laura Dubreuil, Jean-François Gagnon, Jacques Montplaisir, and Aureli Soria-Frisch. Deep learning with eeg spectrograms in rapid eye movement behavior disorder. *bioRxiv*, 2018.

[44] 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*, 51(6):1034–1043, June 2004.

[45] Felix Schirmann. The wondrous eyes of a new technology a history of the early electroencephalography (eeg) of psychopathy, delinquency, and immorality. *Frontiers in human neuroscience*, 8:232, 04 2014.

[46] Robin Schirrmeister, L Gemein, K Eggensperger, Frank Hutter, and Tonio Ball. Deep learning with convolutional neural networks for decoding and visualization of eeg pathology. In *IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, pages 1–7, 12 2017.

[47] Utkarsh Sinha and Kanthi Hegde. Mind controlled wheelchair. *IOSR Journal of Electrical and Electronics Engineering*, 9:19–28, 01 2016.

[48] Jagna Sobierajewicz, Sylwia Szarkiewicz, Anna Przekoracka-Krawczyk, Wojciech Jakowski, and Rob Lubbe. To what extent can motor imagery replace motor execution while learning a fine motor skill? *Advances in Cognitive Psychology*, 12, 12 2016.

[49] Abdulhamit Subasi and Ergun Erelebi. Classification of eeg signals using neural network and logistic regression. *Computer methods and programs in biomedicine*, 78:87–99, 06 2005.

[50] K. P. Thomas, N. Robinson, K. G. Smitha, and A. P. Vinod. Eeg-based discriminative features during hand movement execution and imagination. In *2018 15th*

International Conference on Control, Automation, Robotics and Vision (ICARCV), pages 883–888, Nov 2018.

[51] Brain Vision. The brief history of brain computer interfaces. http://www.brainvision.co.uk/blog/2014/04/the-brief-history-of-brain-computer-interfaces/, April 2014.

[52] M. Vourkas, S. Micheloyannis, and G. Papadourakis. Use of ann and hjorth parameters in mental-task discrimination. In *2000 First International Conference Advances in Medical Signal and Information Processing (IEE Conf. Publ. No. 476)*, pages 327–332, Sep. 2000.

[53] Dalin Zhang, Lina Yao, Xiang Zhang, Sen Wang, Weitong Chen, and Robert Boots. Association for the advancement of artificial intelligence. *Association for the Advancement of Artificial Intelligence*, 08 2017.

Appendices

Appendix A

Code and Dataset Repositories

Lists and briefly describes repositories containing code or datasets that may be useful when performing EEG research or attempting to build a brain-computer interface.

A.1 Github Repository for this research

- https://github.com/Adrang/SystemControl

Extensive code was written in order to support the efforts of this research. This GitHub repository contains the code in its current form as well as the git history. Release *1.0* is the code that was written prior to the conclusion and publication of this thesis.

A.2 OpenBCI

- https://github.com/OpenBCI

OpenBCI open-sources all the hardware and software they develop. This GitHub profile contains multiple repositories containing documentation, guides, and code that can be used to aid in building and using their products.

A.3 BCI Competition IV

- http://www.bbci.de/competition/iv/

The BCI competition was developed to address the issue of high-quality free EEG datasets. It aims to provide a way for researchers and hobyists to validate their signal processing techniques and improve on existing classification techniques. The last dataset was released in 2006 and was featured at NeurIPS (NIPS) 2008.

A.4 EEG Datasets

- https://github.com/meagmohit/EEG-Datasets

This repository contains links to multiple datasets for many different applications of EEG recordings, including motor-imagery classification.

Appendix B

Neuroscience Learning Resources

Various links that contain useful information for learning the neuroscience behind brain-controlled interfaces. The content of the earlier links tend to contain mainly theoretical explanations while the later links are more focused towards short guides, tutorials, and projects.

B.1 The typical M/EEG workflow

- https://mne.tools/stable/overview/cookbook.html

MNE is an open-source Python library for performing M/EEG data analysis. This page, specifically, details the common steps taken and covers the basic considerations that must be taken into account while performing such an analysis. Much of their guide is focused on use of their specific package, but the concepts can be readily applied outside of the framework as well.

B.2 EEG: The Ultimate Guide

- http://neurosky.com/biosensors/eeg-sensor/ultimate-guide-to-eeg/

Neurosky is another low-cost EEG option that is often used to build out BCI systems. This page details some of the background and basics regarding electroencephalography. In particular, it provides a brief overview of the history of the field and goes into some potential applications in which the technology can be applied.

B.3 What is EEG (Electroencephalography) and How Does it Work?

- https://imotions.com/blog/what-is-eeg/

IMotions is a purveyor of mid-range EEG headsets as well as resources aimed to help people learn what EEG is and how to use it. This page offers a brief overview of different brainwave patterns and how they can be interpreted from a neuroscience perspective.

B.4 Reading Minds with Deep Learning

- https://blog.floydhub.com/reading-minds-with-deep-learning
- https://www.kaggle.com/c/grasp-and-lift-eeg-detection/discussion/16479

Samuel Lynn-Evans' guide on using convolutional neural networks is fairly thorough, both in terms of explaining the neuroscience behind interpreting brain-wave patterns as well as on how to use the neural network to classify the signals. It uses the Grasp-and-lift dataset from Kaggle and attempts to detect which hand is performing the action.

B.5 Building a mind-controlled drone

- https://gear.dev/building-a-mind-controlled-drone
- https://github.com/jongear/mindcraft

Jon Gear's implementation of a drone shows how to use two different EEG headsets: the Cyton from OpenBCI and the Mindwave from Neurosky. The blog post walks through setting up a Parrot drone to be controlled by an EEG headset. It is written and described primarily in Javascript, but all the concepts are thoroughly explained. The repository also contains a set of slides and a video demonstrating how to get everything working.