CMPUT 275 Final Project Proposal

March 4, 2020

Project Title: EEG Visualizer
Group Members: Eddie Guo, Jason Kim

Description

In this project, we plan on visualizing live-time mock EEG data streamed over a local network (using Lab Streaming Layer, LSL) on the desktop. A UI will be implemented, and the user will be able to view a live graph of a single-channel EEG signal alongside a live Fourier transformed plot. If time permits, we will also implement a live bar chart of the relative strength of the band power of the received signals[1] (alpha, beta, delta, gamma, theta).

The main focus of this project will be to implement a fast Fourier transform (fft) and to plot live data. We also hope to add support for saving input, scrollable graphs, configurable colour themes, and multiple EEG channels.

Milestones

1. **By March 13:** Implement the fft algorithm. We plan on implementing the Cooley-Tukey fft algorithm. First, we will write the radix-2 decimation-in-time algorithm. Then, we will attempt the chirp Z-transform (aka Bluestein's fft algorithm). We expect that the latter fft algorithm will be mathematically challenging to understand and difficult to implement (but extremely cool to do!).

2. **By March 20:** Build the GUI. Here, we will use Matplotlib (and maybe Seaborn) to animate the graphs. All incoming data will be scaled to the appropriate time range. All data will be sent and received via LSL at 128 Hz. This is so that we can accurately plot within acceptable sampling errors.[2]

3. **By March 27:** Expand on the GUI (perhaps using PyQt5), perform unit tests on the fft algorithms, and refactor the code. We predict that the tasks from the previous week may bleed into this week due to little bugs in application logic. We will consider multi-threading to speed up our application.

Bonus Milestones (Time Permitting)

- Add a homepage which allows users to choose which plots they want (raw EEG data, fft plot, bar chart of band power).
- Add multiple EEG streams. We will stick to one EEG stream initially to prevent excessive busywork.
- Contigent on the previous bonus milestone, we will add buttons which allow the user to toggle individual EEG streams. We may also translate the fft algorithm to C++ code and call it using a wrapper in Python. We may alternatively use Cython to create a shared library.

[1] See here for more details on interpreting EEG data.
[2] See Nyquist-Shannon sampling theorem.