Bayesian Deep Learning for Electroencephalogram Signal Recognition

Ethan Davis
University of Washington

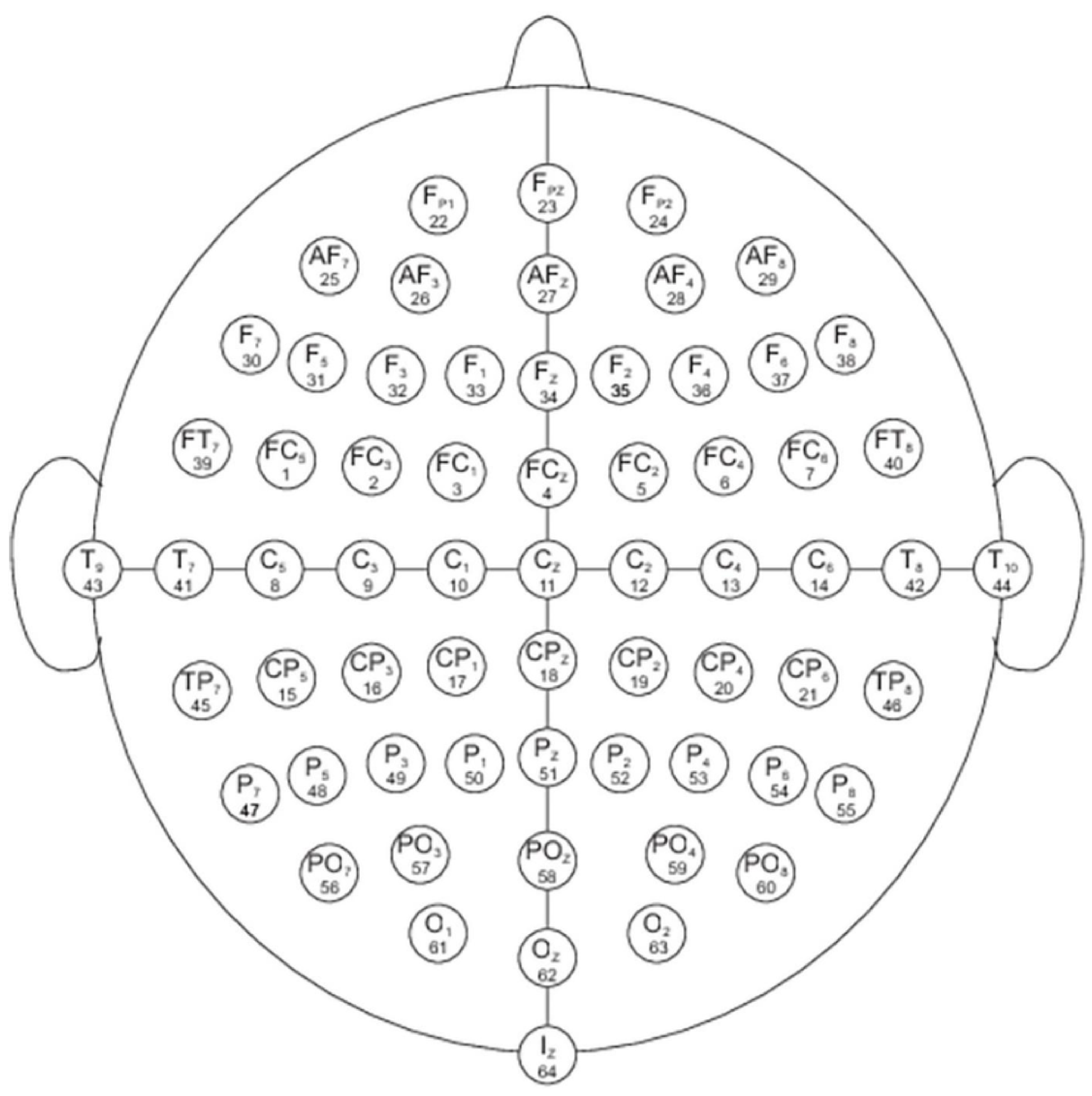
Introduction

Predictive uncertainty is critical for adoption of ML solutions used in high risk scenarios like autonomous driving, financial decisions, and medical treatment. Although deep learning has achieved significant success at modeling complex patterns in many areas, it is a black box that risks overfitting. Bayesian deep learning (BDL) offers a promising paradigm of ML applications that cannot silently fail at generalization without emitting a warning [1].

Motivation

The *Smart NeuroRehab Ecosystem* lab at the University of Washington aims to build electroencephalogram (EEG) brain computer interfaces (BCI) for assisted care in neurological rehabilitation. BCI devices that deliver measurable certainty with models are safer to administer and use for treatment. Other applications of BCI include controlling electronic devices for individuals with limited motor control, monitoring neurofeedback for anesthetics dosage during surgery, and directing robotics in various scenarios including disaster relief.

International EEG Electrode Locations



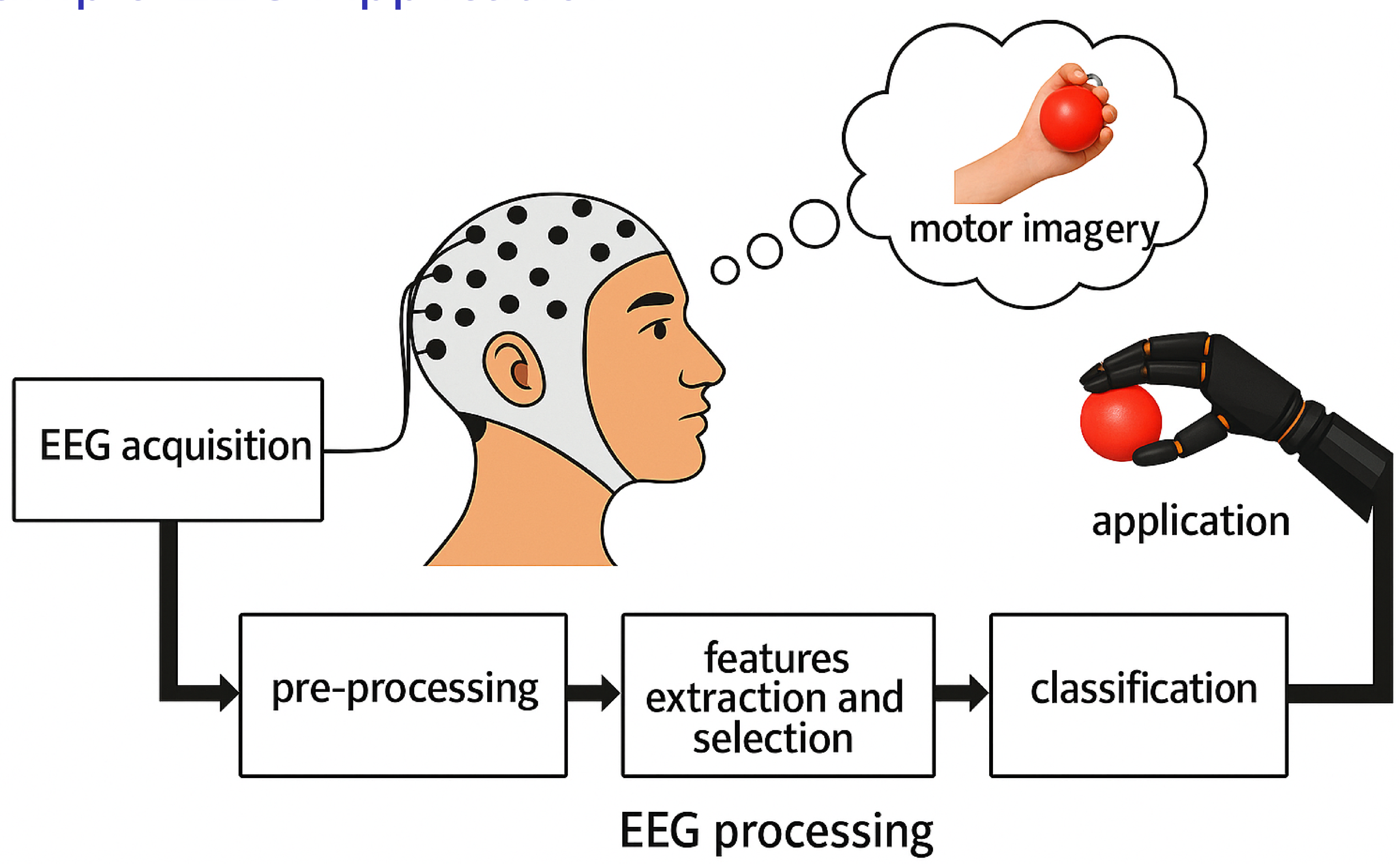
Data

EEG is electrical signals created by the brain that is transmitted between neurons. Motor EEG is signals created from physical activity and motor imagery (MI) EEG is created from imagining the performance of physical activity. EEG is notoriously noisy because it also records eye blinks, heart beats, and level of attention. I will use the MI EEG PhysioNet dataset [2].

Methods

1. **Prepare Data**: Raw EEG signals can be used as training and testing examples for Bayesian convolutional neural networks (BCNNs). Alternatives include time-frequency spectrogram images and covariance matrices between signal channels. We must tradeoff between speed and correctness of classification. Real-time classifiers are critical for MI EEG applications.
2. **Post-Test Only Control Group Design**: Perform random assignment to control and experimental groups. Train and test both groups 3 times with 10-fold cross-validation to create 30 data points each. Use a faster ResNet18, deeper ResNet36, or generalizable EEGNet backbone architecture.
3. **Hypothesis Tests**: Perform the bootstrap method to repeatedly sample the results and collect statistical means. By CLT, we can use student t-tests to compare both groups. The results will tell us whether one group performed better on average.

Example EEG Application



Key Metrics

- **Negative Log Likelihood (NLL)**: How well the predicted probability distribution assigns high likelihood to the ground truth labels [3].
- **Expected Calibration Error (ECE)**: The average difference between predicted confidence and empirical accuracy across probability bins [3].
- **Brier Score**: The mean squared error between predicted probabilities and one-hot encoded ground truth labels [3].
- **Mutual Information (MI) / Epistemic Uncertainty**: How much information about the model parameters is gained by observing the output, i.e. model uncertainty [3].
- **Inference Time / FLOPS**: Computational cost per prediction either in wall clock time or number of floating point operations [4].

Hypotheses

1. A BCNN has higher prediction correctness than a CNN as measured by corrected accuracy metrics like NLL, ECE, and Brier score that consider uncertainty.
2. A BCNN with custom NVIDIA CUDA kernels for Bayesian layers has higher FLOPS than a BCNN built entirely with general purpose technologies like Python, PyTorch, and Pyro.

Next Steps

Work that I have completed includes finding textbooks and literature to study information theory and BNNs [3]. I have identified possible computational bottlenecks that could be improved with custom NVIDIA CUDA kernels: repeated sampling from distributions, Monte Carlo forward pass, and Bayesian convolutional layers [1]. Ongoing work includes deciding the data format and model architecture. I still need to decide what layers to use for Bayesian inference. After building a prototype I can perform runtime profiling to determine any computational bottlenecks.

References

[1] L. V. Jospin *et al.*, "Hands-On Bayesian Neural Networks—A tutorial for deep learning users," , 2022.
[2] *EEG Motor Movement/Imagery Dataset V1.0.0*, 2009.
[3] K. P. Murphy, *Probabilistic Machine Learning*. 2023.
[4] W.-M. W. Hwu, D. B. Kirk, and I. E. Hajj PhD, *Programming massively parallel processors*. 2022.