

Harmful Brain Activity Classification using Deep Learning

Viraj Meruliya

viraj.meruliya@mail.mcgill.ca

Zilu Ma

mazilu95.math@gmail.com

Shreya Shukla

sshukla4@uci.edu

Soumya Ganguly

soumya123ganguly@gmail.com

Introduction

Electroencephalography (EEG) has long been a valuable tool in the diagnosis and monitoring of neurological disorders, including epilepsy. In critically ill hospital patients, the ability to rapidly and accurately detect seizures and harmful brain activity is of utmost importance for timely intervention and treatment. The advent of machine learning techniques offers promising avenues for automating the analysis of EEG data, paving the way for more efficient and precise neurocritical care. This summary outlines the efforts undertaken in a Kaggle competition aimed at developing a model for classifying such seizures, with the overarching goal of advancing medical research and patient care in neurology and related fields.

Methodology

We are given two sets of data as inputs: EEG as time-series numerical data and spectrograms as images for 1950 patients within some time frame. The targets are expert diagnostic votes on six classes of brain activities. After some pre-processing, we use around 17000 instances of data.

First, we select our features following standard techniques. Among 20 raw EEG features within some time frame, we choose 8 important features which represent EEG data detected from 8 different positions of the brain. We consider differences of neighboring EEG features, de-noise them, and take them as the training/inference features.

We prefer EEG data for several reasons. First, spectrograms can be transformed from EEG ([1],[2]). Also, the size of EEG data is significantly smaller than that of spectrograms, which leads to less memory use and faster training. Each epoch of training for EEG data finishes in around 40 seconds (depending on the model), as compared to 80 seconds for spectrograms.

For the deep learning architecture, we roughly follow that of EEGNet. We arrange 4 convolution layers with increasing kernel sizes, followed by 8-10 ResNet blocks with a fixed kernel size. Between the conv and ResNet layers, we use various techniques such as batch normalization, average/max pooling, and dropout, and we only use ReLU as the activation. As a major difference from EEGNet, we aggregate the output of ResNet with a GRU (gated recurrent unit) layer, which is efficient at dealing with time-series data. Finally we use a fully connected layer to map to logits of 6 classes, and the loss function is KL divergence. Our models have 0.2-0.5 million parameters. We also experiment by replacing the GRU layer with a LSTM (Long Short Term Memory) layer. We find that it gives similar validation loss as GRU. Since LSTM has more parameters than GRU, we use GRU for the other models.

Results/Summary

Upon running the experiments using different sets of hyperparameters, the best results we obtained are summarized in the table below:

Table 1: Best performance table for different models using EEG data

nn.GRU hidden size	num folds	Optimizer	learning rate	drop prob EEGNet	drop prob Resnet1D	Min Val loss
128	5	Adagrad	1.00E-03	0.1	0.1	0.5
128	5	Adadelta	1.00E-01	0.1	0.1	0.48
128	5	Adadelta	1.00E-01	0.1	0.1	0.5
256	5	AdamW	1.00E-03	0.1	0.2	0.5
128 (LSTM)	4	Adagrad	1.00E-03	0	0	0.5

Inference and Future Directions

There was a starter notebook provided at Kaggle that used 'Efficientnet' to classify the data. If we use the architecture of the starter notebook, then the minimum validation losses we get are 0.8426 (for spectrogram data), and 1.2149 (for EEG data, treated as images). We also tried with different basic models like Mobilenet2 (trained on imagenet), Yolo version 8, Untrained ResNet50V2 and all of them gave results in the same range (or slightly worse). So the most efficient model from above that we finally obtained after several trials gives significantly better results than the starter notebook. In future there is the scope of using a combination of few of these architectures and/or new architectures such as transformers to get better results.

References

- [1] Chris Deotte. How to make spectrogram from eeg. <https://www.kaggle.com/code/cdeotte/how-to-make-spectrogram-from-eeg>. Accessed: 2024-04-27.
- [2] Marcus C Ng, Jin Jing, and M Brandon Westover. A primer on eeg spectrograms. *Journal of Clinical Neurophysiology*, 39(3):177–183, 2022.