Google Summer of Code 2024 Final Project Report

Shreyas S
Department Of Biomedical Informatics, Emory University

August 2024

1 Introduction

This report provides a comprehensive overview of the project undertaken during the Google Summer of Code (GSoC) program with the Department of Biomedical Informatics at Emory University. The primary objective of this project was to develop an open-source foundational model for Electroencephalography (EEG) data. Our work aimed to enhance the capabilities and performance of the foundational model, making it more adaptable and efficient for EEG data processing.

2 Project Overview

The foundational EEG model development was built upon on the NeuroGPT [1] codebase, a robust foundation for processing and analyzing EEG data. The project encompassed several critical enhancements to this existing codebase, such as adapting data loaders, implementing model architecture modifications, addressing computational challenges, and optimizing evaluation processes. These enhancements were pivotal in enabling the model to handle EEG data more effectively, ensuring improved performance and efficiency.

3 Additions, Achievements, and Outcomes

3.1 Data Loaders and Model Architecture Adaptations

One of the fundamental aspects of this project was the adaptation of data loaders to handle EEG data more efficiently. EEG data, typically stored in `.edf` or `.fif` format, required conversion into a format compatible with the PyTorch-based NeuroGPT model. To address this, we developed a script that converts `.fif` files into `.pt` files, enabling integration of EEG data into the NeuroGPT framework. The script employed the MNE library for loading the data into MNE objects, converting it into NumPy arrays, and subsequently into PyTorch tensors.

Additionally, modifications were made to the model architecture to accommodate the unique characteristics of EEG data. Key changes included adjusting removing unnecessary channels, altering the dataloader parameters to optimize chunk length and overlap, and updating the encoder class to support the revised number of channels. These architectural changes were done to prepare the NeuroGPT for our preprocessed Temple University Hospital (TUH) EEG Corpus.

3.2 Integration of SafeTensors Support for Distributed Environments

During the course of our project, we also identified the need to support models saved in SafeTensors format, especially when training in distributed environments. The SafeTensors format is particularly useful for models trained across multiple GPUs, as it ensures efficient storage and loading of model states. We added SafeTensors support to our codebase, allowing models trained in distributed environments to be loaded and fine-tuned, thereby improving the flexibility and scalability of our system.

4 Adding DistilledGPT Architecture

Another significant addition to the project was the integration of the DistilledGPT architecture, a lightweight variant of GPT-2. The need for a more computationally efficient model became evident as we aimed to optimize training times and resource utilization without sacrificing model performance.

DistilledGPT is designed to offer the same foundational capabilities as GPT-2 but with fewer layers and parameters, making it particularly suited for environments with limited computational resources. This architecture allowed us to achieve faster training times, which is essential in scenarios where rapid iteration is required, such as in distributed computing environments.

The implementation involved incorporating DistilledGPT as an option within the model configuration. This required changes in both the model loading mechanisms and the training scripts to ensure compatibility with the reduced architecture.

4.1 Implementation of DeepLIFT for Model Interpretability

To improve the interpretability of our model's decisions, we implemented the DeepLIFT (Deep Learning Important FeaTures) technique. DeepLIFT allows us to attribute the output of the model to specific input features, in this case, EEG channels. This implementation provided valuable insights into which channels contributed most significantly to the model's predictions (spatial features), aiding in the understanding and debugging of the model. The following example plots were generated to visualize the importance of different channels:

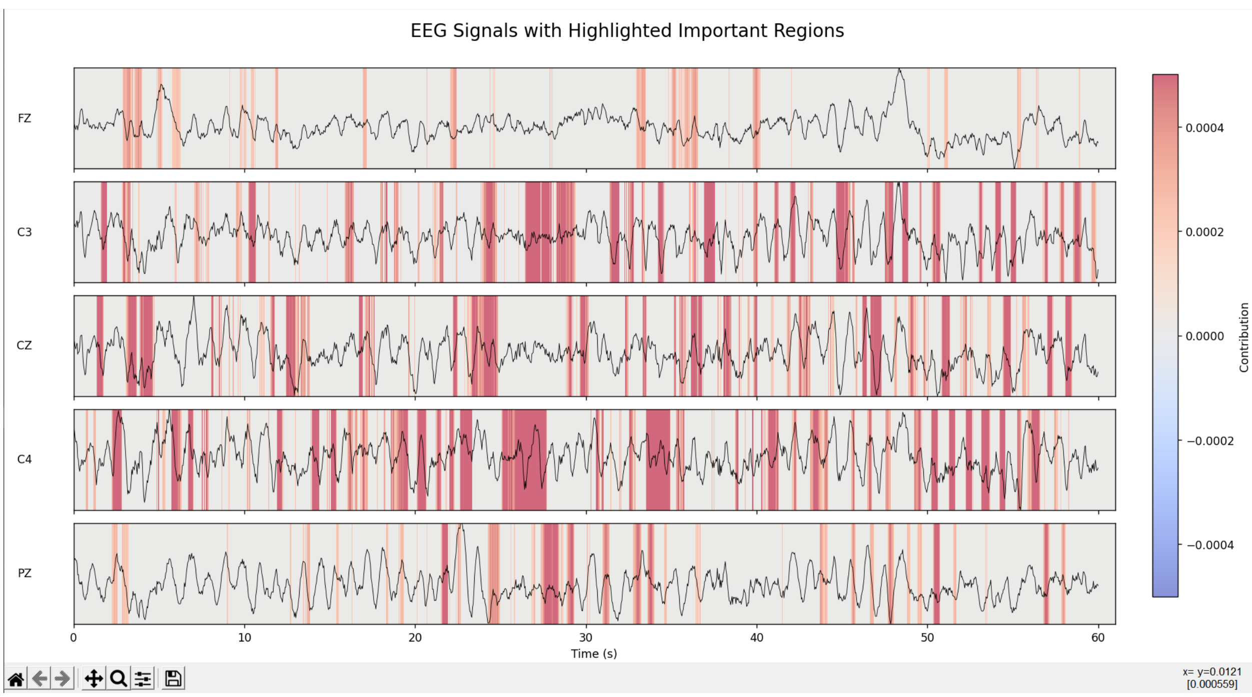

Figure 1: Temporal Features

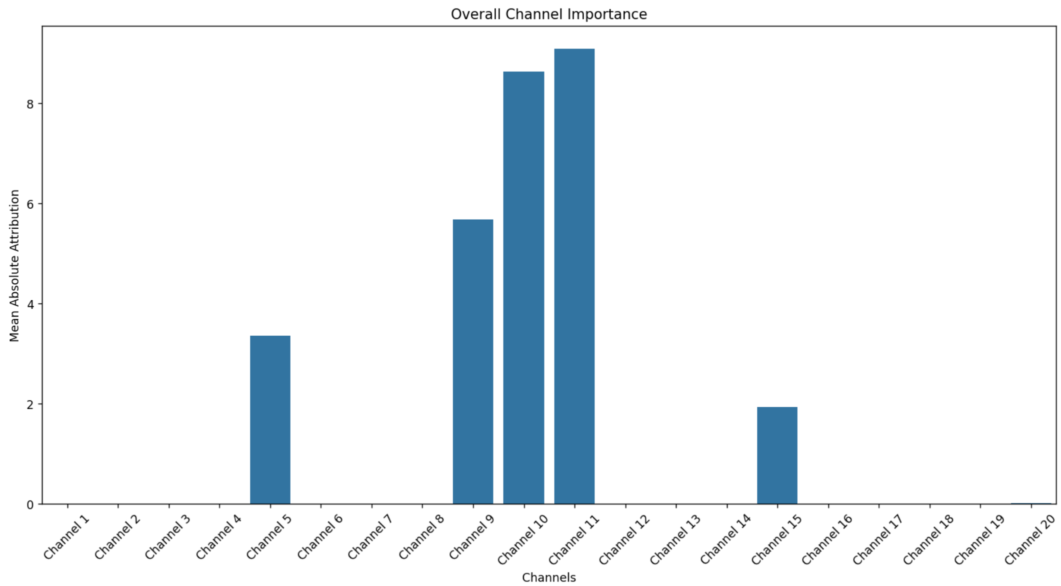

Figure 2: Spatial Features

4.2 Early Stopping Callback Implementation

To prevent overfitting and ensure that the model does not continue training once optimal performance has been reached, we implemented an early stopping callback. This feature monitors the model's performance during training and halts the process when improvements stagnate. By stopping early, we save computational resources and avoid overfitting, which is crucial for maintaining the model's generalizability across unseen data.

5 Challenges and Learning

Throughout the project, several challenges were encountered, each contributing to a deeper understanding of the complexities involved in EEG data processing and model optimization.

One of the most persistent challenges was managing the computational demands of the model, particularly in environments with limited resources, such as the Kaggle environment. The CUDA OOM errors highlighted the importance of efficient memory management and the need for strategies such as gradient and evaluation accumulation. These challenges underscored the significance of balancing computational efficiency with model performance, especially when dealing with large and complex datasets.

We identified that the evaluation bottleneck was primarily caused by inefficiencies in the data loading process. Using PyTorch profiling, we discovered that the data loader's behavior was a significant contributor to slow evaluations, particularly when using a large number of workers. To address this, we set the `num_workers` parameter to 0, which mitigated the stalling issues and significantly improved evaluation times.

Additionally, the option to disable evaluation during training was introduced, which reduced computational overhead and allowed for faster training in environments with constrained resources. The early stopping callback further ensured that the model did not overfit, halting training when improvements in performance plateaued.

6 Future Work and Recommendations

Looking ahead, several avenues for future work and improvements can be explored to further enhance the EEG foundational model and its applications:

- **Expansion of Dataset Compatibility**: Expanding the model's compatibility with other EEG datasets and formats will be crucial in broadening its applicability and ensuring its utility in various research and clinical settings. Future work could focus on developing additional data loaders and preprocessing scripts to handle diverse EEG data formats.

- **Enhancing Model Interpretability**: Improving the interpretability of the model's decisions is another critical area for future work. Techniques such as DeepLIFT and other attribution methods can be further explored and integrated to provide insights into how the model makes decisions and which EEG channels or features contribute most significantly to its predictions.

- **Optimization of Computational Efficiency**: While the integration of DistilledGPT has reduced computational load, further optimizations can be made to enhance the model's efficiency. Exploring alternative architectures or incorporating additional techniques for memory management and parallel processing could yield further improvements in training and evaluation times.

7 Conclusion

This GSoC project has successfully enhanced the NeuroGPT codebase, making it more capable and efficient for EEG data analysis. Through various additions and modifications, the foundational EEG model now stands as a robust tool. Future work will continue to build on these improvements, further expanding the model's capabilities and ensuring its utility in a wide range of applications.

In conclusion, I would like to express my sincere gratitude to the authors of the NeuroGPT codebase [**?**] for providing a solid foundation for this project. Their work has been instrumental in the development of the foundational EEG model, and their contributions to the field of EEG data analysis are greatly appreciated.

I would also like to thank my mentors,Dr. Mahmoud Zeydabadinezhad and Dr. Babak Mahmoudi, for their invaluable guidance and support throughout this project.

Thankyou.

References

[1] Wenhui Cui, Woojae Jeong, Philipp Thölke, Takfarinas Medani, Karim Jerbi, Anand A. Joshi, and Richard M. Leahy. Neuro-gpt: Towards a foundation model for eeg. *arXiv preprint arXiv:2311.03764*, 2023.