

E1-201: Hardware Acceleration on Machine Learning

Course Project: An efficient implementation of Brain Computer Interface (BCI EEGNet) decoder on Arduino Nano 33 BLE sense Lite.

Authors:

1. Hitesh Pavan Oleti (SR.No.:19804, ESE, IISc)
2. Anand Chauhan (SR.No.:19744, ESE, IISc)

Submitted to:

Prof. Chetan Singh Thakur,
NeuRonICS Lab, IISc.

File Hierarchy

1. Colab notebook → Contains colab ipython notebooks for 90 percent and 88 percent accuracy models
 1. Classification_Accuracy_90percent → Model with 90 percent testing accuracy
 2. Classification_Accuracy_88percent → Model with 88 percent testing accuracy
2. Dataset → Contains input dataset (both raw and filtered MNE EEG dataset)
 1. mne_input_data_withoutIIR → Used in testing (Pre-processing happens on Arduino)
 2. mne_iir_elliptic_5thorder → Used in trained (Pre-processed (on MATLAB) data)
3. Model Hex files → Contains Hex code files (quantized and non-quantized) of trained models
 1. Classification_Accuracy_90percent
 1. model → Quantized model file
 2. model_tflite → Quantized model file
 3. model_tflite_float → non-quantized model file
 2. Classification_Accuracy_88percent
 1. model → Quantized model file
 2. model_tflite → Quantized model file
 3. model_tflite_float → non-quantized model file

4. Arduino files → Contains Arduino implementation file along with Arduino libraries

1. Arduino → Contains Arduino libraries

2. BCI_decoder_MNE_with_onchip_preprocessing_90percentAcc →

(Arduino implementation of 90% testing accuracy model with on chip pre-processing)

1. BCI_decoder_MNE_with_onchip_preprocessing_90percentAcc.ino

2. model → Quantized model file

3. model_tflite → Quantized model file

4. model_tflite_float → non-quantized model file

5. MATLAB files → Contains test samples and MATLAB codes of GUI, UART transmission of input data, IIR filter design, pre-processing, and post-processing.

1. test_samples → Contains test samples categorized into all four labels
["Audio_left", "Audio_right", "Visual_left", "Visual_right"]

2. BCI_decoder_GUI → Top module which contains GUI code (initialization and functions)

3. give_all_samples → This function tests BCI decoder for all the test data and returns the number of correctly classified data samples and the overall testing accuracy.

4. give_one_sample → This function tests BCI decoder for a selected test sample and returns the predicted class.

5. model_stats → This function displays the BCI decoder model statistics.

6. SendImageData → This function reads the 16-bit packed fixed point input data and unpacks the read data and sends the two bytes of all the 9060 datapoints (i.e., 18120 Bytes) to Arduino through UART

7. uartinit → This function initializes UART port for serial communication

8. preprocessing_MATLAB_version → Convert input into Fixed point data and apply 5th order Elliptic IIR filter and quantize it.

9. mne_input_data_withoutIIR → Contains raw MNE data

10. Xtest_fixedpoint → Contains 16 bit signed (15-bit fraction part) fixed point data

6. PPT and Demo video → Explains the model and gives information about how to run the GUI.

7. References

8. Readme file

How to run the model

1. Arduino Part:

- Go to '4. Arduino files → BCI_decoder_MNE_with_onchip_preprocessing_90percentAcc'
- Open 'BCI_decoder_MNE_with_onchip_preprocessing_90percentAcc.ino' file.
- Compile the Arduino script.
- If compilation is successful, then upload the file to the Arduino board.
- After Uploading you can check the memory footprint of the model.
- Connect the PORT again and check whether the model is loaded correctly or not in **Serial Monitor**. It will print 'Tensor allocation status and model input dimensions' if model loaded correctly.

2. MATLAB Part:

- Go to '5. MATLAB files' and open 'BCI_decoder_GUI' file (top module).
- Run the GUI in MATLAB.

3. GUI:

- You can observe 3 buttons on the GUI.
 - Model Statistics
 - Select a Sample
 - Run all dataset
- Click on **Model Statistics** to see the model statistics.
- To check a single sample, click on **Select a Sample**.
 - Browse a sample and click open.
 - You can observe the Sample number, predicted label, Actual label and bar chart representing the predicted labels on the GUI
 - You can also check the output by observing the LED glowing on Arduino.
 - RED → Audio Left
 - GREEN → Audio Right
 - BLUE → Visual Left
 - WHITE → Visual Right
- To check all the samples of test data, click on **Run all data**.
 - You can observe the Total number of samples, Accuracy, Number of correctly classified samples, Current sample number, its predicted and actual labels (along with the bar chart representing the predicted labels) on the GUI.

- You can observe the changes in the parameters mentioned above while giving samples one after the other.
- At the end of all samples (71) classification, you can observe the testing accuracy of **90.64%**
- You can also check the output by observing the LED glowing on Arduino.
 - RED → Audio Left
 - GREEN → Audio Right
 - BLUE → Visual Left
 - WHITE → Visual Right

For more information check the PPT, Demo video, GitHub repository and the references.

THANK YOU