Lab 7 Embedded Artificial Intelligence on microcontroller

MicroAI – Build your dataset and make prediction in real-time

Polytech Nice Sophia

The goal of this lab is just to collect the data from the RF Things board and then to train a network on this new dataset.

Part I. Recording the data

Work to be done: Download from moodle the S4Lab7.ino, serial_client_offline.py and microAIGUI files.

The code in S4Lab7.ino is made for recording the data from the accelerometer and sending the data to the serial client that itself send the data to the GUI server to annotate the data. The S4Lab7 code is working in the following way:

- It initializes the accelerometer
- It is waiting 5 seconds, allowing time to disconnect the board and prepare to repeat the specific activity
- When “Ready” appears in the serial console the recording of the data starts
- During the recording the blue led blinks
- The duration of the recording is configured with DURATION constant (by default 5 minutes)
- All the data are stored in the accel_buf array, enabling to disconnect the board from the computer (if you put batteries at the bottom of the board)
- Once the capture is finished and while waiting for the host the LED stays on.
- At this time, you can start the client (serial_client_offline.py) and the server (microAI-GUI) to i) transfer the data and ii) annotate your data (put the labels positive on the data corresponding to the movement/activity and negative otherwise)
- Once the transfer is finished LED turns off.
- You can save your dataset from the microAI-GUI and stop the 3 executions.
- Optionally, you can restart a new capture by pressing the reset button on the card and the reset button on the GUI, keeping the Python script and the GUI open.

Part II Train a model onto your data

From the dataset created at step I, you need to create one training set and one test set (as for UCI HAR and MNIST) before training your model.

This is done by the new notebook provided on moodle. Execute the notebook and adapt the model in order to maximize the accuracy of your model. Note the step of windowing the data to make time series from the instantaneous samples.

After, validating you model, the notebook uses the MicroAI framework to generate the corresponding C code.

Save the network as a new model.h file.

This file will be used during the next session to make the prediction in real-time from the board.

Part III (lab 8) Real-time prediction on the board (during next session)

From the previous Arduino codes, you can adapt it in order to:

- Initialize the accelerometer
- What 5 seconds to disconnect the board and prepare your movements
- Read the data from the accelerometer
- Store the data in windows with the same length as the one used for training
- Call the CNN generated at the previous step for each new collected window
- In the case of a positive prediction, turn on the LED, otherwise turn it off
- When reaching a specific duration, stop prediction and wait for the connection with the computer
- When the connection is made, the transfer starts and the LED blinks
- You can visualize the validation data and adapt your model in the case of bad recognition

Reminder, the code of message types onto the serial link are:

- 0 for accelerometers data without timestamp (for real-time inference)
- 2 for inference data (counter, label and activity of the winner neuron)
- 3 for accelerometers data with timestamp (build a dataset)