Machine Learning for IOT - Homework 3

Sepehr Alidousti Shahraki, Shadi Nikneshan, Ali Ghasemi, Group 22 January 2022

Exercise 1

In this exercise we first recreated the models using the code in the previous homework (both CNN and MLP) considering the requirements of this homework. We used the window generator we made for the previous homework to create the dataset. We created the three Python files needed. The registry service file is used as the server and always must be running. We used MQTT protocols to transfer the alert message between the devices because the main purpose of MQTT is to transfer messages between devices with a low consumption of energy, so the prediction part of the exercise is done by using the MQTT protocols since MQTT is always connected against the intermittent REST Calls and MQTT allows the client to be connected always providing a two-way communication between the client and the server. We also must consider that REST is a one-way connection. The connection to the server is intermittent. The client connects to the server when needed to push data from the client and pulls the data down to the client. The server needs to wait for the clients to connect to send the data that is intended for the client and in this case, we don’t really want to deal with that

We used REST protocols for other tasks of the exercise. We used GET protocol for getting the list of the models because we are retrieving data from the server. We used POST protocol to add the model to the other device because we are trying to make some changes to the server.

The registery_service file is used on the as the server and the client_service file is used on the laptop/PC. We send the requests from laptop/PC, and they are performed on the raspberry pi.

The monitoring file is also used to give us more information about what is happening on the server.

Exercise 2

We used REST over MQTT since we wanted a two-way connection, but we didn’t want the devices to communicate to each other at the same time and, we must consider safety since we were transferring audio files and in real life scenarios those files might be sensitive. We used PUT protocol since we wanted to transfer data from one device to another and make some changes on the data and update it. In order to create the connection, we have to use the IP address of the laptop/PC on 8080 port.

For the preprocessing part we used MFCC for both devices but with small variations in the number of mel bins and the sampling rate. We chose a smaller number of mel bins for the fast registry file since we wanted to run it faster on our raspberry pi device and, we increased the sampling rate to make a positive difference, but it was useless, so we kept the default sampling rate which was equal to 16000, we also change the values for the frame length and frame steps to get a better result for accuracy and other criteria.

In the success checker part, we used two strategies at the same time. In the first strategy we set the threshold equal to 65 and our policy is that if the highest probably given by the model in our raspberry pi is less than 65 percent then we will ask the laptop/PC to perform the inference. Simultaneously, the other strategy is that we will ask the laptop/PC to do the inference if the difference between the two highest predictions made by the model is lower than 2.