Homework III - Machine Learning for IoT 2021-2022

Khudayar Farmanli
s276310

Fabio Tatti
s282383

Fabiana Vinci
s279918

I. Exercise 1

We implemented the registry service in order to manage the models and the different requests that can be done by the client. Each request is characterized by a class in the registry service in particular we have *AddModels, ListModels, Predict*. Respectively we decided to use the following HTTP Methods:

- **POST**, to **create** a new resource (the models) inside the server.
- **GET**, to **read** a representation of the resource (the name of the models)
- **GET**, to **read** the values (prediction of temperature and humidity)

The communication between the server and the new client/clients, which have to receive some alerts in case of error, is managed with the MQTT protocol. The MQTT is the most suitable communication protocol for this task because we can manage the communication with an undefined number of clients.

In particular, we created a publisher in the server and after the comparison between the values of predictions and ground truth with a given threshold, it creates a different publication for both Temperature and Humidity with the topics: */279918/TemperatureAlert, /279918/HumidityAlert*. The file $monitoring\_client.py$ manages the Subscriber part of the communication. It creates a subscription for both the topics and manages the visualization of the alert message.

II. Exercise 2

The collaborative inference involves two different pre-processing done respectively by the raspberry(fast pre-process) and the client(slow pre-process). The difference between them is in the parameters we chose to evaluate them. In the Fast pre-process we decided to apply a down sampling of factor 2 for the input files and change the number of mel bins to 16. In this way the process is faster but still with a good accuracy. The Slow pre-process is composed of the original mfcc hyper-parameters and does not involve the down-sampling of the audio so that the prediction can be more accurate.

We managed the communication between the two devices with the MQTT protocol for the following reasons:

- it enables a more fast communication. The message(both for audio and prediction) should be very fast in order to have an efficient prediction process;
- we do not know the IP address of the client in advance: MQTT can manage this with no problems, being capable to satisfy the request from any client, allowing us to have also multiple of them (we should only implement something that deals with multiple inputs at the same time)

Both the server and the client are built to be the Publisher and the Subscriber at the same time. When the maximum value of the probabilities calculated by the soft-max is under a specific threshold, the first message containing the audio is sent from the server to the client. $Threshold = 0.45$ is the optimal value found in order to respect all the constraints and the topic of the publish made is */282382/SlowAudio*. When the client receives the audio, it performs the evaluation and send a message to the server containing the new prediction with topic */282382/FastAudio*.

The final results obtained are:

- Collaborative Accuracy: 91.625%
- Total fast inference time: 39.40375 ms
- Communication Cost: 1.351732 MB