Machine Learning for IoT
Homework 1

Di Nepi Marco S277959 | Benedetto Irene S276200 | Falletta Alberto Maria S277971

1 Data Preparation: Sensor Fusion Dataset with TFRecord

The script builds a `TFRecord` starting from a csv file. For the sake of simplicity it is assumed that the name of the file in the input folder is always 'samples.csv' In order to minimize the storage, the following data types have been used:

- **Date** has been converted in POSIX format and saved as an `Int64List`. Although the space occupied was slightly less when using float, this led to data loss and incorrect result;
- **Temperature** and **Humidity** have been read as `int32` and saved as `Int64Lists`;
- **Audio** has been firstly read through tensorflow's readfile function and then saved as `BytesList`.

With this solution the dataset obtained has a size that is almost the same of the original set of files, with only a small overhead (some kbs) due to the transition to a more structured format.

2 Low-power Data Collection and Pre-processing

The goal of this script is to minimize the amount of time spent in performance mode (and consequent energy consumption) without compromising the time required to preprocess an audio sample and save the resulting MFCC. The script is divided in 3 parts:

1. In the first phase only the operations that can be done once are present. As instance, the computation of the weight matrix, the instantiation of the `PyAudio` class, the recoring stream, and the resetting of the monitor.
2. In the second phase a sample is recorded. The `Popen` instantiation and the change of the operating mode take 150ms, and in order to run the preprocessing phase at maximum frequency, this operation is performed during the recording. In particular, we decide to set the chunk size equal to 2400: this value allows to obtain 20 iterations of recording loop, each lasting 50 ms. This value is actually the greatest common divisor between 1000 ms (the recording time) and 150 ms (the time required for scaling governor) and, as a consequence, the call for setting the operating mode to performance is made in the third last iteration. Moreover, every time a new recording cycle begins, through Popen, the VF level is set again to the minimum value (except for the first sample recorded, as the powersave mode is set at the beginning of the script). Since the registration is the most time-consuming part of the pipeline, it must be done in powersave mode, in order to minimize the amount of operations made at 1.5GHz. Finally, it can be noticed that in order to avoid writing files on disk (operation that may cost some milliseconds), the chunks have been stored in the in-memory bytes buffer through the `BytesIO` class from `io` package.
3. Finally, the preprocessing phase is done in performance mode. Observe that the tensor returned by the `resample_poly` function has been casted to `float32` to reduce computation time and the MFCC has been saved as a binary file.

In conclusion, the resulting latency of the loop is 1.065s on average. This satisfies the 80ms hard constraint for preprocessing and also minimizes the VF_{max} time.