Homework 1 - Team 17

Francesca Cossu s305746@studenti.polito.it
Niccolò Dimonte s303439@studenti.polito.it
Michele Presti s290453@studenti.polito.it

Exercise 1

The task given consist in optimizing the hyper-parameters if the *is_silence* method in order to satisfy the constraints:

- Accuracy on the *vad-dataset* > 98%
- Average latency < 9 ms

The *is_silence* method classifies an audio file as *silence* or *not silence* using STFT features.

The Short-time Fourier transform is obtained by windowing the signal and taking the Discrete Fourier Transform of each window. The STFT represents the signal by a sum of sinusoids that provide more insights and properties of the signal compared to the DFT; this happens because speech is not a stationary signal, but has properties that change overtime that are not analyzable using a discrete transform.

The parameter *downsampling_rate* is settled at the same value as the sampling rate, because we don't want to apply a downsample on the data since this would have increase latency time significantly.

The parameter *frame_length_in_s* defines the frame length in the STFT in second.
The frame (or window) size of the STFT is typically a power of 2. With a grid search using powers of two, we found out that the best performance is with a frame length of 4 ms.

The parameter *dbFSthresh* is the threshold for energy value of the signal's spectrogram and it's necessary to count the total duration of non-silence frames.
From the data exploration we found out that the average of the energy of "silence" audios in the vad-dataset is -140.
With grid-search we found out that the best results are obtained setting the parameter as -144 because the scale of digital audio goes from 0 to -144, and choosing the bottom of this scale permits to take into consideration all the signals in the spectrum.

The *duration_thresh* is used to classify as *not silence* each audio with a duration time, that is the sum of all the *non silence* frames, greater than this parameter.
Analyzing different values for the parameter *duration_thresh*, the best results are obtained when it is in scale w.r.t. frame length.

After many experiments and grid search for tuning, we can say that the total accuracy is mainly influenced by the combination of dbFS threshold and duration time, while the average latency is influenced only by the window size and the presence of a downsampling. Thus, the dbFS threshold is slightly influenced by the window size.
The last configuration after many tests is the following:

dbFSThresh	Duration thresh (ms)	Accuracy (%)	Average latency (ms)
-144	0.04	98.78	8.74
-133	0.04	98.56	8.88
-100	0.04	91.56	8.50
-144	0.50	75.67	8.97
-144	0.20	96.89	8.54
-144	0.10	98.56	8.41
-144	0.02	98.67	9.45
-144	0.002	88.88	9.04
-144	0.004	98.22	9.07
-144	0.002	88.88	9.04

Table 1: Grid search with *frame_length_in_s* fixed at 4ms

- *frame_length_in_s* : 0.004
- *dbFS Threshold* : -144
- *Duration time* : 0.04

Exercise 2

The battery monitoring system proposed is based on the collection of data about the battery status: battery level and power plugged. The data stored in the timeseries are both int16: for the first timeseries *mac_address:battery* an integer for the value of the percentage, while for *mac_address:power* timeseries a boolean value that is 1 if plugged, 0 otherwise. This first two time-series collect and store data each second. It is also requested a new timeseries that computes and stores how many second the power have been plugged in the last 24h. For all three timeseries, the largest retention time must be set such that the following memory constraints are met:

- *battery* < 5MB
- *power* < 5MB
- *plugged_seconds* < 1MB

According to the documentation the average compression ratio is 90%. The formula used to compute the retention time of *battery* and *power*, is the following:

$$\frac{\text{MEM\_LIMIT}}{\text{RECORD\_SIZE x (1 - COMPR\_PERC)}}$$

The largest retention time is about 37,9 days.
To create the *mac_address: plugged_seconds* time series, we created a rule that sums every 24h the value of *power* time series. It's possible to do this because the value stored is 1 if plugged and it is is stored each second. Therefore the sum of values within 24h will be the total number of seconds which in the power is plugged.

For computing the largest retention time w.r.t. memory constraints, we used the following formula:

$$\frac{\text{MEM\_LIMIT}}{\text{RECORD\_SIZE x (1 - COMPR\_PERC)}} \text{ x DAY\_IN \_SEC}$$

The computed retention time is approximately 1795,5 years.