Machine Learning for IoT - Homework 02

Alice Colombatto s289922 | Arcangelo Frigiola s295406 | Francesco Grandi s306272

Exercise 1 - Training and Deployment of a "Go/Stop" Classifier

The goal of this exercise was to train and deploy a model for a "Go/Stop" words classification task, using the *mini speech command* data-set. We defined the model, the optimization flow, and the correct setting of the hyper-parameters in order to meet the constraints.

- Starting from recommended hyper-parameters seen during laboratories, we adopted a grid-search methodology for both pre-processing and fine tuning part to find the optimal ones. We reached all the given thresholds (accuracy, latency and memory size) implementing a DS-CNN with weights pruning. For the pre-processing step, we performed a grid-search to find the parameters on Table 1, essential to extract the most significant features of a given audio through the *Spectrogram* and the *Log-Mel Spectrogram* functions.

	Downsampling rate	Frame length	Frame step	Num_mel_bins	lower_freq	upper_freq
Spectrogram	16000	0.05	0.025			
Log-Mel spectrogram	16000	0.05	0.025	30	20	8000

Table 1: Pre-processing hyper-parameters

batch_size	initial_learning_rate	end_learning_rate	epochs	alpha	final sparsity
15	0.01	10^{-5}	10	1	0.95

Table 2: Training hyper-parameters

- For the model architecture, we used a DS-CNN with an Adam optimizer and the hyper-parameters reported on Table 2. To guarantee the appropriate accuracy of the CNN and avoid increasing the final model size, we set more *DepthwiseConv2D* layers. This brings to a smaller number of variables compared to a classic convolutional layer. We also fixed the value of α equal to one; this let us insert manually the values of the filters through the different layers of the CNN. Eventually, the layers turn out to have 48 filters.

Accuracy	TFLite Size (in KB)	Total Latency (ms)
99%	23.455	7.7

Table 3: Output values of the final solution

Observation.
Due to the fact that our model has his personal construction, we just want to precise here that the so called PREPROCESSING_ARGS used in the *Inference* for model testing are Downsampling rate=16000, Frame length=0.05, Frame step=0.025, Num_mel_bins=30, lower_freq=20, upper_freq=8000.

- At the end of our pipeline, we obtained the results reported in Tables above. We found out that we have a half overlap both for the spectrogram and the log Mel spectrogram. For the other values - num_mel_bins, lower_frequency and upper_frequency - we maintained values that belong to the suggested intervals studied in the course.
Furthermore, for what concerns the model architecture, we found out that using different types of values of the filters (power of 2) seem to make the model perform better. In conclusion, the batch size have been set to 15 as the accuracy of the model seemed to improve, while not increasing the time needed for training.