# Group 5 - Report Homework 2

s277757 Ruben Berteletti - s277771 Riccardo Baldassa - s276525 Paolo Fiorio Plà

December 2020

## 1 Exercise 1: Temperature and Humidity Forecasting

The aim of this exercise is to create two TFLite models trained on the *Jena Climate Dataset* that respect the following constraints:

- **Version a**: T MAE < 0.5°C and Rh MAE < 1.8% and TFLite Size < 2 kB
- **Version b**: T MAE < 0.6°C and Rh MAE < 1.9% and TFLite Size < 1.7 kB

The starting point was to train the MLP and CNN without optimization, in order to understand which "knob" should have been triggered to optimize the model, obtaining as MAE [0.93°C, 1.85%] in 72.91 kB for the first and [0.60°C, 2.13%] in 65.98 kB for the second.
Given these results, both this models need optimization and since the CNN is lighter than MLP, we decided to use it for the version b which has the stronger constraint on the size, while MLP for the version a due to its smaller MAE on humidity (found as the hardest error to reduce).
Our choice was to start to perform the optimization using a single technique at a time among *Magnitude-based pruning*, *Structured pruning* and *Post-training quantization* (done only for the weights), but despite we obtained improvements, either MAE or size were out of bound.
So, we decided to tune the model with a combination of the previous methodologies and we reached models compliant with the request using the following hyperparameters:
**Version a**: MAE = [0.44°C, 1.75%] in 1.95kB using structured pruning with width multiplier = 0.25; Magnitude-based pruning with final sparsity = 0.90 with *zlib* compression; PTQ.
**Version b**: MAE = [0.47°C, 1.84%] in 1.66kB using structured pruning with width multiplier = 0.12; Magnitude-based pruning with final sparsity = 0.78 with *zlib* compression; PTQ.

## 2 Exercise 2: Keyword Spotting

This exercise requires to train three TFLite models on the *Mini Speach Command Dataset* compliant with the constraints below:

- **Version a**: accuracy > 90% and TFlite size < 25 kB
- **Version b**: accuracy > 90% and TFlite size < 35 kB and inference Latency < 1.5ms
- **Version c**: asccuracy > 90% and TFlite size < 45 kB and total Latency < 40ms

We adopted here the same reasoning as ex.1 and we saw immediately that without optimization the model which performs better has been the DS-CNN then we kept it as a starting point for each version, tuned then differently depending on the models' purposes. We used in each pipeline the *zlib* compression.
**Version a**: here the most strict constraint regards the size, then we have been driven to an usage of structured pruning with a small width-multiplier, set as 0.31 and to be sure to achieve the 90% accuracy we used MFCC as preprocessing which guarantees same size of STFT (for CNNs) with better precision. Finally, we adopted PTQ reaching 90.88% accuracy in 24.12kB.
**Version b**: testing the inference latency of version a, we noticed it to be out of bound then to speed up the inference we decided to keep MFCC (faster than STFT) and since we had margin on the size we avoided to use PTQ which has huge impact on the latency, reducing instead the width multiplier to 0.22. Due to this high pruning we set the learning rate in the training phase to 0.03 in order to reach the 90% accuracy in 20 epochs. With this setting the model achieves an accuracy = 91.12% and an inference latency of 0.60ms in 32.72kB.
**Version c**: here the pre-processing has the highest impact on the total latency. Since MFCC requires several computation more than STFT, we adopted the latter as pre-processing choice. In this way even though the total latency was under 40ms, it was hard to keep the accuracy over than 90%. To do so, we used as width-multiplier 0.52, we set the learning rate = 0.01 through 30 epochs and finally we exploited the PTQ. With this hyperparameters we have accuracy = 90.12%, total latency of 25.60ms in 35.19kB.