Report Homework 1

Machine Learning for IOT

Authors:

Antonio Dimitris Defonte s276033
Eleonora Carletti s280056
Filippo Cortese s273672

Exercise 1

In this exercise we had to train multi output and step models for temperature and humidity forecasting respecting two different set of constraints. The first one was *MAE of Temperature < 0.5°C* , *MAE of Humidity < 1.8%* and size of the *TFLite model < 2kB*, the second one was *MAE of Temperature < 0.6°C* , *MAE of Humidity < 1.9%* and size of the *TFLite model < 1.7kB* . We started with a CNN-1D model and a MLP model without any further improvement to see the starting sizes and errors. Both models were close to the desired errors but both exceeded the size constraints by far (*MLP size: 79 kB*). Then we started to think about techniques capable of reducing the size while keeping the error low. We started with a simple post training quantization of the weights to int8. This would give us a huge reduction in size (*MLP size: 22.64kB*) but too high errors, so we thought that we could act on the size by trying to exploit the redundancy and sparsity of the models through structured pruning via width scaling. This would need the optimization of an hyperparameter alpha to find the sweet spot where the size is reduced and the errors are not too high.
That value was found, after 2 consecutive grid search approaches, in alpha equal to 0.05. With this technique we obtained a good reduction of the size (*MLP:3.16kB*) and we started to compress with zlib the file to further reduce the dimensions. (***MLP: 1.67kB***). The MLP model was respecting the error (***MAE T: 0.387, MEA Hum: 1.757***) and dimension constraints of model “a” and model “b, so we decided to stick with it. Then to be more sure to respect the size constraint of *1.7kB* we decided to apply also a post training quantization to float16 on the weights. This would allow us to reduce even more the size by leaving almost invariant the errors (***MLP compressed size: 1.46 kB, MAE T: 0.388, MAE H: 1.757***). This configuration in the end was able to respect both sets of constraints at once.

Exercise 2

In this exercise we had to create models for keyword spotting that have to fulfill three different sets of constraints.
The first one was about keeping an accuracy higher than *90%* and a TFLite size of the model less l than *25 kB*.
To do so we decided to use the DS-CNN model to have a good compromise between size of model and accuracy and the mfcc’s representation of sound because it was the one performing best in terms of quality of prediction.
We started from a dimension of *555.61 kB* and an accuracy of *92.875%*, so we had to drastically reduce the size.
Since in the first exercise we obtained good results with width scaling and then zipping, we thought we could exploit the redundancy and sparsity also on this net and so we applied the same strategy.
By using a grid search approach we obtained a value for alpha equal to 0.3, but width scaling alone doesn’t allow us to reach the constraint about model size *(DS-CNN size: 61.31kB DS-CNN zipped size: 54.33kB)* and decreasing more the alpha parameter would have led to a consistent decrease in the accuracy.
For this reason we decided to apply a post training quantization on weight to further decrease the size and keep an appropriate level of accuracy, with this operation we were able to reach an ***accuracy of 90.25%*** and a ***TFLite size*** of the zipped model equal to ***22.98 Kb***. This not only satisfies the first constraints but also two of the second set of constraints which are *Accuracy > 90%* and TFLite *model size < 35kB.*
By testing the model for inference latency we found out it took a ***time of 1.16 ms*** which is enough to meet also the last constraint of the second set (*latency < 1.5 ms*), this is due to the fact that width scaling lose lot of the filters of the convolutions and so save also the time that would be spent to execute the operations related to those.
The last constraint asked to reach an accuracy higher than 90%, a TFLite size of the model less than 45 Kb and a total latency less than 40 ms, but by trying out several combinations of pruning via width scaling and quantization, relaxing the size reduction and also by trying to move from mfcc to stft to skip a computation, we found out that these operations were not enough to reduce the latency time while keeping an high accuracy.
For this reason we decided to operate in the preprocessing phase and in particular we applied a resampling to the audio signal to reduce the number of entries and so the preprocessing time. We also had to resize the windows of the STFT in order to keep extracting a good amount of information from the audio samples. So by applying to the input samples a resampling to 8 kHz, STFT with “frame_lenght” equal to 240 and “frame_step” equal to 120 and a standard MFCC and feeding them to a DS-CNN (mfcc) model with width scaling (*alpha= 0.5*) and weight quantization we were capable of obtaining an ***accuracy of 93.25%***, a ***TFlite size*** of the zipped model equal to ***35.68kB*** and a ***total latency*** equal to ***34.61 ms***.

COMMAND LINE PARAMETERS FOR KWS_INFERENCE.PY:

- Model A:. --mfcc --model ./Group1_kws_a.tflite
- Model B: --mfcc --model ./Group1_kws_b.tflite
- Model C: --mfcc --model ./Group1_kws_c.tflite --rate 8000 --length 240 --stride 120