Homework #3 Report

Group #18
Eduard Ciprian Ilas s277901
Gabriele Cuni s277957
Valeria Sorrenti s276146

1. **Big/Little Inference**

   The proposed Big/Little configuration is composed by:

   - A REST web service that receives a raw audio signal in SenML+JSON format, runs inference on the big model and returns the predicted label in JSON format. The big model chosen for this purpose was a DS-CNN with an added depthwise layer, trained for 25 epochs, that managed to score a 95.13% of accuracy on the test set.
   - A client application that reads audio signals and runs inference using the little model and when needed sends a PUT request to the Big model service. Such behaviour is dependant on the level of confidence of the little model prediction: the big model prediction is taken when the difference between the first two predicted labels probability is greater than a certain threshold, which has been empirically determined to be 0.5. The little model is a DS-CNN trained with STFT features, has an accuracy of 89.7% on the test set and a size of 19.715KB once compressed (optimisations: weights-only PTQ, width scaling $\alpha = 0.29$, pruning with sparsity of 0.32, total latency of 19.46ms).

   The final little/big configuration is able to achieve an accuracy of 93.125%, with a communication cost of 4.34 MB, by making 109 calls to the big model service. Although a MQTT implementation might have given a lower bandwith usage, we decided to adopt the REST protocol for the communication between the two application because of the one-way connection setup between the server/client: the client only connects to the (one) server when needed, makes the request and waits for an answer, therefore a REST implementation seemed more adequate.

2. **Cooperative Inference**

   We decided to employ a MQTT implementation for the second exercise. The nature of the problem made it so that the cooperative client only needs to make the requests, and then wait for them to be fulfilled by the array of inference clients, rather than connecting to each of them individually, make the request and wait for a reply. Another advantage is given by the fact that the cooperative client is agnostic with respect to the number of inference clients fulfilling the requests (although the assumption that they are known and all running has been made here).
   The cooperative client reads the audio signals, generates both STFT and MFCC features and sends them to the broker using different topic paths, using SenmML+JSON. The feature publication rate was slowed down to 10 per second, in order not to overwflow the broker. The inference results are then sent by the clients using different paths and saved by the cooperative client for the final test. Five inference clients were deployed, each using a different model: a DS-CNN with MFCC features (95.13% accuracy), a ResNet18 CNN with MFCC features (94.63% accuracy), a simple CNN with MFCC features (91.75% accuracy), a DS-CNN with STFT features (91.12% accuracy), and a simple CNN with STFT features (91.37%). The final prediction lable choice has been made by majority voting (and in case of general disagreement, the most confident prediction was taken). The final accuracy on the test set was of 95.87%.