Group 5 - Report Homework 3

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

January 2021

1 Exercise 1: Big/Little Inference

In this exercise the goal is to create a Big/Little model for Keyword Spotting using the *Mini Speech Command Dataset*. The Big/Little model is composed by a Big neural network running on the notebook that hosts the Web Service and by a Little neural network running on the Raspberry Pi that works as a client communicating with the **REST** protocol because request-response protocol is suitable for this case, since there's only a client and a server and the synchrony between them is necessary. The system works in the following way: the client application reads one by one the audio signals from the Speech Command test-set, tries to predict using the Little model and when necessary it calls the web service that runs inference with the Big model. This decision of calling the web service is done with the so-called Score Margin(SM), calculated as the difference between the first and the second highest probabilities of label predictions. According to the SM chosen the number of calls will change, consequently accuracy and communication cost. Three different examples are reported below, the first represents the minimum SM able to reach the requested accuracy, the second reaches the highest accuracy while the third is the last able to maintain the communication cost under 4,5MB:

- SM <= 0.08 Accuracy: 93,125% — Big requests: 6 — Communication cost: 0,245MB
- SM <= 0.8 Accuracy: 94,625% — Big requests: 71 — Communication cost: 2,835MB
- SM <= 0.973 Accuracy: 94,500% — Big requests: 110 — Communication cost: 4,392MB

Models description:

1. *Little*: DSCNN with 3 convolutions (256 filters each) and a dropout layer (0.3), width multiplier = 0.33, PTQ and magnitude-based pruning usage; Accuracy = 92.875%; Size = 19.55 kB (compressed); Inference time = 34,13ms;
2. *Big*: DSCNN with 4 convolutions (256 filters each) and a dropout layer (0.6); Accuracy = 94.750%.

As preprocessing common technique MFCCs is adopted using as parameters (frame-length: 400, frame-step: 200, mel-bins(big): 20, mel-bins(little): 40). The little model has been fed with 8000 kHz resampled audios.

2 Exercise 2: Cooperative Inference

Here the goal is to reach, through a models ensemble potentially running on different devices, an accuracy greater than 94%. The task is suitable for **MQTT** since the preprocessed audios are sent only once to the broker which delivers them to all the subscribers, i.e.the different models, that predict the labels and send them back. One parameter of MQTT is QoS (Quality of Service), set to 2 for the audio transmission because guarantees the correct reception of files while for the prediction transmission is set to 0 since, given N models, the loss of one prediction does not determine a drop in term of accuracy. The audios are sent by the cooperative client all at once to the inference clients without waiting sample by sample the predictions, instead those are stored in a dictionary as they come back (the dictionary has as key an identifier and as value a list of predictions done by the N inference clients). This choice allow to handle asynchronous inferences, however is needed to wait the last prediction in order to apply the *cooperative policy*. For this task the inference models used are 3 (described in detail below) and since they achieve very similar accuracy, i.e. within 1% range, the policy adopted is the **majority voting** rule, that is able to reach **95**% **accuracy** on the given test set.
Models description:

1. DSCNN with 4 convolutions (256 filters each) and a dropout layer (0.6) — Accuracy = 94.750%;
2. DSCNN with 4 convolutions (128 filters each) and a dropout layer (0.6) — Accuracy = 94.400%;
3. CNN with 3 convolutions (128 filters each) and dropout layer (0.4) — Accuracy = 93.875%.

As preprocessing common technique MFCCs is adopted using as parameters (frame-length: 400, frame-step: 200, mel-bins: 20).