POLITECNICO DI TORINO

MACHINE LEARNING FOR IoT

HOMEWORK 3

GROUP 14,

BERNAUDO Jacopo, s276228

ERFANMANESH Omid, s272519

VASSALLO Maurizio, s276961

REPORT HOMEWORK 3

Ex1) Big/Little Inference

The solution for little model is:

- The model architecture is a **DSCNN** with 3 Conv-2D layers with 256 filters each and 1 final Dense layer with 8 neurons, the possible labels;
- The dataset is generated with **mfcc=True**, **frame_length=480** and **frame_stride=320.** These reduced the preprocessing time;
- 2 types of optimizations are used: Weight-Only Integer Quantization, Structured Pruning (with **alpha=0.234**);
- The model is compressed using *zlib* library.

The results are:

> *Compressed little tflite model size: 16.89kB, Accuracy: 0.904, Total Latency: 38.84ms.*

The solution for big model is:

- The model architecture is a **DSCNN** (same as above).
- The dataset is generated with **mfcc=True**, **frame_length=640** and **frame_stride=320.**
- To increase the performance, we used an alpha greater than 1; in this case **alpha=1.5.**

The results are:

> *Big tflite model accuracy: 0.94.*

The communication protocol used is **REST**, since the system is "simple": only one client that asks and only one server that answers. The client sends the data with a **PUT** method.

The Success Checker policy is implemented as follows: we find the difference between the two highest confidence predictions of the little model, if this difference is larger than a threshold we ask the bigger model. This threshold is found trying different values and the best found is: **threshold=0.4**.

Result of the Big/Little model system:

> *Accuracy: 93.500%*
> *Communication Cost: 4.229.*

Ex2) Cooperative Inference

For the Cooperative Inference we use 3 models (**N=3**). All models are of the type **DSCNN**(same configuration as above)**, mfcc=True**, **frame_length=640** and **frame_stride=320;** what changes is the value of alpha:

- Model 1: alpha=1.5. `Accuracy: 0.940.`
- Model 2: alpha=1.6. `Accuracy: 0.936.`
- Model 3: alpha=1.53 (first Conv2D layer has **filters=256*2**; this in order to get more information from the input data). `Accuracy: 0.930.`

The protocol used for the communication is **MQTT**, since the system is more complex: there is one client who sends the data and multiple inference clients answering.
The client, after sending the data, does not stop and wait for all the N answers but it keeps sending the data; since this process can be done in parallel. To achieve this parallelism, on the client we store in a dictionary the answers; in particular: the key of this dictionary is the line number of the audio sample (the number line inside the *kws_test_split.txt*); and the value is an array of length 1+N, where the first element is the true label and the others are the answers. After receiving all the answers we find the most voted label and this will be compared with the true label. With this implementation, in our case, 800 test samples, it requires around 48 seconds for sending data and receiving the answers.

Result of the Cooperative Inference model system:

```
Accuracy: 94.750%
```