Machine Learning for IoT - Report Homework 2

Giuseppe Acquaviva, Mario Capobianco, Marco Mungai Coppolino
Politecnico di Torino, Italy

I. Exercise 1 - Training & Deployment of a "Up/Down" Keyword Spotter

A. Methodology for Hyperparameters Optimization

The primary goal was to reduce the TFLite model below 50 KB while keeping the median latency of the inference under 40 ms. Once this was achieved, a grid search was conducted to ensure the required accuracy was met:

Pre-processing type: MFCCs were chosen for audio feature extraction due to their superior performance over Mel spectrograms in our tests. While Mel spectrograms capture frequency information, MFCCs provide a more compact representation by emphasizing perceptually relevant features. This improved representation proved more effective in maintaining accuracy for keyword spotting tasks.

Hyperparameters search: The grid search focused on optimizing MFCC parameters by limiting frame lengths and steps to power-of-two values, improving efficiency. Parameters were initially tested individually while fixing others at average values within predefined ranges. This approach narrowed the ranges, reducing the # of param combinations and therefore the computational cost of a subsequent full grid search.

TABLE I
Pre-processing Type and Hyperparameters

Parameter	Value
Pre-processing Type	MFCCs
Frame Length	16 ms
Frame Step	8 ms
Num Mel bins	10
Lower Frequency	40 Hz
Upper Frequency	4000 Hz
Num Coefficients	10

TABLE II
Training Hyperparameters

Parameter	Value
Batch Size	20
Learning Rate	0.01
End Learning Rate	1.e-6
Epochs	22
Lr Scheduler	linear decay
Optimizer	Adam
Loss Function	Sparse Categorical Crossentropy

B. Model Architecture and Optimizations

Model Architecture: The model is a convolutional neural network (CNN), comprising three convolutional blocks with 256 filters, a 3×3 kernel, batch normalization, and ReLU activation. The first block uses a stride of 2 with 'valid' padding, reducing spatial dimensions, while the next two use a stride of 1 with 'same' padding to preserve them. A global average pooling layer condenses features, and a dense layer with softmax activation maps them to output probabilities.

Optimization Techniques:

- DSCNN (Depthwise Separable CNN): To address size constraints, a Depthwise Separable CNN (DSCNN) was adopted, significantly reducing the number of parameters by decoupling spatial and channel-wise convolutions. This approach was implemented by adding a DepthwiseConv2D layer, with 1×1 kernel and 1×1 stride, after the first and the second convolution. Despite the reduction, the TFLite size was still above the required limit, and further optimizations were needed.
- Structured Pruning: Structured pruning was applied to the DSCNN architecture, leveraging a width multiplier of 0.25 to scale down the number of filters in the convolutional layers. This approach successfully reduced the TFLite size below 50 KB while maintaining acceptable accuracy and latency, fulfilling all constraints.

TABLE III
Model Performance Metrics

Metric	Result
Accuracy (Lite)	99.5%
Model Size (Lite)	44 KB
Accuracy (Zipped)	99.5%
Model Size (Zipped)	39 KB
Total Median Latency	25.8 ms

C. Results and Discussion

The TFLite model achieved an accuracy of 99.5%, with a compact size of 44 KB, and a median latency of 25.8 ms. These results demonstrate compliance with all constraints while maintaining high performance. Additionally, the zipped model showed similar accuracy and a reduced size of 39 KB, further optimizing deployment for resource-constrained devices.

TABLE IV
Grid Search Parameter Ranges

Parameter Ranges
Frame Length (s): {0.008, 0.016, 0.032}, Overlap: {0.0, 0.25, 0.5, 0.75}, Num Mel Bins: {10, 20, 30}
Lower Frequency (Hz): {20, 40, 60}, Upper Frequency (Hz): {2000, 4000, 6000}, Num Coefficients: {0, 10, 20}