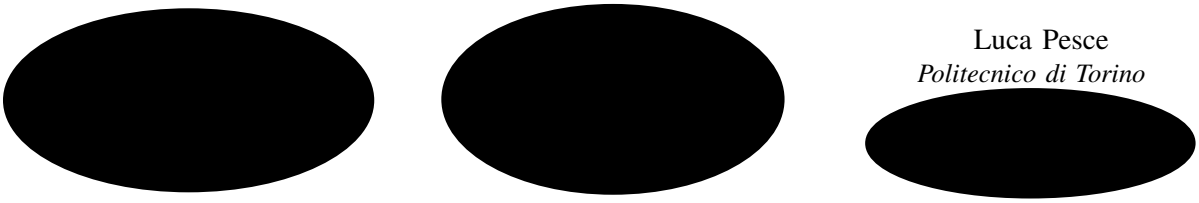
Second homework Machine Learning for IoT



Luca Pesce
Politecnico di Torino

***Abstract*—This research delves into the optimization of a audio binary classification model, the main focus is on hyperparameter tuning and architectural enhancements, by fine adjusting parameters and structural elements. The target is to minimize model complexity and accelerate inference speeds without affecting accuracy, but rather increasing it.**

I. Model optimization

A. Hyperparameters Tuning

The methodology utilized to determine hyperparameters, aligned with the constraints, involved a search of the best values to reduce the input shape on our model, capturing as much information as possible, in order to increase the model's speed mantaining an acceptable accuracy. Since extracting mfccs (Mel-frequency cepstral coefficients) is more computationally expensive than using only mel bins, we believed that the minimum number of mel bins and mfccs would suffice for a binary classification task, so we maintained these two parameters as low as possible. We tried different combinations of frame length and frame step concluding that processing the audios using only 34 frames (against 49) was enough. Then, we continuously trained the model with different epochs, while scheduling the learning rate and checking the loss status, these operations allowed us to tune better the number of mel bins, the frequency range and the number of mfccs. We noticed that the high frequencies were disrupting the performance of the models that were being tested, so we restricted the frequency range. In terms of training parameters, we kept constant the learning rate range during all the trials, and we also maintained the batch size constant at 10 to get an improved generalization during the training.

B. Model's Architecture

The model's architecture consists of a convolutional neural network comprising two convolutional layers. Each layer is followed by batch normalization and ReLU activation functions. To prevent overfitting (and to improve performance), we applied dropout regularization after each convolutional block, by randomly disregarding 20 percent of nodes during training after the convolutions in each epoch. The use of a 3x3 kernel size enables a detailed examination of the input data, covering a small receptive field. Additionally, employing 1x1 strides ensures that no information is skipped during the convolution process, preserving intricate temporal MFCC patterns. Furthermore, the model incorporates a global average pooling layer which facilitates the comprehensive feature extraction. We deliberately minimized the number of filters in each layer to reduce model complexity. Post-training, we removed the dropout elements from the trained model and converted it into TFLite format. We applied default optimizations provided by TFLite, including quantization, reducing the precision of the model's weights and activations from floating-point to lower bit precision. This precision reduction significantly reduced the model's size and enhanced inference speed while maintaining accuracy.

TABLE I
Preprocessing hyperparameters

Hyperparameters	**Value**
Sampling Rate **[Hz]**	16000
Num of Mel Bins	20
Lower Frequency **[Hz]**	20
Upper Frequency **[Hz]**	5000
Frame Lenght **[s]**	0.05
Frame Step **[s]**	0.028
Num of MFCC Coefficent	10

TABLE II
Training Hyperparameters

Hyperparameter	**Value**
Batch Size	10
Initial Learning Rate	0.1
Final Learning Rate	1×10^{-4}
Epochs	40

TABLE III
Final Solution Results

Hyperparameter	**Value**
Accuracy **[%]**	99
Model Size **[Kb]**	11.52
Total Latency Savings **[%]**	38.3

C. Commentary on the solution

The outcomes highlight the model's notable performances. An evaluation test accuracy rate of 99% coupled with a minimal test loss of 0.059 underline the strenght of the model's predictive power. The notable 38.34% reduction in latency compared to the reference model shows a marked

improvement in inference speed. Furthermore, with a compact model size of 11.52 KB (or 9.92 KB when zipped), we have effectively optimized the deployment, achieving a streamlined and efficient framework. Collectively, these findings reflect a commendable equilibrium between accuracy and efficiency, showcasing the model's capability to achieve high accuracy with reduced complexity.