LOW-MEMORY CONVOLUTIONAL NEURAL NETWORKS FOR ACOUSTIC SCENE CLASSIFICATION

Technical Report

Paulo Lopez-Meyer[1], *Juan A. del Hoyo Ontiveros*[1], *Hong Lu*[2], *Hector Corcourier*[1], *Georg Stemmer*[3], *Lama Nachman*[2], *Jonathan Huang*[4]

[1] Intel Corp, Intel Labs, Av. Del Bosque 1001, Zapopan, JAL, 45019, Mexico, {paulo.lopez.meyer, juan.antonio.del.hoyo.ontiveros, hector.a.cordourier.maruri}@intel.com
[2] Intel Corp, Intel Labs, 2200 Mission College Blvd., Santa Clara, CA 95054, USA, {hong.lu, lama.nachman}@intel.com
[3] Intel Corp, Intel Labs, Lilienthalstrasse 15, 85579, Neubiberg, Germany, georg.stemmer@intel.com
[4] Work done at Intel, jonathan.huang@ieee.org

ABSTRACT

In this work, we describe the implementation of four different convolutional neural networks for acoustic scene classification, complying with the memory size restrictions defined in the DCASE2020 Task 1b challenge guidelines. Quantization, pruning, knowledge distillation, and GCC-grams as input features, were explored as means to achieve the highest accuracy possible while reducing the number of resources in terms of the models trainable parameters and memory. Our experimental results yield to higher than the 87.30% reported accuracy in the challenge's baseline, where our four submissions managed to achieve $>$ 90.00% of acoustic classification accuracy using CNN models with $<$ 500 KB .

Index Terms— Acoustic Scene Classification, Low-Memory, Convolutional Neural Networks, End-to-End Audio Classification, Model Quantization, Model Pruning, Knowledge Distillation, GCC-grams.

1. INTRODUCTION

For the 2020 Detection and Classification of Acoustic Scenes and Events challenge (DCASE2020), acoustic data were provided to solve different acoustic related tasks. Task 1 refers to the challenge of building a model to classify different recordings into predefined classes corresponding to different urban environment scenes.

This challenge's dataset consists of 10-second audio recordings obtained in 10 different acoustic scenes from 12 major European cities, grouped in three major classes: indoor scenes, outdoor scenes, and transportation related scenes [1]. This acoustic dataset comprises binaural audio signals at 48 kHz of sampling rate in 24 bit resolution.

The challenge suggests the usage of a 1-fold arrangement for development as part of this task, i.e. 9,185 audio samples for training, and 4,185 for evaluation. Through the development stage of our implementations, we used Google Audioset data [2] to construct efficient audio embedding generators customized for three of our four implemented classification models.

2. METHODOLOGY

Following the guidelines provided by the challenge in the Task1 subtask b (Task 1b), we experimented with four low-memory implementations of convolutional neural network architectures (CNN) based on different techniques: FP32 to INT8 quantization, pruning of models using the lottery ticket approach, knowledge distillation from a large CNN to a smaller one, and the use of generalized cross-correlation with phase transformation (GCC-grams) as input features to a CNN. Two different CNN architectures were used as part of the four implementations above; the INT8 quantization and the pruning approaches are based on our end-to-end (e2e) AclNet that takes raw audio data as the input into two 1D convolutional layers followed by a 2D multi-layer CNN; the knowledge distillation and the GCC-gram approaches were implemented using typical CNNs. Pythorch was the framework of choice for our experimental setups.

In the following subsections, we describe in detail the experimentation followed around the four low-memory implementation mentioned above, as part of our submissions to the DCASE2020 Task 1b challenge.

2.1. INT8 quantization

AclNet is an e2e CNN architecture that takes raw time-domain input waveform, as opposed to more commonly used spectral features, e.g. Log-Mel filterbank or Mel-frequency cepstral coefficients (MFCC). One of the advantages of these types of e2e architectures is that the front-end feature makes no assumptions of the frequency response; its feature representation is learned in a data-driven manner, thus are optimized for the task at hand provided there are sufficient training data.

For this implementation, we conditioned the settings corresponding to the work described in [3], with a width multiplier of 0.5, and conventional depth-wise convolution layers. This architecture was pre-trained with Audioset [2] to generate a vector of 512 audio embeddings that are sent to a fully-connected layer classifier with ReLU activation functions in a transfer learning manner. Raw audio data downsampled at 16 kHz from the Task 1b dataset was

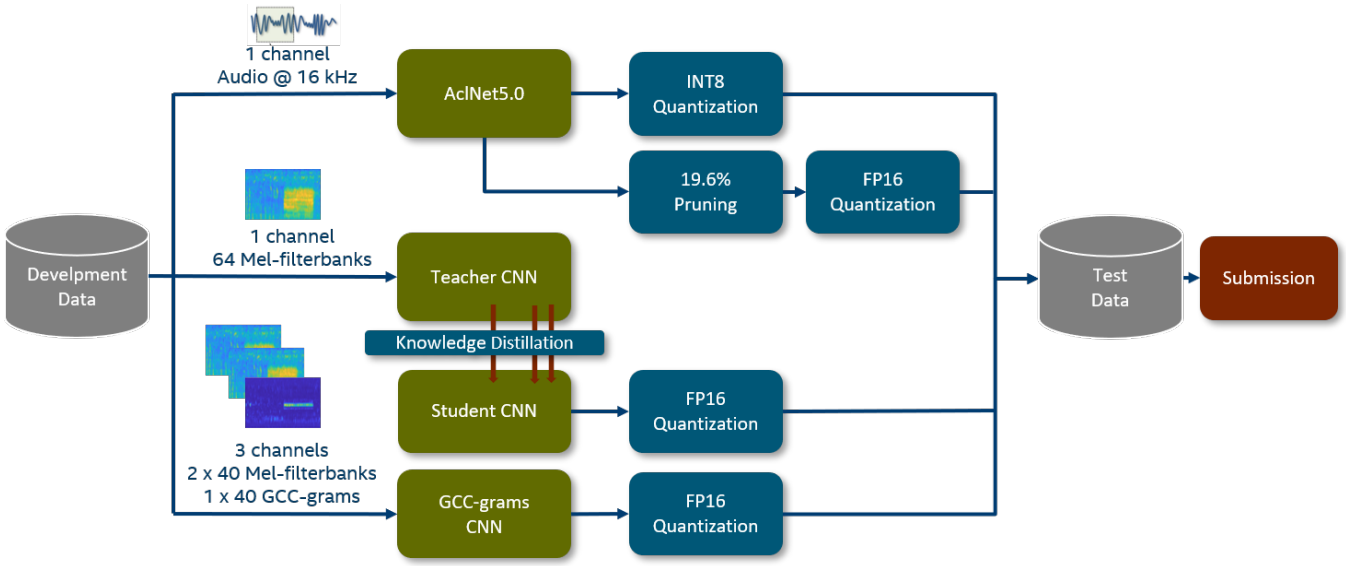

Figure 1: Development of our proposed implementations of four low-memory CNN architectures for acoustic scene classification in the DCASE2020 Task 1b challenge.

downsampled to 16 kHz and fed to the pre-trained AclNet, where the generated embeddings were used to train the classifier.

We performed a search for the optimal parameters of this e2e acoustic classification CNN model. We experimented with different values and configurations, that yield to the best performing models. Additionally, in order to increase the robustness of the training process, we also used different audio data augmentation techniques commonly used in audio processing, such as random noise addition, random cropping of 1-second of the audio signal, and random gain variation, together with the widely used mixup data augmentation technique [4]. During the training, acoustic data were randomly selected to form mini-batches of training clips. At evaluation time, we run the inference on 1-second non-overlapping consecutive audio segments, and then averaged the outputs over the length of the evaluation audio.

The resulting AclNet0.5 constitutes an FP32 base model with 317,038 trainable parameters, which yields into 1,238.43 KB of memory size, clearly above the 500 KB restriction in the challenge. In order to decrease the memory size of this model, we applied a straight FP32-to-INT8 quantization based on the methodology described in [5], through the use of the available tool accessible in [6], that results in a 309.60 KB CNN model.

2.2. Pruning based on the lottery ticket hypothesis

In this low-memory implementation, we went through the same training of an FP32 AclNet0.5 base model, exactly as described in the INT8 quantization described above, i.e. same handling of the data and training strategy . This AclNet0.5 base model, with 317,038 parameters is pruned at 19.6% in order to have a final model of 255,740 parameters that can be quantized from FP32-to-FP16 for a final 499.49 KB of memory size CNN model. The FP16 quantization used in all of our experiments were executed using the available functions from Pytorch.

As mentioned before, we pruned our AclNet0.5 base model through the lottery ticket hypothesis [7]. We initially trained our model to generate our base AclNet; after training, we removed 19.6% of the parameters by a typical pruning scheme, i.e. remove the parameters that are contributing less to the model's classification behavior. The lottery ticket hypothesis comes into place when, after identifying the post-pruning weights, a new training process is carried out with the original randomly initialized weights values assigned at the initial pre-training stage. This constitutes the spirit of the lottery ticket proposal, where subnetworks can be found in post-training pruning, that could reach evaluation accuracy comparable to the original network.

2.3. Knowledge distillation

We explored the use of knowledge distillation [8][9] from a large pre-trained Teacher CNN, into a significantly smaller Student CNN. For the development of both Teacher the Student, spectral based features were used; the acoustic data were processed to generate Log-Mel filterbank representations with 64 filter bands over a time window of 25 milliseconds and overlaps of 10 milliseconds, resulting in one Log-Mel filterbank channel as the input to the CNNs. Spec augment [10] was used a data augmentation process during training.

The Teacher CNN consists of a Vgg12 [11] with the exact same architecture as the one used in our submission for Task 1a; it has a total of 12 convolutional layers, with the first one having an output of 64 channels, and the last one is defined by 512 used to generate embeddings (pre-trained with Audioset [2]) that are needed during the knowledge distillation process. At the output of each convolutional layer, we apply batch normalization followed by ReLU activation. The output of the last convolutional layer is average pooled, to always produce a vector length of 512 values. This vector is then followed up by a fully connected layer to produce the 3-class output defined by the challenge's Task 1b. The Vgg12 CNN comprises 12.6 M trainable parameters, and 49 MB of memory.

The Student model is a small 3-layers CNN. Each convolutional layer is followed by batch normalization, ReLU activation functions, and max pooling. The output consists of a fully connected linear layer to perform the classification of the 3 acoustic scenes de-

fined in this task. The resulting Student model has 252,713 trainable parameters in 978.16 KB of memory with FP32 format.

Knowledge distillation between the Teacher and the Student CNNs is executed in training time by means of the following general loss function defined by:

$$loss = \alpha(E_loss) * \beta(S_loss) * \gamma(C_loss) \quad (1)$$

where E_loss represent the KL divergence loss between the 512 embeddings of the last convolutional layers of the Teacher and Student; S_loss is the KL divergence loss between the soft scores (softmax) at the output of both networks; and C_loss is the cross-entropy loss between the predicted and target labels. A search for the optimal α, β, and γ resulted in values of 0.25, 0.25, and 0.50 respectively. A final FP32 to FP16 quantization is performed to the Student model, resulting in a CNN model of size 493.58 KB.

2.4. Use of GCC-grams

Since the data provided for Task 1b are presented in a binaural manner, we propose the use of an additional feature based on the vector of the generalized cross-correlation with phase transform algorithm (GCC-PHAT), described in [12]. Such vector presents a delta-like response in which the maximum value has an offset from the center numerically equal to the amount of delay samples between the two signals, and it is normally used for sound source location. We segment the middle part of the vector, to generate time matrices we call GCC-grams (as an analogous name to spectrograms), which are synchronized with Log-Mel filterbank representations. The Log-Mel filterbanks were generated for both binaural channels downsampled to 16 kHz, with 40 filter bands over a time window of 64 milliseconds and overlaps of 45 milliseconds, resulting in one Log-Mel filterbank spectrogram matrrix per audio channel. When assembling all these representations together, the features constitute a 3 x 40 x 500 (channels x frequency x time) for each 10-sec audio clip, that are used for training and evaluation of a small depth-wise CNN. Our hypothesis is that GCC-grams, and therefore sound directivity information, can improve sound classification performance.

The 3-channel input CNN used in this implementation consists of a small 5 depth-wise convolutional layers with added batch normalization, ReLu activation functions, and maxpooling. At the output of the last convolutional layer, a linear fully connected layer is used to act as the 3-class classifier. The total number of trainable parameters for this CNN base model is 252,491, which results in a memory size of 986.29 KB in FP32 format. By applying an additional FP16 quantization, the resulting model presents a memory size of 493.15 KB.

3. RESULTS AND DISCUSSION

The experimental results obtained by our implementations are presented in the Tables 1, 2 and 3. Table 1 shows the performance of the base models over the Task 1b evaluation dataset, i.e. these models were trained in FP32 format to maximize results in the evaluation set. AclNet0.5 constitutes the base model for the INT8 quantization (AclNet0.5 INT8) and the lottery ticket prunning (AclNet0.5 LT) approaches; the Teacher CNN is used after training for knowledge distillation, and the GCC-grams CNN was initially designed to have less than 256,000 parameters for an efficient FP16 quantization. It is not surprising to see a significantly higher accuracy performance of the Teacher CNN as compared to the other base models, due to the high number of parameters.

Table 1: Experimental evaluation results obtained from the CNN base models

Model	Accuracy	Parameters	Memory KB
Baseline	87.30%	–	450.00
AclNet0.5	91.52%	317,038	1,238.43
Teacher CNN	94.60%	12,621,635	49,303.26
GCC-grams CNN	91.23%	252,491	986.29

Table 2: Experimental evaluation results obtained after the different low-memory implementations were executed on the CNN base models

Model	Accuracy	Parameters	Memory KB
AclNet0.5 INT8	90.94%	317,038	309.60
AclNet0.5 LT	91.47%	255,740	499.49
Student CNN	90.35%	252,712	493.58
GCC-grams CNN	91.23%	252,491	493.15

In Table 2, the experimental results obtained over the evaluation dataset are displayed with the low-memory implementations of our work, that constitute the four allowed submissions to the Task 1b challenge. It can be observed how all these optimized models achieve a higher performance than the baseline reported in the Task 1b guidelines, with at least 90.35% of acoustic scene classification, and with less than 500 KB, complying with the the challenge's submission restrictions.

Additional context metrics for comparison between the base models and the low-memory implementations are presented in Table 3. These results present some interesting insights. The GCC-grams CNN resulted in a good accuracy performance; this is surprising, since this model is at disadvantage as compared to the other base modes, by being trained from weights randomly initialized, i.e. no transfer learning used. Knowledge distillation resulted in the most significant memory size reduction, but also its accuracy performance gets impacted the most, were the lowest performance was obtained. The best accuracy was presented by the AclNet0.5 pruned and quantized to FP16; also, straight INT8 quantization over the AclNet0.5 base model seems to be an efficient approach, where the accuracy drop observed was less than 1.00%.

4. CONCLUSIONS

In this work, we present four different low-memory implementations of CNNs trained for acoustic scene classification as defined in

Table 3: Compression metrics used to compare the CNN base models with the low-memory implementations

Model	Reduction	Acc drop	Format
AclNet0.5 INT8	4.0X	0.58%	INT8
AclNet0.5 LT	2.5X	0.05%	FP16
Student CNN	99.9X	4.25%	FP16
CGG-grams CNN	2.0X	0.00%	FP16

the DCASE2020 Task 1b challenge. By exploring different methodologies to execute neural networks model optimization, e.g. transfer learning, knowledge distillation, pruning, and quantization, we were able to successfully construct CNN models that achieve $>$ 90% accuracy performance with less than 500 KB of memory size.

5. REFERENCES

[1] T. Heittola, A. Mesaros, and T. Virtanen, "TAU Urban Acoustic Scenes 2020 3Class, Evaluation dataset," Feb. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3685835

[2] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[3] J. J. Huang and J. J. A. Leanos, "Aclnet: efficient end-to-end audio classification CNN," *CoRR*, vol. abs/1811.06669, 2018. [Online]. Available: http://arxiv.org/abs/1811.06669

[4] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *CoRR*, vol. abs/1710.09412, 2017. [Online]. Available: http://arxiv.org/abs/1710.09412

[5] A. D. Kozlov, I. A. Lazarevich, V. Shamporov, N. Lyalyushkin, and Y. Gorbachev, "Neural network compression framework for fast model inference," *ArXiv*, vol. abs/2002.08679, 2020.

[6] "Neural network compression framework for pytorch (nncf)," 2020. [Online]. Available: https://https://github.com/openvinotoolkit/nncf_pytorch

[7] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Training pruned neural networks," *CoRR*, vol. abs/1803.03635, 2018. [Online]. Available: http://arxiv.org/abs/1803.03635

[8] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NIPS Deep Learning and Representation Learning Workshop*, 2015. [Online]. Available: http://arxiv.org/abs/1503.02531

[9] M. Phuong and C. Lampert, "Towards understanding knowledge distillation," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. Long Beach, California, USA: PMLR, 09–15 Jun 2019, pp. 5142–5151. [Online]. Available: http://proceedings.mlr.press/v97/phuong19a.html

[10] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *INTERSPEECH*, 2019.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[12] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 4, pp. 320–327, 1976.