LIGHTWEIGHT CONVOLUTIONAL NEURAL NETWORKS ON BINAURAL WAVEFORMS FOR LOW COMPLEXITY ACOUSTIC SCENE CLASSIFICATION

Nicolas Pajusco, Richard Huang, Nicolas Farrugia

IMT Atlantique, Lab-STICC, Department of Electronics, Brest - France
nicolas.farrugia@imt-atlantique.fr

ABSTRACT

In this paper, we investigate the feasibility of training low complexity convolutional neural networks directly from waveforms. While the vast majority of proposed approaches perform fixed feature extraction based on time-frequency representations such as spectrograms, we propose to fully exploit the information in waveforms directly and to minimize the model size. To do so, we train one dimensional Convolutional Neural Networks (1D-CNN) on raw, subsampled binaural audio waveforms, thus exploiting phase information within and across the two input channels. In addition, our approach relies heavily on data augmentation in the temporal domain. Finally, we apply iterative structured parameter pruning to remove the least important convolutional kernels, and perform weight quantization in floating point half precision. We apply this approach on the TAU Urban Acoustic Scenes 2020 3class dataset, with two network architectures : a 1D-CNN based on VGG-like blocks, as well as a ResNet architecture with 1D convolutions, and compare our results with the baseline model from the DCASE 2020 challenge, task 1 subtask B. We report four models that constitute our submission to the DCASE 2020 challenge, task 1 subtask B. Our results show that we can train, prune and quantify a small 1D-CNN model to make it 20 times smaller than the 500 KB challenge limit with an accuracy at baseline level (87.6 %), as well as a larger model achieving 91 % of accuracy while being 8 times smaller than the challenge limit. ResNets could be successfully trained, pruned and quantified in order to be below the 500 KB limit, achieving up to 91.2 % accuracy. We also report the stability of these results according to data augmentation and monoraul versus binaural inputs.

Index Terms— raw audio, Convolutional neural networks, auditory scene classification, residual networks, data augmentation, pruning

1. INTRODUCTION

Modern approaches for Deep Learning in computer vision or natural language understanding have been very successful in automatically learning flexible feature extractors, using convolutional neural networks [1], residual networks [2], or attention-based / transformer models [3]. Such feature extractors are trained using large amounts of data, limiting the need for hand-crafted features or representations. In contrast, most deep learning approaches for audio applications still rely on expertly defined, fixed feature extractors based on time-frequency representations, such as spectrograms or mel-spectrograms [4].

In this work, we were interested in testing the hypothesis that using a fixed feature extractor is detrimental for computational complexity, for two reasons. First, considering a spectrogram (or equivalent) as an image-like input may tend to overparametrize the downstream network, as the effort in training for classification becomes a two-dimensional problem. Second, a spectrogram only considers the power in frequency bands, ignoring the phase. In particular, when considering two channels as input, the phase difference between the channels could be informative. As a consequence, our goal is to show the feasibility to train low complexity networks, i.e. with significantly less parameters than using time-frequency feature extraction followed by a 2D CNN, using end-to-end learning, from feature extraction to classification, by training 1D-CNNs on raw waveforms.

Learning from raw waveforms is costly, due to the size of input vectors. In 2015, Sainath et al. [5] demonstrated that a raw waveform feature extracted with a convolutional layer matches the performance of the log-mel features when trained with more than 2 000 hours of speech. While approaches such as recurrent networks [6, 7, 8] or dilated convolutions [9] have previously been considered, such approaches need a very large number of parameters to be successful. SoundNet [10] is one notable example of successful training of a fully convolutional network on raw waveforms, and was trained on unlabeled videos using a teacher-student approach by distilling knowledge from vision networks. SoundNet demonstrated that it is feasible to train an efficient approach using raw waveforms, although two millions videos (corresponding to over one year of sound) were used for training, and the obtained model contains about 2.8 million parameters, for a size of approximately 11MB. Other approaches have successfully been trained on raw waveforms using knowledge distillation [11] from larger networks that were trained using log-mel feature [12, 13].

We suggest that it is possible to train a network for auditory scene classification using less than 50 hours of audio and less than 500 KB parameters. We tackle this problem by proposing an approach that relies on several techniques relevant in audio signal processing, as well as recent advances in deep learning training and compression techniques. We demonstrate the feasibility of this approach on the TAU Urban Acoustic Scences 2020 dataset, which consists in binaural recordings of urban soundscapes. First, we performed resampling after a careful examination of the dataset. Next, we use both input channels to train 1D-CNNs, ie with one-dimensional convolution kernels, coupled with stride and/or max-pooling to reduce the size of internal feature maps. Third, we use various strategies for data augmentation, in order to challenge the network the learn relevant audio features with degraded or masked versions of the waveforms. Our data augmentation strategies are inspired both by recent progress in deep learning in computer vision, as well as classical audio signal processing operations such as filtering. Finally, we apply parameter pruning, fine tuning and quantization to the best models obtained, in order to reduce the number of parameters and the memory footprint of the approach.

The rest of this paper is organized as follows. We begin by detailing our strategies for data augmentation during training in section 2. Next, we describe the two proposed network architectures in section 3. In section 4, we detail how we achieve to compress our models using structured parameter pruning and quantization. We explain our experimental and training setup in section 5, including an ablation study showing the separate effect of the various techniques we introduced. Finally we present and discuss our results in section 6.

2. DATA AUGMENTATION

The proposed approach relies heavily on data augmentation (DA), with the underlying hypothesis that combining various forms of DA can yield better flexibility with a smaller set of parameters, as well as better generalization. We use five forms of DA: temporal masking, filtering, noise addition, Mixup and CutMix [14]. The various hyperparameters of DA were chosen in preliminary analysis on subsets of the dataset. All DA are applied to 99% of the training set randomly at each epoch.

2.1. Temporal masking

Random crop using a rectangular window is an extremely common DA strategy for training 2D CNN in computer vision applications. We adapted this strategy to the temporal domain, by considering a temporal mask that is positioned randomly in the signal. We implement temporal masking by multiplying the signal by a rectangular window. The position of this window is randomly chosen within the total length of the signal, with a total of 1000 possible positions. The window length is randomly chosen according to a Gaussian distribution, with an average of four seconds and a standard deviation of one second. Importantly, the resulting signal after temporal masking is still 10 seconds long, which enables us to train and validate the network with the full signal length.

2.2. Filtering

We perform DA using filtering in order to augment the variety of the frequency content in the training set, thus challenging the network training to extract the most relevant frequency features when degrading the frequency content of the dataset. As a consequence, we apply eight finite impulse response filter (FIR): three low pass filters, three high pass and two band pass filters. These filters are applied after the temporal masking (if present) on the 10 second long signal. The cut-off frequencies of the low pass and high pass filter are respectively 300, 1000 and 2000 Hz. Two band pass filters are used : one with a bandwidth of 1200 to 3400 Hz, and one with a bandwidth of 340 to 3400 Hz.

2.3. Additive noise

The third DA strategy is to add white Gaussian noise into the signal. The signal to noise ratio is randomly chosen between 6 and 32 dB, by steps of 1 dB. Noise is added in the signal after temporal masking and filtering.

2.4. CutMix

CutMix [14] has been previously introduced as a very efficient DA strategy for training CNNs for computer vision task. The general idea of CutMix is to produce a new sample by concatenating two segments belonging to two different categories, and set the target by weighting according to lengths of each segment.

2.5. Mixup

Mixup [15] is another DA strategy that is similar to cutmix. Instead of adding two portions of signal from two different categories with a mask, mixup computes an weighted average of the signals across categories, with the same weights given to the targets.

3. NETWORK ARCHITECTURE

In this section, we present four models that were submitted to the DCASE 2020 Task 1 subtask B challenge. These architectures are inspired by popular CNN, namely VGG [1], and ResNet [2]. We adapt these architectures for raw binaural waveforms.

3.1. 1D-CNN

The first architecture is a one dimensional small standard CNN (models A and B in table 1 and figure 1) composed of successive blocks, each including a sequence of a convolution, Batch norm, Rectified Linear Unit Activation (ReLu), and Max Pooling. These blocks are similar to the ones found in networks such as VGG [1]. The particularities of this architecture are (1) the two input channels to deal with binaural sound, (2) one-dimensional operators such as 1D convolutions, 1D max pooling and 1D average pool. Note that the first convolutional layer doesn't share parameters between the two input channels, but two sets of filters are learnt separetly for each channel. We propose two networks, detailed in figure 1.

3.2. ResNet

The second architecture we use is ResNet (models C and D in table 1). ResNet [2] is a very efficient architecture that enables to train very deep neural networks, which have established the state of art result in many computer vision tasks. The proposed 1d-ResNet is based on a basic block described in figure 2, including a shortcut that uses a 1x1 convolution, and convolutions with 3x1 kernels. The architecture that we present here is composed of a first convolutional layer (2 input channels Conv1d, 32 output channels, kernel of size 64, stride of 4) followed by three modules. Each module is made of respectively 3, 4 and 3 basic blocks for model C, and 3,3,3 basic blocks for model D. Convolutions within each module are composed of respectively 32, 64 and 128 feature maps. Note that no max pooling is used in ResNet, and strides of 4 are used in the first convolution of each group of basic block. Average pooling is performed before the final fully connected layer.

3.3. Baseline architecture

As this work was performed for the DCASE 2020 challenge, we compare our results with the baseline architecture proposed by the challenge [17]. The baseline system performs audio feature extraction by computing log mel-band energies with 40 bands and 40 ms frame (50% overlap). These features are fed to a 2D CNN with two convolutional layers, the first layer including 32 filters with 7x7 kernels, maxpool 5x5, ReLu, followed by a second layer including 64 filters with 7x7 kernels, maxpool 4x100, and 1 fully connected layer with 100 hidden neurons. For this model, inputs are sampled at 48 kHz, 24 bits. In order to have a fair comparison with our approach, we also consider a binaural version of the baseline model, with two

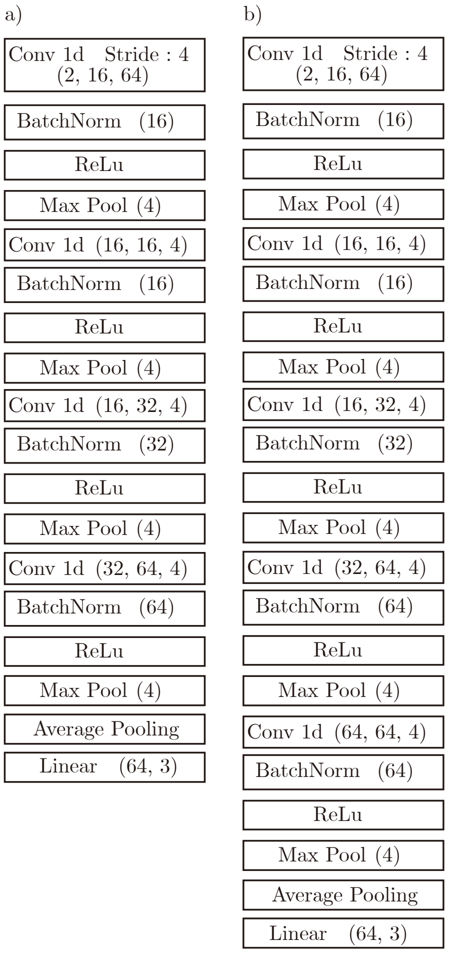

Figure 1: Two simple 1-dimensional CNN on raw waveform. Panel a corresponds to Model A and Panel b corresponds to Model B. For Conv1d modules, the numbers correspond respectively to input feature maps, output feature maps, and kernel size. For the Linear module, numbers correspond to number of inputs, number of outputs.

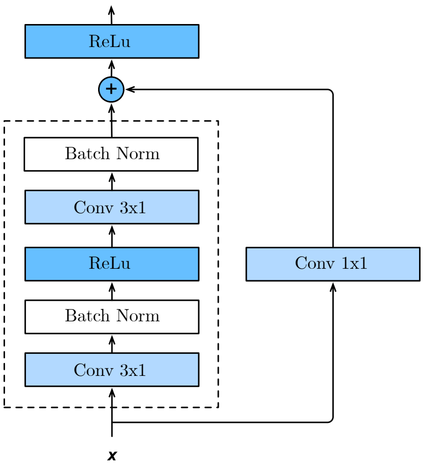

Figure 2: ResNet Block (figure adapted from [16]).

input channels in the first convolutional layer. We train the baseline using the same DA strategies applied on the waveforms before feature extraction, and we also tested 2D DA strategies applied on the mel-features (Random Crop, and 2D versions of Mixup and Cutmix).

4. MODEL COMPRESSION

4.1. Structured Pruning

Pruning of network parameters (ie setting parameters to zero) is a common technique used to decrease the number of non-zero parameters. We perform structured pruning, i.e. the parameters of whole convolution kernels or input features of linear modules are set to zero. The importance of convolution kernels is estimated using the L1 norm of its parameters, and the least important kernels are set to zero. Previous studies have shown that structured pruning can lead to high compression rates while keeping good performance on standard computer vision tasks [18, 19]. After the initial training of the model (see section 5.2), we perform pruning using an iterative approach based on fine-tuning (similar to [18]), as follows :

1. Ranking of convolution kernels' importance using the L1 norm.
2. Pruning of the least important ones by setting the corresponding parameters to zero (we used both 10 % and 20 % in our experiments).
3. Fine tuning of the pruned model on the training set dataset, with DA identical to initial training. We fix a relatively low learning rate (1e-5) and train the model using early stopping on validation set (see section 5.2).
4. Repeat from 1 until a stopping criterion is reached.

We use different stopping criterion for pruning : prune while the total number of nonzero parameters is above the challenge allowed maximum (model D), minimize parameter count while keeping an accuracy above the DCASE baseline of 87.3% (model A and C), or minimize parameter count while keeping an accuracy close to the full model (model B).

4.2. Quantization

After having performed training, pruning and fine-tuning iterations, our final step is the quantization of all model parameters. We quantize all inputs and parameters to floating point half precision, which uses 16 bits for each data. This level of precision enables to keep very similar test accuracy when compared with the full precision model, even slightly increasing the accuracy in some cases.

5. EXPERIMENTS

5.1. Datasets

The dataset for this task is based on TAU Urban Acoustic Scenes 2020 3Class [20, 17]. All samples were recorded from the same device in different sites (shopping mall, metro, bus, ...). There are three possible acoustic scene categories : transportation, indoor and outdoor. Each audio sample is 10 second long, binaural, sampled at 48 kHz in 24 bits precision. An extensive inspection of frequency content and frequency coherence of the development set has indicated that most of the signal energy is below 9000 Hz. Therefore, we resample all audio to 18 kHz in 16 bits precision, using

Model name	Accuracy	Loss	Total params	Non-zeros params	Size (KB)
Baseline	87.2	0.363	115441	115441	220.0
Model A	87.6	0.360	13632	12160	23.8
Model B	90.9	0.288	30080	29888	58.4
Model C	87.6	0.379	398400	130730	255.3
Model D	91.2	0.269	373696	238896	466.6

Table 1: Performance and model complexity of the models submitted to the challenge.

the Fourier method. We generate a validation set using 20% of the available training data, and use the remaining 80% as training set.

5.2. Training protocol

All models are trained using an Adam optimizer with a starting learning rate of 0.001 and a batch size of 64. We use a scheduler to divide the learning rate by 2 when the loss on the validation set does not improve during five epochs. The model with the best accuracy on the validation set is kept and tested on the test set. Model C and D are trained using only CutMix, while Model A and B are trained using Temporal masking, filtering and additive noise. For all models, the training protocol is perfomed in the following sequence:

- Training until early stopping as indicated by validation set performance,
- Iterative structured pruning on parameters and fine tuning, as described in section 4,
- Quantization to floating point half precision, and final evaluation on the test set.

5.3. Ablation study

We perform an ablation study in order to demonstrate the influence of our design choices. We perform a fine grain comparison between our most efficient model with the baseline model, by isolating the different DA strategies, as well as the effect of mono versus binaural input. For the mono case, we use an average of the two channels, as preliminary tests have shown that using either an average or a single channel out of the two yielded similar results.

6. RESULTS AND DISCUSSION

Table 1 presents the results obtained by the four models (as well as the official baseline) on the test split, as well as the parameter count (batch norm layers are not included). As our approach does not use fixed feature extraction (e.g. spectrogram), we included the whole model in the calculation of model parameters [1]. Our results show the feasibility of using raw binaural waveforms to train a model 20 times smaller than the 500 KB limit (model A), as well as a model achieving 90.9 % of accuracy, while being 8 times smaller than the challenge limit (model B). We also provide results for larger ResNet models approaching the 500 KB (model C and D).

Interestingly, when considering DCASE challenges, over 491 submissions on Acoustic scene classification from 2013 to 2020, there are 26 submissions that use raw waveforms as input features together with other types of features, and only 15 contributions using waveforms only. Considering datasets with binaural data since 2018, 11 submissions use binaural with raw-waveforms and spectrograms as input features. To the best of our knowledge, the present contribution is the first using binaural raw waveforms as sole input features, by submitting four such models to the DCASE 2020 Task 1 subtask B. In addition, Model B is ranked first when considering only models trained on waveforms, and has rank 48 overall.

DA	Baseline (Retrained)	Baseline (binaural)	Model B (mono)	Model B (binaural)
None	87.9 ±0.6%	89.5 ±0.5%	84.3 ±0.5%	89.2 ±0.3%
TM, FILT, Noise	87.8 ±0.6%	89.7 ±0.4%	85.7 ±0.4	90.9 ±0.4
CutMix	87.0 ±0.9%	90.2 ±0.5%	86.3 ±0.2%	91.1 ±0.2%
Mixup	87.2 ±0.7%	90.0 ±0.6%	85.1 ±0.3%	89.6 ±0.4%
2D random crop	88.2 ±0.6%	89.6 ±0.7%	-	-
2D cutmix	88.0 ±0.7%	90.3 ±0.7%	-	-
2D mixup	86.7 ±0.6%	90.2 ±0.3%	-	-

Table 2: Ablation study comparing DA strategies, mono versus binaural, between the baseline and model B (5 repetitions, average + 95% confidence interval). Our best result for Model B binaural + cutmix was obtained after the challenge deadline, thus the difference with table 1.

We perform the ablation study on Model B, which shows the best compromise between accuracy and model complexity (table 2). Binaural inputs leads to significant accuracy gains, both with raw waveform (+ 5.9%) or mel-features in the baseline model (+1.6%). However, the larger gain obtained for Model B may be explained by the lack of instantaneous phase information in log-mel features. DA strategies do not provide clear gains in accuracy for the baseline model (table 2). For 2D input, we test 2D DA such as cutmix, mixup and random crop [14]. When using raw waveform as inputs, the most efficient DA are CutMix (+2%) and temporal masking, filtering and noise together (+1.6%). Table 2 also shows that binaural inputs increase accuracy by 4.9% with raw-waveforms and 1.6% with log-mel. This suggests that binaural inputs are more beneficial on raw-waveforms, an hypothesis that could be challenged in future work by performing a comparison with Fourier features in 2D.

The proposed ResNet models coud not be trained very efficiently; model C achieves a performance slightly better than baseline (0.4 % increase) with about 12 % more parameters, and model D achieves an increase of 4 % with twice as much parameters than the baseline. Note that both ResNets could be extensively pruned, by zeroing about two thirds (resp. one third) of model C's parameters (resp. model D). This result that training large ResNets on raw audio on this task leads to over parametrization, which may be due to the small size of the dataset. Future studies could consider larger scale datasets to test whether performance of networks such as ResNet can be efficiently trained.

[1] Pytorch training and evaluation code is available here `https://github.com/brain-bzh/dcase-2020-task1-subtaskB`)

7. REFERENCES

[1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[4] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, *et al.*, "Cnn architectures for large-scale audio classification," in *2017 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE, 2017, pp. 131–135.

[5] T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals, "Learning the speech front-end with raw waveform cldnns," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[6] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, and Y. Bengio, "Samplernn: An unconditional end-to-end neural audio generation model," *arXiv preprint arXiv:1612.07837*, 2016.

[7] Y. Tokozume and T. Harada, "Learning environmental sounds with end-to-end convolutional neural network," 03 2017, pp. 2721–2725.

[8] Y. Tokozume, Y. Ushiku, and T. Harada, "Learning from between-class examples for deep sound recognition," *arXiv preprint arXiv:1711.10282*, 2017.

[9] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[10] Y. Aytar, C. Vondrick, and A. Torralba, "Soundnet: Learning sound representations from unlabeled video," in *Advances in neural information processing systems*, 2016, pp. 892–900.

[11] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[12] J.-w. Jung, H. Heo, H.-j. Shim, and H.-J. Yu, "Distilling the knowledge of specialist deep neural networks in acoustic scene classification," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, New York University, NY, USA, October 2019, pp. 114–118.

[13] J. Huang, H. Lu, P. Lopez Meyer, H. Cordourier, and J. Del Hoyo Ontiveros, "Acoustic scene classification using deep learning-based ensemble averaging," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, New York University, NY, USA, October 2019, pp. 94–98.

[14] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6023–6032.

[15] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[16] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, *Dive into Deep Learning*, 2020, https://d2l.ai.

[17] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in dcase 2020 challenge: generalization across devices and low complexity solutions," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, 2020, submitted. [Online]. Available: https://arxiv.org/abs/2005.14623

[18] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient convnets," *arXiv preprint arXiv:1608.08710*, 2016.

[19] J.-H. Luo, J. Wu, and W. Lin, "Thinet: A filter level pruning method for deep neural network compression," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5058–5066.

[20] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, November 2018, pp. 9–13. [Online]. Available: https://arxiv.org/abs/1807.09840