DCASE 2022 TASK 1: STRUCTURED FILTER PRUNING AND FEATURE SELECTION FOR LOW COMPLEXITY ACOUSTIC SCENE CLASSIFICATION

Technical Report

Lorenz P. Schmidt, Beran Kiliç, Nils Peters

Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany
International Audio Laboratories, Erlangen, Germany
{lopa.schmidt, beran.kilic, nils.peters}@fau.de

ABSTRACT

The DCASE challenge track 1 provides a dataset for Acoustic Scene Classification (ASC), a popular problem in machine learning. This years challenge shortens the provided audio clips to 1 sec, adds a Multiply-Accumulate operations (MAC) constrain and additionally counts all parameters of the model. We tackle the problem by using three approaches: First we use a linear model with global moments of the spectrogram, getting into reach of the baseline; then we use feature selection to reduce generalization gap and MACs; and finally, structured filter pruning to bring the number of parameters below the parameter constraint. Using the evaluation split of the development dataset, our result shows an increase to 49.1% overall accuracy compared to the baseline system with 42.9% accuracy.

Index Terms**—** ASC, structured pruning, quantization aware training

1. INTRODUCTION

The task of Acoustic Scene Classification (ASC) groups recordings into general scenes, such as ”airport”, ”bus”, ”metro”, etc. It helps applications to specialize for certain acoustical environments. For example the algorithms of hearing aids should work in a close conversation differently than in an outside walk.

The DCASE challenge task 1 provides a dataset for training and evaluation such models [1]. One prominent difficulty of the DCASE challenge is inbalance of per-device data. Recordings from three real devices (40h, 3h, 3h respectively) are mixed to provide six simulated devices (18h in total). The number of samples per-class is on the other hand balanced and equal for all ten classes.

This year Task1 a) increases the classification difficulty considerably [2]. The recording length is decreased from 10s to a single second. The baseline system dropped in accuracy from 47% to 42.9% and even as a human listener many recordings of classes are not distinguishable. Especially those which are silent or contain only low-frequency components are hard cases.

Furthermore the complexity constraints of the model are tightened. The maximum number of parameter constrain includes zero valued parameters (counting all of them) which makes unstructured pruning impossible. For the first time the Multiply-Accumulate counts (MACs) is constrained to 30MMACs - making the use of convolutional layer (which are naturally small in parameter count, but high in MACs) more difficult.

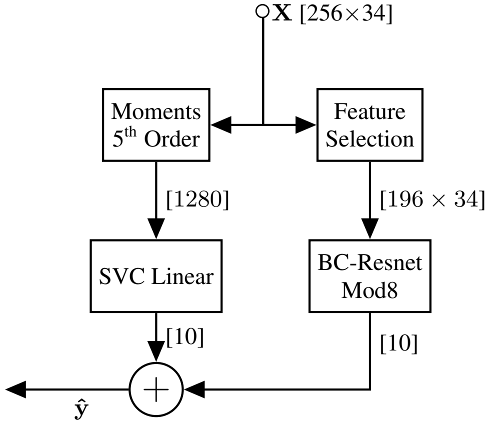


Figure 1: BC-Resnet model with feature selection and skip connection. $\mathbf{X}$ and $\hat{\mathbf{y}}$ are input features and class estimation, dimensions are depicted in brackets.

We opt for the best model of last year’s submission [3] and use a combination of feature selection and structured filter pruning to keep the model complexity in constraints. Section 2 describes the network architecture and compression techniques. Section 3 provides results and explains experiments. Finally we conclude our results in Section 4 and provide an outlook.

2. METHODS

We will introduce first the model architecture, a combination of last year’s winner with Support Vector Classifier (SVC) as a baseline, and then discuss feature selection, structured filter pruning and quantization respectively.

2.1. Model Architecture

Listening to audio samples from different classes shows that many samples contain temporally few information. We use therefore a combination of two models (see Figure 1) . First a linear model, classifying global features, and a BC-Resnet Mod-8 [4]. As input feature we use a log-mel filterbank with 256 bands. The linear model uses moments up to 5th order as input features and throws away temporal information. It is pre-trained with l_1 sparsity and selects approximately half of the input features. A Broadcast-Residual

The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg and Fraunhofer IIS.

network (BC-Resnet) [4] is used in tandem and improves prediction of the linear model. It can detect temporal-frequency patterns in the input spectrogram by using 1D/2D CNN layers. The BC-Resnet is in the same flavor as the last year's winning submission. It adds residual normalization to all layers [3]. We make two modifications: First, we add a bias to the last convolution layer. This makes adjustments to the learned SVC bias possible before normalizing the log probabilities. Second, we initialize weights and biases with zeros, ensuring that at the beginning of training the model predictions only depends on the linear part.

2.2. Feature Selection

After training the model to full accuracy, we conducted a second experiment selecting important features from the full feature set. The MACs are linear dependent on the input image size and subselecting rows in the spectrogram allows reducing the total MACs without making any model adjustments.

We setup the following optimization problem as

$$\min_{\mathbf{w},\theta} f_\theta(\mathbf{w} \odot \mathbf{X}) \quad \text{s.t.} \quad \text{card}(\mathbf{w}) < C, \tag{1}$$

where $\mathbf{w}$ is the feature mask along frequency axis, operation $\odot$ indicates multiplication per row and θ is the pre-trained model parameter set. We relax the constrain with the alternating direction method of multipliers (ADMM) [5] and use the hard constrain operator as solution to the proximal sub-problem [6].

The derived optimization routine is not guaranteed to converge because of the model architecture. We use three training phases to improve convergence, we pre-train our tandem model until convergence. Then we optimize for sparsity in the feature set by keeping the learning rate of the duals fixed and anhealing those of the primal variables (the model parameters). Finally we fine-tune the model by setting selected features to zero for a number of epochs. The accuracy of the final phase is reported in the results. A hyper-parameter search over both initial learning rates is required to find a good solution.

Together with the linear initialization in Section 2.1 we observed a much improved convergence of our model to full accuracy. Accidentally we used the masked feature only once, which resulted in a huge generalization gap (53% accuracy) and may indicate that our method works.

2.3. Structured Filter Pruning

This section describes how we apply structured filter pruning to the model of Section 2.1. The advantage of the BC-Resnet is that it only consists of convolutional layer, making the pruning technique simpler as we don't have to deal with other architectures.

The filter importance is measured as the max or Frobenius norm over all its entries

$$W_j^i = \|\mathbf{W}^i[j,k,:,:]\|_{\infty,\mathrm{F}}, \quad k \in \mathcal{K},$$

with i the layer index, j the output channel and $\mathcal{K}$ the set of reachable input indices from previous layers. In experiments we observed a more stable convergence for the Frobenius norm, so we used it in all our experiments.

We create a global, ordered list of filter importance (with all input assumed reachable) and perform binary search over the threshold until constrains are fulfilled. In each update step a reachability analysis of layers give the input channels $\mathcal{K}$ and filter importances are updated accordingly. The routine gives a filter importance threshold, which ensures that MACs and number of parameters are fulfilled. We ensure that at least one output channel is active, such that the network is not disconnected.

With the threshold a similar optimization problem as in Section 2.2 is setup and the same combination of dual gradient-ascent and hard-threshold operator is used to solve the problem. We also use the same scheduler and three phases for optimization.

2.4. Quantization

First we planed to combine the filter pruning with 8bit constraints [7], but because of time limitations we used quantization aware training (QAT) from the TensorRT library [8]. We apply QAT after feature selection and filter pruning and quantize all our 32bit floating points to 8bit integers.

3. EXPERIMENTS AND RESULTS

3.1. Training and testing dataset

In all our experiments we use the DCASE 2022 challenge task 1 [9] split providing 139619/29680/29680 samples for training/validation/testing. Each sample belongs to one of 12 European cities, 3 real (A, B, C) or 6 simulated devices (S1-S6) and to one of 10 different acoustic scenes. Those are "airport", "bus", "metro", "metro station", "park", "public square", "shopping mall", "street pedestrian", "street traffic", "tram". The simulated devices S1-S6 are generated by using measured impulse responses and applying range compression to recordings of device A. Each sample has a length of 1sec and a sampling rate of 44 1kHz.

3.2. Feature extraction

We downsample the input signal to 16kHz and use a Mel filterbank for the feature extraction. The Mel spectrogram has window length of 130ms, overlap of 30ms and 256 Mel bands. We apply a logarithmic transformation to the filterbank output. For the moments we use mean, variance, skewness, kurtosis and hyperskewness.

3.3. Training details

The input features are augmented to generalize the model. We use a random roll of 40% of the signal length. We also use Specaugment [10] in frequency domain with mask parameter of 20 and Mixup [11] with $\alpha = 0.2$. We apply stochastic gradient descent (SGD) to the model and train for 80 epochs. The learning rate is increased to 0.035 in a warmup phase of 3 epochs and then decreased to 0.00035 in a period of 77 epochs. We use momentum of 0.9, weight decay of 0.001 and a mini-batch size of 64.

For the ADMM optimization we use 5 update epochs and 2 finetuning epochs. For the update step we initialize a linear SGD with learning rate of 0.001 and decrease it gradually to 0.0001 duing the 7 optimization epochs.

3.4. Results

We observe multiple outcomes from our results. The linear model decreases accuracy by around 2.7% while using 2282 times less MACs and four times less parameters (see Table 2). Especially for classes like "park", "shopping mall" and "street traffic" the linear model is on-par or even out performs the tandem model (see Figure

	A	B	C	S1	S2	S3	S4	S5	S6	Overall
Linear	54.58	44.15	44.42	45.42	45.12	46.70	38.25	41.79	38.52	39.90
Feature Sel.	72.36	61.44	64.54	62.93	59.39	60.10	47.10	47.02	40.34	51.52
Pruned	71.95	58.24	63.19	60.13	58.89	60.07	46.77	47.66	40.67	50.76
Pruned+Quant.	70.54	55.68	60.30	57.95	55.99	58.05	44.48	47.46	40.13	49.06

Table 1: Top 1 accuracy (%) on the test split of the DCASE 2022 task1 dataset, A,B,C are real recordings and those of S1-S6 are simulated

2). This may be due to few temporal features (like in a park) or easy distinguishable global frequency distributions (like cars on a street). When using the BC-Resnet Mod-8 [4] from last year, the accuracy for the new dataset shows that the task at hand is much harder. Especially the reduced recording length of 1sec makes inference about the acoustical scene in many cases difficult. The feature selection reduces the total number of MACs and we have seen no decrease in accuracy when masking out 64 features. Applying structured filter pruning works well in our scenario, as seen in Table 2 and decreases performance only by 0.76%. This may also be due to using a linear model in tandem and reducing the number of input features. For the device accuracy distribution we get similar results from last year. The real device A, which has a larger portion of the samples, has a much better accuracy when compared to other devices (see Table 1). On the other hand, devices A4-A6, which are not in the training dataset, have the worst accuracy during testing. The QAT worked not as well as hoped for and reduced accuracy by another 1.7 % (as seen in Table 2). We tried out quantization of Nemo [12] and torch native quantization routines [13], but only TensorRt [8] gave us viable results. It may be interesting to see whether combined pruning and quantization constraints can improve that performance.

We submit our best model to the DCASE challenge with feature selection, pruning and quantization combined. It gives a total accuracy of 49.06% with 127.84k params and 17.383 MMACs in 8bit depth for weights, as well as activation functions.

	Params	MMACSs	Bits	Acc (%)	LogLoss
Baseline [2]	47k	29.234	8bit	42.90	1.575
Linear	12.81k	0.01281	32bit	39.90	1.858
Pruned	127.84k	17.383	32bit	50.76	1.521
Pruned+Quant.	127.84k	17.383	8bit	49.06	1.565

Table 2: Parameter count and MACs for each model. We used the Pruned+Quant model as our final submission.

4. CONCLUSION

Our submission for the DCASE challenge 2022 extends the model architecture and feature extraction of the winner in DCASE 2021. We modified and developed an optimizer for structured pruning and feature selection to fit the new challenge constraint. We applied methods from convex optimization, such as ADMM [5] and proximal operators [6], in context of DNNs to perform pruning and feature selection. Structured filter pruning slightly reduced the model accuracy by 0.76%. The final model requires only 17.38 MMACS and 127.84k parameters. It achieves an total accuracy of 49.06% on the testing split of the DCASE challenge 2022 task 1 [9] and thus, outperforms the baseline system in accuracy and total MACs.

5. REFERENCES

[1] 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, pp. 56–60. [Online]. Available: https://arxiv.org/abs/2005.14623

[2] I. Martín-Morató, F. Paissan, A. Ancilotto, T. Heittola, A. Mesaros, E. Farella, A. Brutti, and T. Virtanen, “Low-complexity acoustic scene classification in DCASE 2022 challenge,” 2022. [Online]. Available: https://arxiv.org/abs/2206.03835

[3] B. Kim, S. Yang, J. Kim, and S. Chang, “QTI submission to DCASE 2021: Residual normalization for device-imbalanced acoustic scene classification with efficient design,” DCASE2021 Challenge, Tech. Rep., June 2021.

[4] B. Kim, S. Chang, J. Lee, and D. Sung, “Broadcasted residual learning for efficient keyword spotting,” 2021. [Online]. Available: https://arxiv.org/abs/2106.04140

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[6] N. Parikh, S. Boyd, *et al.*, “Proximal algorithms,” *Foundations and trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[7] C. Leng, H. Li, S. Zhu, and R. Jin, “Extremely low bit neural network: Squeeze the last bit out with ADMM,” 2017. [Online]. Available: https://arxiv.org/abs/1707.09870

[8] Nvidia, “Nvidia/tensorrt: Tensorrt is a C++ library for high performance inference on nvidia GPUs and deep learning accelerators.” [Online]. Available: https://github.com/NVIDIA/TensorRT

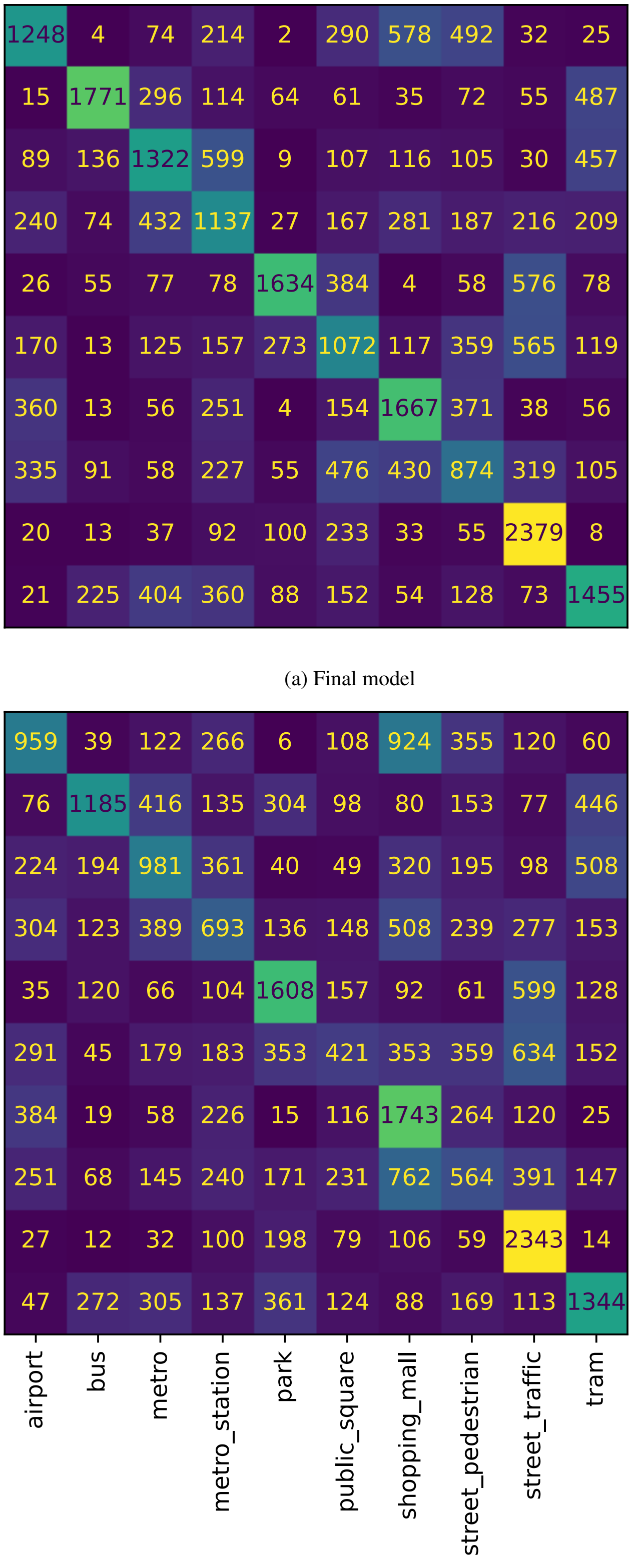
[9] http://dcase.community/challenge2022/.

[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*. ISCA, sep 2019. [Online]. Available: https://doi.org/10.21437%2Finterspeech.2019-2680

[11] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” 2017. [Online]. Available: https://arxiv.org/abs/1710.09412

[12] F. Conti, “Technical report: Nemo dnn quantization for deployment model,” 2020. [Online]. Available: https://arxiv.org/abs/2004.05930

[13] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito,



(a) Final model

(b) Linear model

Figure 2: Confusion matrices for ASC ten class problem on the testing dataset of DCASE 2022 task 1. The final submission is on the top and a linear model on the bottom. Classes are only shown on the X axis to improve readability.

M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf