FREQUENCY-AWARE CNN FOR OPEN SET ACOUSTIC SCENE CLASSIFICATION

Technical Report

Alexander Rakowski, Michał Kośmider

Samsung R&D Poland, Audio Intelligence Dept., Warsaw, Poland, {a.rakowski2, m.kosmider}@samsung.com

ABSTRACT

This report describes systems used for Task 1c of the DCASE 2019 Challenge - Open Set Acoustic Scene Classification. The main system consists of a 5-layer convolutional neural network which preserves the location of features on the frequency axis. This is in contrast to the standard approach where global pooling is applied along the frequency-related dimension. Additionally the main system is combined with an ensemble of calibrated neural networks in order to improve generalization.

Index Terms— convolutional neural network, frequency aware, ensembling

1. INTRODUCTION

Task 1C of the DCASE 2019 Challenge [1] extends the original task (1A) with the problem of detecting samples not belonging to any of the 10 known classes, termed *open set classification*. Samples of the "known" classes are taken from the TAU Urban Acoustic Scenes 2019 Openset development dataset [2], while the out-of-distribution samples are extracted from the TUT Acoustic scenes 2017 dataset [3].

2. DATA PREPROCESSING

2.1. Main system

Input features for the main model are obtained by computing 64-bin log-mel spectrograms of the audio recordings, resampled to 32 kHz, with a window size of 1024 frames and hop size of 500. This results in data of shape $H \times W$, where $H = 64$ is the number of frequency bins and $W = 640$ is the number of spectrogram frames. Before being fed to the network the inputs are standardized using the mean and standard deviation computed across all samples from the training set. Note that these statistics are computed for each frequency bin.

2.2. Secondary system

Similar to the main system log-mel spectrograms were used as input features for the calibrated CNNs, with 256 mel bins, a window size of 2048 and hop size of 512. Frequency bin values were standardized in the same manner as in the main system.

3. MODELS

3.1. Frequency-aware CNN (*main system*)

The architecture of the main model is based on the 5-layer CNN from [4]. It consists of several convolution blocks which extract spectro-temporal feature maps from the input. These maps are then aggregated using a global max pooling operation yielding a feature representation for the whole recording, which is then processed by the final fully-connected layer with a sigmoid activation. Structure of a single convolution block is shown in Figure 1.

Contrary to the original system global pooling is applied only across the temporal dimension. The frequency-related dimension is instead merged with the channel dimension. More formally, an input sample x of shape $1 \times F \times T$ is passed through the convolution blocks, resulting in a feature map of shape $C \times F' \times T'$, where C is the number of filters in the last convolution block and F' and T' are the sizes of the spectral and temporal dimensions respectively, after being reduced due to the pooling operations inside each block. Merging the channel and spectral dimensions results in a tensor of shape $C \cdot F' \times 1 \times T'$. Finally, applying global max pooling over the temporal dimension reduces the tensor to a shape of $C \cdot F' \times 1 \times 1$.

In the domain of image processing global pooling is often applied over both spatial dimensions, motivated by the assumption that when performing object classification the obtained features should be translation-invariant[1] However this is not necessarily desirable when handling spectro-temporal data like spectrograms - even though they are processed using the same methods (two-dimensional convolutions). Whether certain pattern are detected in lower or in higher frequency bands might be of importance for classification.

Final predictions are obtained in the same manner as in [4]. If none of the probabilities assigned to each of the 10 known classes is higher than 0.5, the sample is treated as the *unknown* class. Otherwise the class with the highest probability is returned as the prediction.

Because of this approach it is desirable for the model not to yield overly confident predictions on examples considerably different from those seen during training, so that the out-of-distribution samples will be correctly recognized at test time. In order to achieve this Manifold Mixup [5] is employed (with $\alpha = 0.5$), which forces the model to have smoother predictions on hidden state interpolations of different examples. It is also argued that this technique has a regularizing effect on the network. In fact, models trained with Manifold Mixup generalized better than those trained with dropout (both standard and spatial dropout [6]), while combining these two techniques together did not yield better results.

Two variants of this model are used for the final submission, differing in number of layers and convolutional filters used. Their configurations are shown in Table 1.

[1] Unless, naturally, an additional task such as object localization is being solved.

Filters	Pooling
Variant 1	
64	2×2
128	2×2
256	2×2
512	-
Variant 2	
32	2×2
64	2×2
128	2×2
192	-
384	2×2

Table 1: Layer configurations of the two variants of the main system

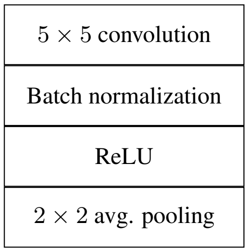

Figure 1: Structure of a convolution block used in the main system

3.2. Calibrated CNNs (*secondary system*)

The secondary system is an ensemble of 15 separately trained convolutional neural networks. Architecture of these networks is shown in Table 2. System predictions are obtained using calibrated soft voting. The calibration is done for each model, with a randomly created validation set, using isotonic regression for each class. Finally, these predictions are further aggregated by using "plain" soft voting (without further calibration) with two of the main models.

The following data augmentation techniques are used when training these models. Mixup [8] with $\alpha = 0.2$ is applied on 30% of the training samples. Additionally, following SpecAugment [9], blocks of time are randomly zeroed out with maximum width of 80, random frequency bands are zeroed out with maximum height of 27 and time warping is applied with a width of 40 frames.

Layer	Channels	Kernel	Stride
Conv2D+ReLU+BN	16	3	1
Conv2D+ReLU+BN	32	3	2
Conv2D+ReLU+BN	32	3	1
Conv2D+ReLU+BN	64	3	2
Conv2D+ReLU+BN	64	3	1
Global average pooling	64	-	-
Fully-connected	10	-	-

Table 2: Architecture of a single network of the calibrated ensemble. *BN* stands for Batch Normalization [7].

Model	Accuracy [%]
Main model 1	57.5
Main model 2	57.5
Ensemble of 1 and 2	53.5
Ensemble of 1, 2 and the calibrated models	55.3

Table 3: Results of the submitted systems on the public leaderboard

4. EXPERIMENTS AND RESULTS

4.1. Setups

4.1.1. Main system

The two variants of the main system are trained for 10,000 iterations on mini-batches of size 32, using the AMSGrad [10] variant of the Adam [11] optimizer, with an initial learning rate of $1\mathrm{e}{-3}$. After each 200 iterations the learning rate is decayed by a factor of 0.9.

4.1.2. Secondary system

The calibrated ensemble is trained for 50,000 iterations using the Ada-delta [12] optimizer, on mini-batches of size 64. Learning rate is reduced by half if accuracy does not improve by at least 10^{-3} for 16 epochs, starting at 0.5. Focal loss [13] with $\gamma = 1$ and no class weights is used as the loss function. Additionally, an ℓ_2 norm regularization term on model parameters is added with a weight of 10^{-5}.

4.2. Results

Results of each of the 4 submitted systems: two variants of the main model, their ensemble, and an ensemble of the two with the calibrated models are shown in Table 3. Note that the two variants of the main system obtained an identical score. Additionally, their ensemble performs worse than each of the single models, indicating that their decisions are not complementary. However, while still performing worse than the single models, adding the calibrated models to the ensemble improves the aggregated score, probably due to bigger differences in architecture and training procedures.

5. CONCLUSIONS

The main system presented in this report introduces a *frequency-aware* CNN architecture for the task of open set acoustic scene classification. This modification to the standard approach of handling spectrogram-like data preserves a coarse information about the position of detected patterns in the frequency-related dimensions. The models are trained using Manifold Mixup in order to force their predictions to be less certain for out-of-distribution data. Along two variants of the proposed CNN model, two ensembled predictions are submitted, one being a simple average of the two single models, and the other one including additionally a bigger "sub-ensemble" system of 15 calibrated CNNs. However results on the public leaderboard seem to indicate better perfromance of just the single models.

6. REFERENCES

[1] http://dcase.community/challenge2019/.

[2] 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

[3] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, “Dcase 2017 challenge setup: Tasks, datasets and baseline system,” in *DCASE 2017-Workshop on Detection and Classification of Acoustic Scenes and Events*, 2017.

[4] Q. Kong, Y. Cao, T. Iqbal, Y. Xu, W. Wang, and M. D. Plumbley, “Cross-task learning for audio tagging, sound event detection and spatial localization: DCASE 2019 baseline systems,” *CoRR*, vol. abs/1904.03476, 2019. [Online]. Available: http://arxiv.org/abs/1904.03476

[5] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, and Y. Bengio, “Manifold mixup: Better representations by interpolating hidden states,” in *International Conference on Machine Learning*, 2019, pp. 6438–6447.

[6] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, “Efficient object localization using convolutional networks,” *CoRR*, vol. abs/1411.4280, 2014. [Online]. Available: http://arxiv.org/abs/1411.4280

[7] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[8] 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

[9] 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,” *arXiv preprint arXiv:1904.08779*, 2019.

[10] S. J. Reddi, S. Kale, and S. Kumar, “On the convergence of adam and beyond,” *CoRR*, vol. abs/1904.09237, 2019. [Online]. Available: http://arxiv.org/abs/1904.09237

[11] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[12] M. D. Zeiler, “ADADELTA: an adaptive learning rate method,” *CoRR*, vol. abs/1212.5701, 2012. [Online]. Available: http://arxiv.org/abs/1212.5701

[13] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” *CoRR*, vol. abs/1708.02002, 2017. [Online]. Available: http://arxiv.org/abs/1708.02002