ARBORESCENT NEURAL NETWORK ARCHITECTURES FOR SOUND EVENT DETECTION AND LOCALIZATION

Technical Report

Daniel Krause

AGH University of Science and Technology
Department of Electronics, Al. Mickiewicza 30
Krakow, 30-059, Poland
danielkrause2h@gmail.com

Konrad Kowalczyk

AGH University of Science and Technology
Department of Electronics, Al. Mickiewicza 30
Krakow, 30-059, Poland
konrad.kowalczyk@agh.edu.pl

ABSTRACT

This paper describes our contribution to the task of sound event localization and detection (SELD) using first-order ambisonic signals at the Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge 2019. Our approach is based on arborescent convolutional recurrent neural networks with the aim to achieve joint localization and detection of overlapping acoustic events. Three submitted systems can be briefly summarized as follows. System 1 splits the neural network into two branches associated with localization and detection tasks. This splitting is performed directly after the first convolutional layer. System 2 utilizes depthwise separable convolutions in order to exploit interchannel dependencies whilst substantially reducing the model complexity. System 3 exhibits a tree-like architecture in which relations between the channels for phase and magnitude are exploited independently in two branches, and they are concatenated before the recurrent layers. Finally, System 4 is based on score fusion of the first two systems.

Index Terms—acoustic event detection, source localization, convolutional recurrent networks, DCASE2019

1. INTRODUCTION

Sound event detection (SED) is one of the most important tasks in current audio research. Automatization of the process of detecting and classifying signals present in an acoustic environment is of interest in numerous applications. Robots can utilize environmental information for improved behavior [1], surveillance systems can be significantly enhanced by detecting audio associated with hazardous events [2]. Multimedia annotation and criminology analyses can be accelerated by automatic sound event recognition [3]. Many of these practical applications would benefit from obtaining additional information about the location of the detected sound event. Therefore combining sound event detection and localization (SELD) tasks is a natural next step in the field of machine learning driven audio research. In this technical report we describe our contribution to the SELD task provided by the DCASE Challenge 2019 organizers [4]. Our method is based on convolutional recurrent neural networks (CRNNs) and utilize first-order ambisonic recordings.

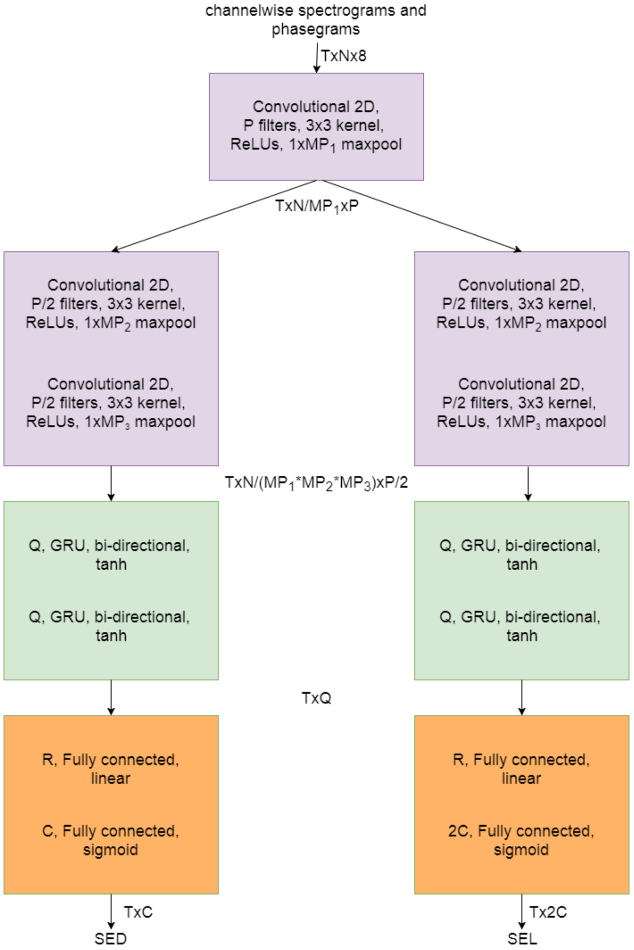
channelwise spectrograms and phasegrams
TxNx8
Convolutional 2D, P filters, 3x3 kernel, ReLUs, 1xMP$_1$ maxpool
TxN/MP$_1$xP
Convolutional 2D, P/2 filters, 3x3 kernel, ReLUs, 1xMP$_2$ maxpool
Convolutional 2D, P/2 filters, 3x3 kernel, ReLUs, 1xMP$_3$ maxpool
Convolutional 2D, P/2 filters, 3x3 kernel, ReLUs, 1xMP$_2$ maxpool
Convolutional 2D, P/2 filters, 3x3 kernel, ReLUs, 1xMP$_3$ maxpool
TxN/(MP$_1$*MP$_2$*MP$_3$)xP/2
Q, GRU, bi-directional, tanh
Q, GRU, bi-directional, tanh
Q, GRU, bi-directional, tanh
Q, GRU, bi-directional, tanh
TxQ
R, Fully connected, linear
C, Fully connected, sigmoid
R, Fully connected, linear
2C, Fully connected, sigmoid
TxC
SED
Tx2C
SEL

Figure 1: Overall architecture of both Lambda and LambdaDep.

2. MODELS

Similarly to [5], we use the Short-Time Fourier Transform (STFT) to extract magnitude and phase information, respectively for each of the four ambisonic channels, resulting altogether in 8 feature maps. These spectrograms and phasegrams are obtained using a 20ms long Hamming window for framing with the DFT length of N. Depending on the model, these channels are processed by the convolutional layers in different ways.

System 1 and System 2 share the same model structure depicted in Figure 1. In both cases we utilize a CRNN architecture, which consists of three convolutional layers (CNN), two recurrent layers (RNN) and two fully connected layers (FC). In System 1 (called Lambda) spectrograms and phasegrams used as eight separate channels are fed to the standard 2D convolutional layers. Each CNN block produces feature maps utilizing a 3x3 kernel per channel, whereas the outputs are processed using batch normalization and rectified linear unit (ReLU) activations. The dimensionality of the feature maps is being reduced using maxpooling (MP$_i$) along the frequency axis, with the sequence length T left unchanged. In order to learn strong task-dependent features, the network is split into two branches right after the first layer. While the first branch is responsible for the SED task, the second performs multi-output regression to obtain directions-of-arrival (DOAs). While the first convolutional block uses P filters, the following layers consist of P/2 filters to keep the model complexity fairly similar.

In both branches, the outputs of convolutional blocks are fed into the RNN layers to learn additional temporal context information. In order to utilize both past and future information, we use bi-directional gated recurrent units (GRU), each consisting of Q nodes, followed by tanh activation functions. Finally, the model outputs in both branches are produced by fully connected layers. The first FC layer contains R neurons with no specific activation function (i.e. linear). To obtain probability values for each of the detected classes, the last layer in the SED branch utilizes C nodes, followed by a sigmoid activation function. As the DOA branch performs a regression task, it produces 2C linear outputs.

System 2 (hereafter referred to as LambdaDep) is described by the same block diagram as System 1. The main difference is that depthwise separable convolutional layers (DSCNN) were used instead of traditional CNNs. Standard CNNs combine interchannel information by simply summing the convolution outputs from each separate channel. On the other hand, DSCNNs split the convolution process into two parts [6]. Firstly, a single depthwise convolution is applied to each channel separately, keeping the depth dimension the same. Next, P 1x1x8 kernels are used to mix the interchannel information and lower the dimension size to one. This enables efficient multichannel information extraction in conjunction with a significant complexity reduction.

System 3 (called X-tree) introduces a different concept of an arborescent neural network architecture which is depicted in Figure 2. In aforementioned models (i.e. System 1 and 2) the channels were mixed from the very first layer. In contrast, here we present an idea of learning channel-dependent features with respect to the magnitude and phase separately. In the first layer, each channel is processed by a separate CNN block using P/8 filters. Moreover, spectrograms and phasegrams are kept in completely separate trees, each containing independent branches for all channels. In the second step, outputs from the previous layer are concatenated for each two adjacent branches and processed by convolutional blocks utilizing P/4 filters. The third layer repeats this scheme by applying P/2 filters in each block. This way the network independently learns magnitude and phase features, which are further concatenated. Finally, the network is split into two task-dependent branches, just like in the Lambda system, starting with the GRU layers.

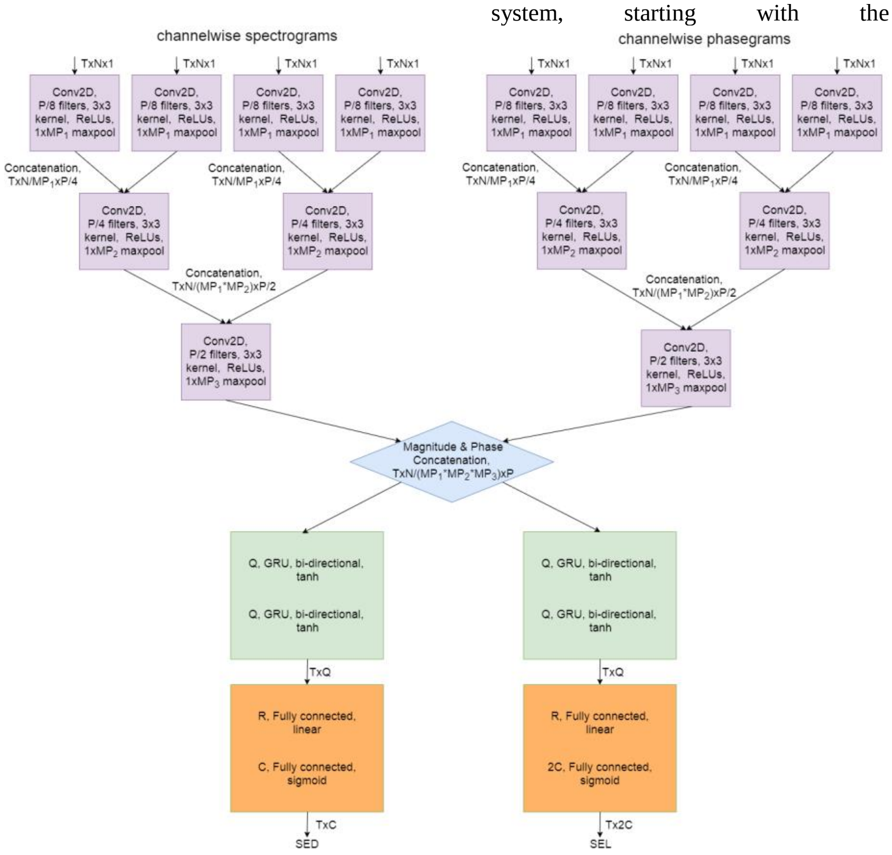
channelwise spectrograms
channelwise phasegrams
TxNx1
Conv2D, P/8 filters, 3x3 kernel, ReLUs, 1xMP$_1$ maxpool
Concatenation, TxN/MP$_1$xP/4
Conv2D, P/4 filters, 3x3 kernel, ReLUs, 1xMP$_2$ maxpool
Concatenation, TxN/(MP$_1$*MP$_2$)xP/2
Conv2D, P/2 filters, 3x3 kernel, ReLUs, 1xMP$_3$ maxpool
Magnitude & Phase Concatenation, TxN/(MP$_1$*MP$_2$*MP$_3$)xP
Q, GRU, bi-directional, tanh
Q, GRU, bi-directional, tanh
TxQ
R, Fully connected, linear
C, Fully connected, sigmoid
2C, Fully connected, sigmoid
TxC
SED
Tx2C
SEL

Figure 2: X-tree architecture.

Additionally, a fourth system (LambdaFus) was submitted. It can be described as a straightforward fusion of the first two systems trained separately, combining the detection and regression results by an arithmetic mean. The parameters used for all systems are as follows: P=64, MP={8,8,4}, Q=128, R=128 N=1024. Table 1 summarizes model complexity of the submitted systems. The Lambda, LambdaDep and X-tree models are characterized by a much smaller complexity compared with the baseline system. In particular, the first two systems use a total number of parameters two times less than the baseline system.

Table 1: Total number of parameters used in the submitted models compared with the baseline system.

System	Name	Model parameters
0	Baseline	613,537
1	Lambda	333,537
2	LambdaDep	282,089
3	X-tree	429,857
4	LambaFus	615,626

3. EXPERIMENTS

Systems are evaluated using separate metrics for SED and DOA estimation. We measure the SED performance using F_1 score and Error Rate (ER) [7], while DOA error and frame recall evaluate the localization performance [8]. Moreover SELD score is obtained using the following formula:

$$SELD = \frac{[(1 - F_1) + ER + \frac{DOA\ error}{180°} + (1 - Recall)]}{4} \quad (1)$$

All models are trained with the Adam optimizer in the cross-validation step for 200 epochs. Training is stopped after 30 epochs of no improvement in both SELD score and training loss. For the evaluation, models are trained for 300 epochs with 50 patience. The SED output uses the binary cross-entropy loss function, whereas the log-cosh loss function is used for DOA estimation. Experiments are performed using the Keras and Theano libraries, the code is based upon the baseline system provided by the organizers [5]. Results obtained over 4 cross-validation splits are presented in Table 2.

Table 2: Cross-validation results for all submitted systems.

	SED		SEL		SELD
	ER	F1 score [%]	DOA error [°]	Frame recall [%]	
Baseline	0.34	79.9	**28.5**	85.4	0.21
Lambda	0.33	80.9	32.6	85.4	0.21
LambdaDep	0.35	80.7	62.2	84.2	0.26
X-tree	0.34	72.0	70.8	79.3	0.31
LambdaFus	**0.19**	**88.6**	46.85	**88.61**	**0.17**

The Lambda model outperformed the baseline system in the SED task, however results for localization turned out to be slightly worse in terms of the DOA error. The SELD scores for both systems are equal to 0.21, which is a promising outcome, as the Lambda architecture shows a significantly smaller complexity. LambdaDep seems to perform worse than its primary model, although differences are not very significant, except for an increase in DOA error by two times. As this model demonstrates further reduction of system complexity, one way to improve performance would be to increase the number of filters. X-tree shows overall the worst results. Since it is the most complex among the proposed single models, it is the most difficult one to train efficiently. It might also need more training iterations. Best results are shown for LambaFus, which significantly outperforms all compared systems in terms of SELD score, showing that the first two systems work efficiently together even for a simple ensemble concept. However even in this case the DOA error turns out to be larger than that of the baseline system.

We note that 30 epochs of patience was used for the cross-validation sets to obtain training time reduction. A fuller picture of these ideas will be shown with some further experiments using a bigger patience, as well as with forthcoming evaluation results.

4. CONCLUSIONS

In this report, three single models created for the Sound Detection and Localization task are presented. The described architectures utilize ideas such as arborescent neural networks and depthwise separable convolutions. The proposed systems showed an important complexity reduction compared to the baseline system. Experiments performed for cross-validations splits showed promising results, with Lambda demonstrating an outcome comparable to the baseline. Additionally, an ensemble model LambdaFus is presented, showing that the proposed methods can complement each other in this field. However more research on these ideas is necessary.

5. REFERENCES

[1] S. Chu, S. Narayanan, C. C. J. Kuo, and M. J. Mataric, „Where am I? Scene recognition for mobile robots using audio features”, *IEEE International Conference Multimedia and Expo (ICME)*, 2006, pp. 885-888.

[2] V. Carletti, P. Foggia, G. Percanella, A. Saggese, N. Strisciuglio, and M. Vento, “Audio surveillance using a bag of aural words classifier”, *IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2013, pp. 81-86.

[3] M. Xu, C. Xu, L. Duan, J. S. Jin, and S. Luo, “Audio keywords generation for sports video analysis”, *ACM Transactions on Multimedia Computing, Communications and Applications (TOMM)*, vol. 4, no. 2, 2008, article no. 11.

[4] http://dcase.community/workshop2019/

[5] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen. “Sound event localization and detection of overlapping sources using convolutional recurrent neural networks”. *IEEE Journal of Selected Topics in Signal Processing*, 2018, vol. 13, pp. 34-38.

[6] F. Chollet, “Xception: Deep Learning with Depthwise Separable Convolutions”, *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[7] A. Mesaros, T. Heittola, and T. Virtanen, “Metrics for polyphonic sound event detection”, *Applied Sciences*, Vol. 6, No. 6, 162, 2016.

[8] S. Adavanne, A. Politis, and T. Virtanen, “Direction of arrival estimation for multiple sound sources using convolutional recurrent neural network”, *2018 26th European Signal Processing Conference (EUSIPCO)*, IEEE, 2018, pp. 1462–1466.