REASSEMBLY LEARNING FOR SOUND EVENT LOCALIZATION AND DETECTION USING CRNN AND TRELLISNET

Technical Report

Sooyoung Park, Wootaek Lim, Sangwon Suh, Youngho Jeong,

Electronics and Telecommunications Research Institute
Realistic AV Research Group
218 Gajeong-ro, Yuseong-gu, Daejeon, Korea
{sooyoung, wtlim, suhsw1210, yhcheong}@etri.re.kr

ABSTRACT

This technical report proposes a deep learning based approach, reassembly learning, for polyphonic sound event localization and detection. Sound event localization and detection is a joint task of two dependent sub-tasks: sound event detection and direction of arrival estimation. Joint learning has performance degradation compared to learning each sub-task separately. For this reason, we propose a reassembly learning to design a single network that deals with dependent sub-tasks together. Reassembly learning is a method to divide multi-task into individual sub-tasks, to train each sub-task, and then to reassemble and fine-tune into a single network. Experimental results show that the reassembly learning has good performance in the sound event localization and detection. Besides, the convolutional recurrent neural networks have been known as a state of art in both sound classification and detection applications. In DCASE 2019 challenge task 3, we suggest new architecture, trellis network based on temporal convolution networks, which can replace the convolutional recurrent neural networks. Trellis network shows a strong point in the direction of arrival estimation and has the possibility of being applied to a variety of sound classification and detection applications.

Index Terms— sound event localization and detection, reassembly learning, trellis network, convolutional recurrent network

1. INTRODUCTION

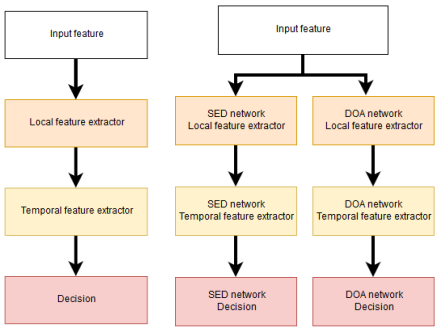
Sound event localization and detection (SELD) [1, 2] is a new estimation problem that combines sound event detection (SED) and direction of arrival estimation (DOAE) into a single task. So it has not been studied much yet. Therefore, SELD has many weaknesses. First, SELD should be able to assign the direction of arrivals (DOAs) with appropriate sound events. The DCASE 2019 baseline [1, 2] uses the Hungarian algorithm to minimize the pair-wise costs between individual DOA predictions and reference labels. Second, SELD should be able to estimate multiple DOAs simultaneously for polyphonic sound events. The DCASE 2019 baseline uses multi regression to estimate continuous DOAs for polyphonic sound events. However, these methods used in DCASE 2019 baseline do not fully solve the weaknesses of SELD.

Multi regression interrupts training because of its label configuration. When using multi regression to solve polyphonic SELD, the DOA label must be in a format that has azimuth and elevation values for all events. Therefore, DOA labels have true DOA values for active events and default DOA values for inactive events. The DCASE 2019 baseline sets the default direction to a value outside the target DOA range. However, DOA loss is still obtained from both active events and inactive events. So, it is not a proper method for polyphonic sound events. SELD with multi regression was designed to estimate the array containing the default DOA, not designed to estimate DOA for active events. Cao [3] proposed a two-stage learning method to avoid loss from inactive events. Two-stage learning excludes DOA prediction for inactive events by masking using ground truth event labels.

Two-stage learning solves the problem of multi regression by excluding the inactive events from the DOA loss. However two-stage learning still has the problem with inactive events at the stage of SED inference. Two stage learning derives the final SELD output prediction by concatenating SED network prediction and DOA network prediction. However, the DOA network excludes the inactive events in training, so the DOA prediction values of the inactive events can be random. Therefore, if the SED network does not inference the correct polyphonic sound events, DOA predictions for incorrect sound events interrupt matching between the events and directions in the Hungarian algorithm of SELD. As a result, DOA prediction for incorrect events brings performance degradation for DOA accuracy.

In this technical report, we propose a reassembly learning method to overcome the problem from inactive events. This method divides the existing SELD into three stages: the SED-only stage, DOA-only stage, and SELD stage. The first step is to decompose SELD into SED and DOAE. In the SED-only stage, the SED network is trained alone except for the DOA network. Likewise, the DOA network is trained alone except for the SED network in the DOA-only stage. When training the DOA network, inactive sound events are masked using ground truth event labels. Also, local feature layers of the SED network are transferred into the DOA network. Finally, the SELD network is reassembled through transfer learning for each trained SED network and DOA network.

Convolutional recurrent neural network (CRNN) is used in DCASE 2019 baseline. Furthermore, many SED and DOAE studies [1, 2, 3, 4] choose CRNN as basic network architecture. CRNN is currently a state of art in sound classifications and detection. CRNN architecture uses CNN as a local feature extractor and RNN as a temporal feature extractor as shown in Figure 1(a). In this technical report, we propose a new network structure for sound classification and detection using a trellis network [5] based on the temporal convolutional network (TCN).



(a) (b)

Figure 1: Network architecture - (a): Basic model for sound classification and detection, (b): SELD network with separable SED network and DOA network

The contribution in this technical report is as follows.

- We propose a method of minimizing DOAE performance degradation from the inactive event through reassembly learning. Also, reassembly learning can be used for not only SELD but also other joint of dependent sub-tasks
- We propose a new type of network architecture based on the temporal convolution network for sound classification and detection applications

2. DATASET

DCASE 2019 challenge task 3 provides audio data set for 11 classes of sound events. The sound event of DCASE 2019 data set are synthesized using spatial room impulse response recorded in five indoor locations. The development dataset consists of 400 files. Each audio file is a 1-minute duration with a sampling rate of 48000 Hz. The development dataset is provided as two different types: four channel tetrahedral microphone arrays and a first-order ambisonic (FOA) format. Besides, DCASE challenge task3 targets polyphonic sound events with a maximum of two sound events overlap.

3. FEATURE

Our models use log mel-band energy (4 channels), mel-band acoustic active intensity (3 channels) and mel-band acoustic reactive intensity (3 channels). Log mel-band energy is extracted from the tetrahedral microphone dataset. On the other hand, mel-band acoustic active and reactive intensity are extracted from FOA dataset.

3.1. Log mel-band energy

In the DCASE2018 challenge, many participants used the log mel-band energy as an input feature for a SED task. Mel-band energy is a feature that applies a mel filter to an energy spectrogram. The mel filter mimics the non-linear human auditory perceptions. The DCASE 2018 challenge results proved that this non-linear feature has strength for SED. Also, we expect to obtain information of time difference, loud difference for sound localization from a multi-channel log mel-band energy feature.

3.2. Mel-band acoustic intensity

Ambisonic is a coefficient of the spatial basis of the audio signal. The first order ambisonic consists of 0th order spherical harmonics coefficient and three 1st order spherical harmonics coefficients. The 0th order spherical harmonic coefficient can be viewed as a recording by using a virtual omnidirectional microphone. The 1st order spherical harmonic coefficients can be viewed as recording from virtual polarized bidirectional microphones. The sound source in an anechoic environment can be expressed easily by using FOAs as equation (1).

$$\begin{bmatrix} W(t,f) \\ X(t,f) \\ Y(t,f) \\ Z(t,f) \end{bmatrix} = \begin{bmatrix} 1 \\ \sqrt{3}cos\theta cos\phi \\ \sqrt{3}sin\theta cos\phi \\ \sqrt{3}sin\phi \end{bmatrix} \quad (1)$$

However, in the presence of multiple sources or reverberant environments, it is impossible to express complex sound fields using FOAs. Therefore, we need additional methods to extract spatial information from FOAs for the reverberant environment. Acoustic intensity [6] is one of these methods that extract spatial information from FOAs.

Acoustic intensity is one of the physical quantities representing the sound field. The acoustic intensity vector can be expressed by using FOA as equation (2). Active acoustic intensity vector is a real part of acoustic intensity that represents the flow of sound energy. It is a physical quantity directly related to DOA. The active acoustic intensity is expressed as a real part of the product of the pressure and the particle velocity. Reactive intensity is an imaginary part of acoustic intensity that represents a dissipative local energy transfer. It is a physical quantity dominated by direct sound from a single source. We expect to obtain spatial decomposed information and phase information from acoustic intensities of 6 channels of mel-band acoustic intensity.

$$\mathbf{I}(t,f) = p(t,f)\mathbf{v}^*(t,f) = -W(t,f)\begin{bmatrix} X^*(t,f) \\ Y^*(t,f) \\ Z^*(t,f) \end{bmatrix} \quad (2)$$

Finally, it is important that the size of the acoustic intensity feature and the size of mel-band energy feature are equal to deal with those features in the single network. Therefore, mel filter is applied to resize acoustic intensity features.

4. NETWORK ARCHITECTURE

The proposed models follow a basic architecture in which local feature extractor and temporal feature extractor are connected. Figure 1(a) shows the basic architecture for the sub-tasks of SELD. We propose a network as shown in Figure 1(b). The model has the same local feature extractor for the SED network and DOA network. This is because transfer learning for local feature extractor from SED network to DOA network can improve the performance of DOAE [3].

4.1. Reassmebly learning

We propose reassembly learning to design a single network for multi-task problem consists of dependent sub-tasks. Reassembly learning consists of three stages. The first stage is learning SED-only networks. After then the local feature extractor of the learned

SED-only network is transferred to the DOA-only network. In the second stage, the loss of the DOA-only network is calculated with masking inactive event by the ground truth SED labels. The last stage is the SELD stage. SELD network initializes the whole parameter from the pre-trained SED network and the DOA network as shown in Figure 2. At this stage, we do not use masking in the DOA-only stage, but use the same multi regression in the baseline. We tried to solve the problem in two-stage learning through fine-tuning training in the SELD stage.

4.2. Local feature extractor

In Figure 5 and 6, the local feature extractor consists of four gated linear unit blocks and global average pooling that compresses the frequency (mel-bin) axis. Both the SED network and the DOA network use the same local feature extractor structure.

4.2.1. Gated linear unit (GLU)

Gated linear unit [7] is a type of context gating layer. It is known as CNN based local attention mechanism. In DCASE 2018 challenge task 4, Jiakai [4] makes the best performance through GLU based CRNN. For this reason, we use GLU block for a local feature extractor.

Figure 3 shows specification of GLU blocks. Each GLU block consists of two convolution layers with (3,3) kernel, (1,1) stride, (1,1) padding, and the same number of filters. Batch normalization layer[8] and GLU activation layer follow after each convolution layer. At the end of the block, temporal and spectral compression is performed with (2,2) average pooling layer. So local feature extractor can take out frequency information and local temporal information. Finally, the frequency information is fully compressed to a single dimension using global average pooling at the end of the local feature extractor.

4.3. Temporal feature extractor

There are two different types of temporal feature extractors for the proposed model. One is Bidirectional-GRU, one of RNNs. The other is Bidirectional-TrellisNet which is a special form of TCNs.

4.3.1. Bidirectional gated recurrent unit (Bi-GRU)

RNN is widely used for sequence modeling. The basic idea of RNN is to predict the output of the current state by using previous input. Theoretically, the output of the current state can be predicted

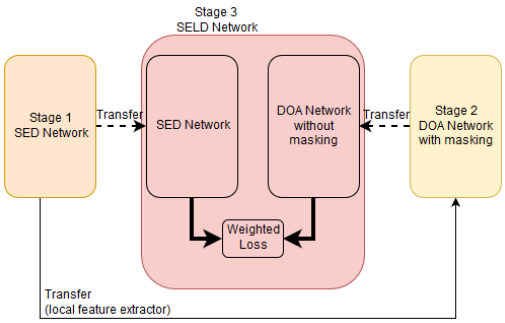
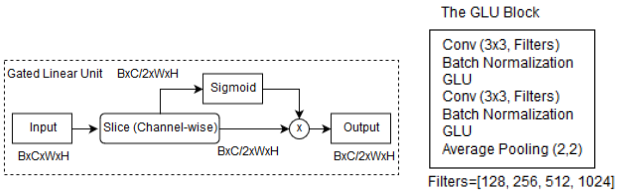


Figure 2: Reassembly learning for SELD



(a) (b)

Figure 3: Description of GLU block - (a): GLU module, (b): GLU block (B: batch size, C: channel, W: width, H: height)

by using an infinite length of past information. However, in actual RNN, the vanishing gradient occurs while repeating the sequence operation. It means that the stability of the RNN structure is not guaranteed. Therefore gating mechanism has been proposed such as LSTM [9] and GRU [10]. Both methods have almost similar performance. However, since the computation amount of GRU is less than LSTM. So GRU is used for sequence modeling in various applications like DCASE 2019 baseline.

4.3.2. Bidirectional trellis network (Bi-TrellisNet)

Trellis Network [5] tried to combine CNNs and RNNs through direct input injection and weight sharing among TCN layers. The fusion of CNNs and RNNs will take advantage of both structural and algorithmic elements. Figure 4 shows a simple structure of TrellisNet. TrellisNet can replace the recurrent structure of the RNNs by stacking of multiple temporal convolution layers and adding the input injection and weight sharing techniques. In a basic trellis network, LSTM is applied between each temporal convolution layer. TrellisNet with the specific sparse weight matrix is proved to equal to M-truncated RNNs [5]. Therefore, the full weight matrix is expected to have greater power for sequence modeling.

4.4. Proposed model

The proposed networks for sub-tasks are the combinations of GLU, Bi-GRU, and Bi-TrellisNet. Figure 5 and 6 show the proposed models for DCASE 2019 task 3.

5. EVALUATION RESULT

5.1. Experimental setup

The input features are 10 channels (C): log mel-band energy (4 channel) and mel-band acoustic intensity (6 channel) with 96 mel-

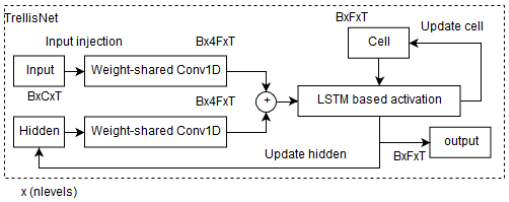


Figure 4: The diagram of TrellisNet (B: batch size, C: channel, T: time, F: filter, nlevels: layers in TrellisNet)

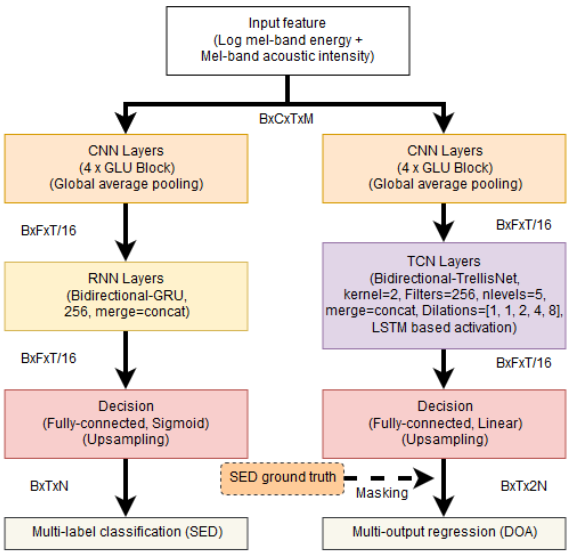


Figure 5: Architecture of proposed model: Reassembly v1 and v2 (SED network: GLU-RNN, DOA network: GLU-TrellisNet; B: batch size, C: channel, T: time, M: mel-bin, F: filter, N: classes)

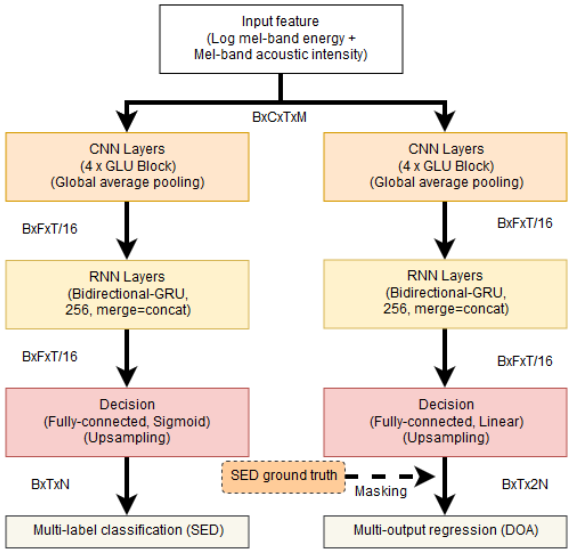


Figure 6: Architecture of proposed model: Reassembly v3 and v4 (SED network: GLU-RNN, DOA network: GLU-RNN; B: batch size, C: channel, T: time, M: mel-bin, F: filter, N: classes)

bin (M). The sample rate of STFT is set to 32 kHz to avoid unnecessary high-frequency noisy components in the input feature. For STFT, a 1024-point Hanning window is applied with 1024 nfft, and the hop length was set to 10 ms. For training, 200 frames (T) of data were used and the overlap is set to 100 frames (1 second). The batch size (B) used for training is 32. The number of epochs for the SED-only stage and DOA-only stage is 50. The learning rate of SED-only and DOA-only is 0.001 and decreases by 10% per epoch after 30 epochs. On the other hand, the learning rate of the SELD stage for all proposed model is 0.0001 and decreases after 30 epochs. The number of epochs for Reassembly v2 and v4 are 50, while we trained Reassembly v1 and v2 until 18 epochs.

5.2. Results

Table 1 shows the experimental results by using pre-defined four-fold cross-validation split for DCASE 2019. We combined training and validation split for training proposed models. In the early stage of reassembly learning, the proposed SED network has improved error rate and F score by 15% and 10%, respectively, compared to baseline. Also, the proposed DOA network achieves 20 degrees improvement for DOA error. In the DOA network, GLU-TrellisNet shows better performance compared to GLU-RNN. The final stage of reassembly learning has improved about 1.7 degrees for DOA error compared to DOA-only. Generally, the performance of the final SELD network through reassembly learning inherited the performance of SED-only. However, some models suffer SED performance degradation due to the retraining of the SED network in reassembly learning. This is expected to be solved by transferring the SED network to the SELD network as non-trainable.

Table 1: Experimental results for DCASE 2019 task 3

Name	ER	F	DOA	FR	SELD
Baseline (FOA)	34	79.9	28.5	85.4	-
Baseline (MIC)	35	80.0	30.8	84.0	-
SED-only	**16**	**90.6**	-	**85.6**	-
DOA-only (Trellis)	-	-	**8.21**	-	-
DOA-only (RNN)	-	-	9.94	-	-
Reassembly v1	**16**	**90.6**	6.41	**85.7**	11.0
Reassembly v2	17	90.5	**6.39**	85.6	11.1
Reassembly v3	18	89.9	8.26	85.2	11.9
Reassembly v4	17	90.1	8.26	85.4	11.7

6. CONCLUSION

We proposed reassembly learning to solve SELD consist of two dependent sub-tasks. Reassembly learning is a way to retrain network consists of pre-trained sub-task networks. Through reassembly learning, we tried to solve the problem of multi regression loss used for continuous polyphonic SELD. As a result, the proposed models significantly improved both SED and DOAE performance compared to the baseline. Reassembly learning has the potential to apply not only to DCASE task 3 but also to other multi-task problems. We proved that the log mel-band energy and mel-band intensity are helpful input features for SED and DOAE. Also, the DOAE network using TrellisNet showed better performance than CRNN. Thus TCN based architecture demonstrated the possibility for other sound classification and detection applications.

7. ACKNOWLEDGMENT

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2017-0-00050, Development of Human Enhancement Technology for auditory and muscle support)

8. REFERENCES

[1] 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*, pp. 1–1, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8567942/

[2] S. Adavanne, A. Politis, and T. Virtanen, "Direction of Arrival Estimation for Multiple Sound Sources Using Convolutional Recurrent Neural Network," in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, sep 2018, pp. 1462–1466. [Online]. Available: https://ieeexplore.ieee.org/document/8553182/

[3] Y. Cao, Q. Kong, T. Iqbal, F. An, W. Wang, and M. D. Plumbley, "Polyphonic Sound Event Detection and Localization using a Two-Stage Strategy," may 2019. [Online]. Available: http://arxiv.org/abs/1905.00268

[4] L. JiaKai, "Mean Teacher Convolution System for DCASE 2018 Task 4," DCASE2018 Challenge, Tech. Rep., sep 2018.

[5] S. Bai, J. Z. Kolter, and V. Koltun, "Trellis Networks for Sequence Modeling," in *ICLR, International Conference on Learning Representations*, 2019, pp. 1–17. [Online]. Available: http://arxiv.org/abs/1810.06682

[6] L. Perotin, R. Serizel, E. Vincent, and A. Guerin, "CRNN-Based Multiple DoA Estimation Using Acoustic Intensity Features for Ambisonics Recordings," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 1, pp. 22–33, mar 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8643769/

[7] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language Modeling with Gated Convolutional Networks," dec 2016. [Online]. Available: http://arxiv.org/abs/1612.08083

[8] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 2015, pp. 448–456. [Online]. Available: http://proceedings.mlr.press/v37/ioffe15.html

[9] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, nov 1997. [Online]. Available: http://www.mitpressjournals.org/doi/10.1162/neco.1997.9.8.1735

[10] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.