INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Learning How to Listen: A Temporal-Frequential Attention Model for Sound Event Detection

Yu-Han Shen, Ke-Xin He, Wei-Qiang Zhang[*]

Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
yhshen@hotmail.com, hekexinchn@163.com, wqzhang@tsinghua.edu.cn

Abstract

In this paper, we propose a temporal-frequential attention model for sound event detection (SED). Our network learns how to listen with two attention models: a temporal attention model and a frequential attention model. Proposed system learns when to listen using the temporal attention model while it learns where to listen on the frequency axis using the frequential attention model. With these two models, we attempt to make our system pay more attention to important frames or segments and important frequency components for sound event detection. Our proposed method is demonstrated on the task 2 of Detection and Classification of Acoustic Scenes and Events (DCASE) 2017 Challenge and outperforms state-of-the-art methods.

Index Terms: sound event detection, convolutional neural network, recurrent neural network, attention model, temporal-frequential attention

1. Introduction

Nowadays, sound event detection (SED), also named as acoustic event detection(AED), is considered as a popular topic in the field of acoustic signal processing. The aim of SED is to temporally locate the onset and offset times of target sound events present in an audio recording.

The Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge is an international challenge concerning SED, and has been held for several years. In DCASE 2017 Challenge, the theme of task 2 is “detection of rare sound events” [1]. It provides dataset [2] and baseline for rare sound event detection in synthesized recordings. “Rare” means that target sound events (babycry, glassbreak, gunshot) would occur at most once within a 30-second recording. And the mean duration of target sound event is very short: 2.25 s for babycry, 1.16 s for glassbreak, 1.32 s for gunshot, leading to a serious problem of data imbalance. All audio recordings are annotated with ground-truth labels of event class, onset and offset time. According to the task description, a separate system should be developed for each of the three target event classes to detect the temporal occurrences of these events [1].

Among the submissions in DCASE 2017, most models are based on deep neural networks. Both of the top 2 teams [3, 4] utilized Convolutional Recurrent Neural Networks (CRNN) as their main architecture. They combined Convolutional Neural Networks (CNN) with Recurrent Neural Networks (RNN) to make frame-level predictions for target events and then adopted post-processing to get the onset and offset time of sound events. Kao et al. [5] proposed a Region-based Convolutional Recurrent Neural Network (R-CRNN) to improve previous work in 2018. In our work, we followed the main architecture of those three models and used CRNN as main classifier.

Inspired by the excellent performance of attention model in machine translation [6], image caption [7], speaker verification [8], audio tagging [9], we proposed an attention model for SED. Currently, most attention models in speech and audio processing only concentrate on time domain. We proposed a temporal-frequential attention model to focus on important frequency components as well as important frames or segments. Our attention model can learn how to listen by extracting not only temporal information but also spectral information. Besides, we visualized the weights of attention models to show what our models have actually learnt.

The rest of this paper is organized as follows: in Section 2, we introduce our methods in detail, mainly including feature extraction, baseline and temporal-frequential attention model. The dataset, experiment setup and evaluation metric are illustrated in Section 3. The results and analysis are presented in Section 4. Finally, we conclude our work in Section 5.

2. Methods

2.1. System overview

As shown in Figure 1, our proposed system is a CRNN architecture with temporal-frequential attention model. The input of our system is a 2-dim acoustic feature. It is fed into a frequential attention model to produce frequential attention weights. Our system learns to focus on specific frequency components of audios using those attention weights. The input acoustic feature will multiply with those attention weights and then pass through CRNN architecture. Compared with traditional CRNN [3, 4], we add a temporal attention model to let our system pay different attention to different frames. The temporal attention weights will multiply with the outputs of CRNN by element-wise. A sigmoid activation is used to get normalized probabilities. Then we utilize post processing to get final detection outputs.

2.2. Feature extraction

The acoustic feature used in our work is log filter bank energy (Fbank). The sampling rate of input audios is 44.1kHz. To extract Fbank feature, each audio is divided into frames of 40 ms duration with shifts of 20 ms. Then we apply 128 mel-scale filters covering the frequency range 300 to 22050 Hz on the magnitude spectrum of each frame. Finally, we take logarithm on the amplitude and get Fbank feature. The extracted Fbank feature is normalized to zero mean and unit standard deviation before being fed into neural networks.

2.3. Baseline

We adopt state-of-the-art CRNN as baseline. The input is Fbank feature of 30-second audios. And the output of our system gives binary predictions for each frame with time resolution of 80 ms

[*] corresponding author

http://dx.doi.org/10.21437/Interspeech.2019-2045

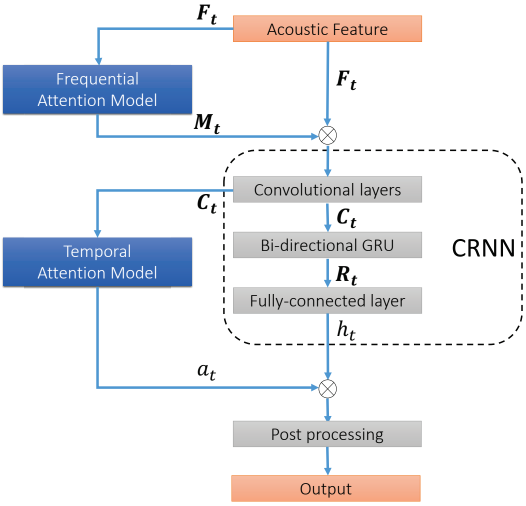


Figure 1: *Illustration of overall system. The frequential attention model is added to filter input features. And the temporal attention model is used after convolutional layers in order to produce attention weights for different frames.*

(the pooling layers have reduced frame rate from 50 Hz to 12.5 Hz).

The CRNN architecture consists of three parts: convolutional neural network (CNN), recurrent neural network (RNN) and fully-connected layer. The architecture of our CRNN is similar to that in [5], and it is shown in Figure 2.

The CNN part contains four convolutional layers, and each layer is followed by batch normalization [10], ReLU activation unit and dropout layer [11]. We add two residual connections [12] to improve the performance of CNN. Max-pooling layers (on both time axis and frequency axis) are used to maintain the most important information on each feature map. At the end of CNN, the extracted features over different convolutional channels are stacked along the frequency axis.

The RNN part is a bi-directional gated recurrent unit (bi-GRU) layer. Compared with uni-directional GRU, bi-GRU can extract temporal structures of sound events better. We add the outputs of forward GRU and backward GRU to get final outputs of bi-GRU. The size of the output of bi-GRU is (375, U), where U is the number of GRU units.

After the bi-GRU, a single fully-connected layer with sigmoid activation is used to give classification result for each frame (80 ms). The output denotes the presence probabilities of the target event in each frame.

In order to determine the presence of an event, a binary prediction is given for each frame with a constant threshold of 0.5. These predictions are post-processed with a median filter of length 240 ms. Since at most one event would occur in a 30-s audio, we select the longest continuous sequence of positive predictions to get the onset and offset of target events.

2.4. Learning when to listen

As shown in Figure 1, we add a temporal attention model at the end of CNN to enable our system to learn when to listen. This attention model was proposed to ignore irrelevant sounds and focus more on important frames. Unlike the attention model in audio classification [9] that only focuses on positive frames (including events), our temporal attention pays more attention to both positive frames and hard negative frames (only backgrounds, but easily misclassified as events) because they should be further differentiated.

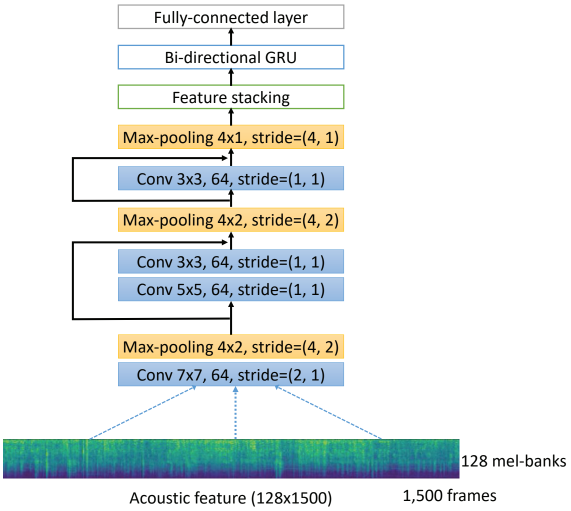

Fully-connected layer
Bi-directional GRU
Feature stacking
Max-pooling 4x1, stride=(4, 1)
Conv 3x3, 64, stride=(1, 1)
Max-pooling 4x2, stride=(4, 2)
Conv 3x3, 64, stride=(1, 1)
Conv 5x5, 64, stride=(1, 1)
Max-pooling 4x2, stride=(4, 2)
Conv 7x7, 64, stride=(2, 1)
128 mel-banks
Acoustic feature (128x1500)
1,500 frames

Figure 2: *The architecture of CRNN. The first and second dimensions of convolutional kernels and strides represent the time axis and frequency axis respectively.*

The output of CNN will pass through a fully-connected layer with N_t hidden units, followed by an activation unit (sigmoid, ReLU, or softmax). Then a global max-pooling on the frequency axis is used to get one weight for each frame. Those attention weights will be normalized along time axis. In our experiments, this operation of normalization has shown great effectiveness because it takes into account the variation of weight factors along time axis instead of considering only current frame. Then we multiply the temporal attention weights with the output of the fully-connected layer after bi-GRU. A sigmoid function is used to normalize the probabilities to $[0, 1]$. The final output can be computed as follows:

$$\hat{a}_t = \max_{n \in \{1,2,3,\ldots,N_t\}} \{\sigma(\boldsymbol{W_n} \boldsymbol{C_t} + b_n)\}, \tag{1}$$

$$a_t = T \frac{\hat{a}_t}{\sum_t \hat{a}_t}, \tag{2}$$

$$y_t = \frac{1}{1 + \exp(-a_t h_t)}, \tag{3}$$

where $\sigma(\cdot)$ is an activation function, $\boldsymbol{C_t}$ denotes the output of CNN, $\boldsymbol{W_n}$ and b_n represent the weights and bias for the n-th hidden unit respectively, $n \in \{1, 2, 3, \ldots, N_t\}$ and N_t is the number of hidden units in temporal attention model. $\hat{a}_t$ is the candidate temporal attention weight, T is the total number of frames in an audio, a_t is the normalized temporal attention weight, and y_t is the final output probabilities.

2.5. Learning where to listen

Apart from temporal attention model, we proposed a frequential attention model. As we all know, various sound events may have different spectral characteristics. So we assume that we should treat those frequency components differently based on the characteristic of each frame.

Table 1: *Performance of proposed models and other methods, in terms of ER and F-score (%). *** indicates that class-wise results are not given in related paper. We compare the following models: (1) Baseline: our bi-GRU-based CRNN; (2) CRNN+TA: our bi-GRU-based CRNN with temporal attention model; (3) Proposed: our bi-GRU-based CRNN with temporal-frequential attention model; (4) R-CRNN: Region-based CRNN; (5) 1d-CRNN: DCASE 1st place model; (6) CRNN: DCASE 2nd place model.*

Model	Development Dataset				Evaluation Dataset			
	babycry	glassbreak	gunshot	average	babycry	glassbreak	gunshot	average
Baseline	0.14|92.6	0.04|98.0	0.19|89.6	0.12|93.4	0.31|83.4	0.08|95.9	0.26|85.5	0.22|88.3
CRNN+TA	0.14|92.8	0.03|98.4	0.17|90.9	0.11|94.0	0.25|87.4	0.05|97.4	0.18|90.6	0.16|91.8
Proposed	0.10|95.1	0.01|99.4	0.16|91.5	0.09|95.3	0.18|91.3	**0.04|98.2**	**0.17|90.8**	**0.13|93.4**
R-CRNN [5]	0.09| ***	0.04| ***	**0.14**| ***	0.09|95.5	******	******	******	0.23|87.9
1d-CRNN [3]	**0.05|97.6**	**0.01|99.6**	0.16|91.6	**0.07|96.3**	**0.15|92.2**	0.05|97.6	0.19|89.6	0.13|93.1
CRNN [4]	******	******	******	0.14|92.9	0.18|90.8	0.10|94.7	0.23|87.4	0.17|91.0

The structure of frequential attention model is similar to temporal attention model. The input Fbank feature will go through a fully-connected layer with N_f hidden units, followed by an activation function (sigmoid, ReLU, or softmax). Here, N_f is set to 128 to correspond with the number of mel-filters. Then it is normalized along the frequency axis to get frequential attention weights. Finally, an element-wise multiplication is adopted between the frequential attention weights and input Fbank feature before the feature is fed into CRNN architecture. The weighted feature is computed as follows:

$$\hat{\boldsymbol{M}}_{\boldsymbol{n,t}} = \sigma(\boldsymbol{V_n F_t} + c_n), \tag{4}$$

$$\boldsymbol{M}_{\boldsymbol{n,t}} = N_f \frac{\hat{\boldsymbol{M}}_{\boldsymbol{n,t}}}{\sum_n \hat{\boldsymbol{M}}_{\boldsymbol{n,t}}}, \tag{5}$$

$$\tilde{\boldsymbol{F}}_{\boldsymbol{t}} = \boldsymbol{M_t} \otimes \boldsymbol{F_t}, \tag{6}$$

where $\sigma(\cdot)$ is an activation function, $\boldsymbol{F_t}$ is the input acoustic feature, $\boldsymbol{V_n}$ and c_n represent the weights and bias for the n-th hidden unit respectively. $\hat{\boldsymbol{M}}_{\boldsymbol{n,t}}$ is the candidate frequential attention weight, $\boldsymbol{M}_{\boldsymbol{n,t}}$ is the normalized frequential attention weight, $\otimes$ represents element-wise multiplication and $\tilde{\boldsymbol{F}}_{\boldsymbol{t}}$ is the weighted feature.

3. Experiments

3.1. Dataset

We demonstrate proposed model on DCASE 2017 Challenge task 2 [1]. The task dataset consists of isolated sound events for each target class and recordings of everyday acoustic scenes to serve as background [2]. There are three target event classes: babycry, glassbreak and gunshot. A synthesizer for creating mixtures at different event-to-background ratios is also provided. The dataset is comprised of development dataset and evaluation dataset. The development dataset also consists of two parts: train subset and test subset. Participants are allowed to use any combination of the provided data for training, and evaluate their models on the test subset of development dataset. Ranking of submitted systems is based on their performance on evaluation dataset. Detailed information about this task and dataset is available in [1][2].

We use the provided synthesizer to generate 3000 mixtures for each class. The event-to-background ratios are -6, 0, 6dB, and the event presence probability is set to 0.9 (default value: 0.5) in order to gain more positive samples and mitigate the problem of data imbalance. We use the development test subset to optimize our model and finally evaluate it on the evaluation dataset.

3.2. Experiment setup

Our model is trained using Adam [13] with learning rate 0.001. Due to data imbalance, we use weighted cross-entropy loss function to reduce deletion error. The loss weight for positive samples is 10.

In order to accelerate training, we adopt pre-training strategy. We firstly train the baseline CRNN for 10 epoches and then use the pre-trained CRNN to initialize the weights during the training of proposed model. The training is stopped after 200 epoches. The batch size is 64. The number of hidden layer unit in temporal attention model N_t is 32. The number of GRU units U is 32.

Because our work is a 0/1 classification system, we use sigmoid and ReLU activation in attention models. According to experimental results, our system can achieve the best performance with ReLU activation in temporal attention model and sigmoid activation in frequential attention model.

3.3. Metrics

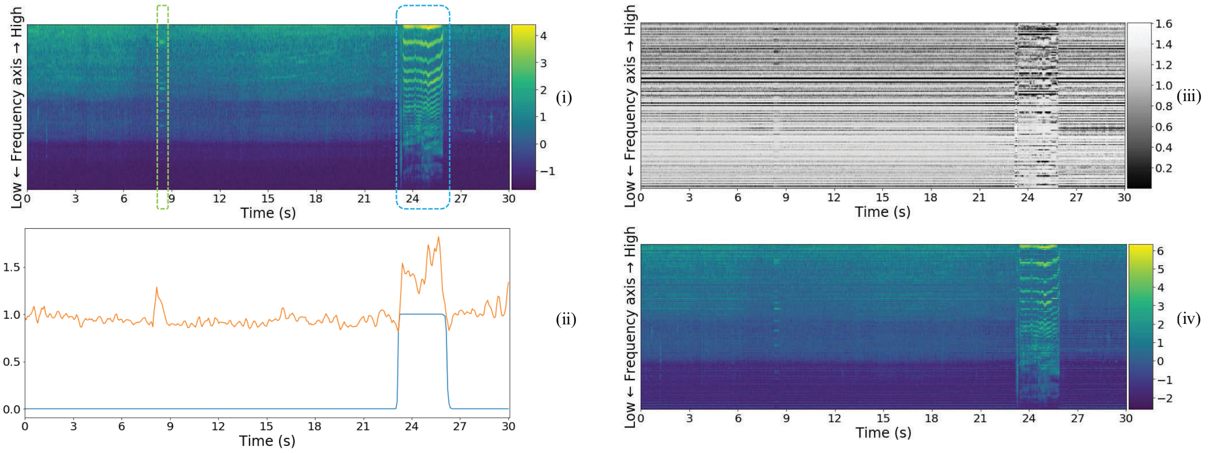
We follow the official evaluation metrics of DCASE Challenge. Our method is evaluated based on two kinds of event-based metrics: event-based error rate (ER) and event-based F-score. Both metrics are computed as defined in [14], using a collar of 500 ms and considering only the event onset. If the output accurately predicts the presence of target event and its onset, we denote it as correct detection. The onset detection is considered accurate only when it is predicted within the range of 500 ms of the actual onset time. ER is the sum of deletion error and insertion error, and F-score is the harmonic average of precision and recall. We compute these metrics using sed_eval toolbox [14] provided by DCASE organizer.

4. Results

4.1. Experimental results

The performances of proposed models and other methods, in terms of ER and F-score, are shown in Table 1. Results show that temporal attention model can improve the performance of bi-GRU based CRNN baseline, and frequential attention model can make further improvement. Compared with baseline, proposed method can improve the performance of all classes on both development dataset and evaluation dataset.

Compared with other state-of-the-art methods, the performance of our model is also competitive. Note that both of the top 2 teams adopt ensemble method. Lim et al. [3] combined the output probabilities of more than four models with different time steps and different data mixtures to make final decision.



(a) Visualization of temporal attention weights

(b) Visualization of frequential attention weights

Figure 3: *Visualization of attention models. (i) is the mel-spectrogram of an audio recording, where the yellow box denotes noisy and the blue box denotes target event. (ii) is the representation of temporal attention model, where the blue line denotes the output probability and the orange line denotes the temporal attention weights. (iii) is the representation of frequential attention weights and (iv) is the spectrogram of weighted feature.*

Cakir et al. [4] utilized the ensemble of seven architectures. We can achieve comparable results on development dataset without any model ensemble. Moreover, the average ER only increases slightly from 0.09 to 0.13 on evaluation dataset. We believe that our proposed model has a better capability of generalization. Proposed model achieves the lowest average ER (0.13) and the highest average F-score (93.4%) on evaluation dataset, outperforming all other methods.

4.2. Visualization of attention models

In order to know more about our attention models, we visualize the weights of both temporal attention model and frequential attention model. Presented in Figure 3 is a good example of what our proposed temporal-frequential attention model has actually learnt. Figure 3 (a) and (b) are visualization of temporal attention weights and frequential attention weights respectively.

In Figure 3, (i) is the mel-spectrogram of an audio in the evaluation dataset. In this audio, babycry occurs from 23.13 s to 26.16 s with "bus" background. There is a "beep" sound at around 9-th second. In (ii), the blue line denotes the output probability and the orange line denotes the temporal attention weights. We can notice that the weight value is bigger when "beep" and "babycry" occur, which conforms with our previous assumption that temporal attention model gives more attention to positive frames and hard negative frames. (iii) is the visualization of frequential attention weights and (iv) is the spectrogram of weighted feature. We can find that the value of frequential attention weight is bigger in low-frequency area, which means that our frequential attention pays less attention to high frequency components. This can be considered as a low-band filter and frequential attention model can ignore some high-frequency noise.

5. Conclusion

In this paper, we proposed a temporal-frequential attention model for sound event detection. Proposed model is tested on DCASE 2017 task 2. Our system can achieve the best performance on DCASE evaluation dataset even without model ensemble. In the future, we can adopt this method to solve more difficult tasks of sound event detection, such as polyphonic sound event detection. In addition to sound event detection, our temporal-frequential attention model can be applied in speaker verification, speech recognition, audio tagging for further research.

6. Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. U1836219.

7. References

[1] 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 *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop*, 2017, pp. 85–92.

[2] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *2016 24th European Signal Processing Conference (EUSIPCO)*, Aug 2016, pp. 1128–1132.

[3] H. Lim, J. Park, K. Lee, and Y. Han, "Rare sound event detection using 1d convolutional recurrent neural networks," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop*, 2017, pp. 80–84.

[4] E. Cakir and T. Virtanen, "Convolutional recurrent neural networks for rare sound event detection," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop*, 2017, pp. 803–806.

[5] C.-C. Kao, W. Wang, M. Sun, and C. Wang, "R-CRNN: Region-based Convolutional Recurrent Neural Network for Audio Event Detection," *arXiv preprint arXiv:1808.06627*, Aug. 2018.

[6] 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.

[7] J. Lu, C. Xiong, D. Parikh, and R. Socher, “Knowing when to look: Adaptive attention via a visual sentinel for image captioning,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3242–3250.

[8] F. R. rahman Chowdhury, Q. Wang, I. L. Moreno, and L. Wan, “Attention-based models for text-dependent speaker verification,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5359–5363.

[9] Q. Kong, Y. Xu, W. Wang, and M. D. Plumbley, “Audio set classification with attention model: A probabilistic perspective,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 316–320.

[10] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International Conference on Machine Learning*, 2015, pp. 448–456.

[11] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[12] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

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

[14] A. Mesaros, T. Heittola, and T. Virtanen, “Metrics for polyphonic sound event detection,” *Applied Sciences*, vol. 6, no. 6, 2016.