BIDIRECTIONAL GRU FOR SOUND EVENT DETECTION

Rui Lu

Tsinghua University
Department of Automation
Beijing, P.R.China
lur13@mails.tsinghua.edu.cn

*Zhiyao Duan**

University of Rochester
Department of Electrical and Computer Engineering.
Rochester, NY USA
zhiyao.duan@rochester.edu

ABSTRACT

Sound event detection (SED) aims to detect temporal boundaries of sound events from acoustic recordings. Sound events in real-life recordings often overlap with each other (i.e., polyphonic), making this task difficult. Recently, multi-label recurrent neural networks (RNNs) have shown promises on polyphonic sound event detection. However, similar to many other deep learning approaches, the relative scarcity of carefully labeled data has limited the capacity of RNNs. In this paper, we first present a multi label bi-directional recurrent neural network to model the temporal evolution of sound events. Secondly, we explore data augmentation techniques that have shown success in sound classification [1]. We evaluate our approach on the development subset of the DCASE2017 task3 dataset [2]. Combining with data augmentation and ensemble techniques, we reduce the error rate by over 11% compared to the officially published baseline system. In addition, on the final evaluation dataset, our submitted system won the fourth place among all the 34 systems.

Index Terms— Sound event detection, recurrent neural networks, bidirectional GRU, data augmentation

1. INTRODUCTION

In this work, we intend to deal with the problem of data scarcity. We mainly have two contributions: first of all, we make use of the gated recurrent units (GRUs) [3] instead of LSTM cells. Though GRU cells and LSTM cells perform almost the same in general, GRU cells are less prone to over-fitting on relatively small datasets since they have two gates while LSTM cells have three. Secondly, inspired by the success of data augmentation in sound classification tasks [1], we also exploit the deformation techniques which are originated from the field of music information retrieval (MIR) [4]. We conduct experiments on the development dataset of task3 in DCASE2017 following the officially required cross validation setup and evaluation metrics [5][6]. Experimental results show that with the data augmentation techniques, the problem of over-fitting is greatly reduced. And when we ensemble several models for final prediction, a reduction of over 11% on error rate is achieved compared to the official baseline system. Our proposed systems also performed good on the evaluation dataset of task3 in DCASE2017, we ranked No.3 among the 13 teams.

*ZD acknowledges the National Science Foundation grant No. 1617107.

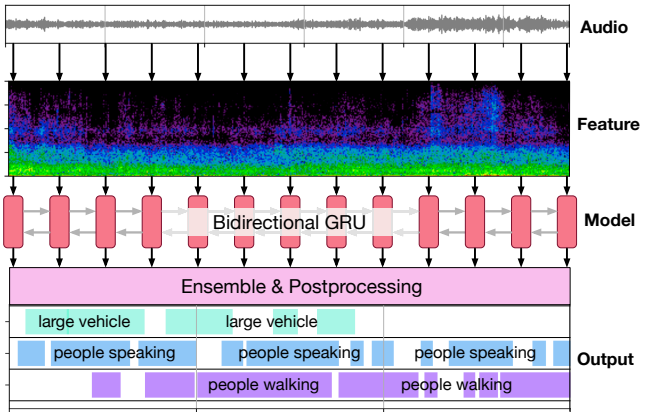

Figure 1: Overview of the system: we first extract frame-level features given the raw audio signal. Then we feed the features into the presented bidirectional recurrent neural network for multi-label prediction. Ensemble and post-processing techniques discussed in Sect.2 are applied for final sound event outputs.

2. METHOD

The overview of our system is shown in Fig.1: we first extract features such as mel-frequency cepstral coefficients ($mfcc$), log mel-spectrograms (lms), pitch frequency ($pitch$) and the time difference of arrival of FFT ($tdoa$) features [7] from the raw audio data. In our experiments, sounds from both channels are averaged (except $tdoa$ features since they make use of both channels) and resampled to 44100 Hz and all the features are extracted at a hop length of 20 ms to keep consistency (since the frame lengths are feature related, we introduce them in the following paragraphs). Their concatenations are then fed into our model for prediction. Finally we apply ensemble and post-processing for sound event outputs.

2.1. Features

We make use of two sets of features described below.

- $mfcc$, $mfcc0$ and lms: for lms, we apply hann window and STFT at a window length of 40 ms. Then, we apply 40-band mel-scale triangular filter ranging from 0 Hz to 22050 Hz to obtain the 40-dimension lms; Similarly, we extract 20-dimension $mfcc$ features and their first- and second-order derivatives with respect to time, which results in a total number of 59-dimension augmented $mfcc$ features by excluding the first di-

Feature	BLSTM	BGRU
mfcc	0.667 ± 0.012	0.662 ± 0.005
mfcc0	0.668 ± 0.010	$\mathbf{0.658 \pm 0.007}$
lms	0.752 ± 0.030	0.728 ± 0.029
mfcc + pitch	0.664 ± 0.015	$\mathbf{0.650 \pm 0.011}$
mfcc0 + pitch	0.661 ± 0.007	0.660 ± 0.022
lms + pitch	0.750 ± 0.029	0.712 ± 0.012
mfcc + tdoa	0.669 ± 0.013	0.662 ± 0.009
lms + tdoa	0.753 ± 0.042	0.745 ± 0.030
mfcc + pitch + tdoa	0.667 ± 0.011	0.669 ± 0.017
mfcc0 + pitch + tdoa	0.677 ± 0.029	$\mathbf{0.660 \pm 0.008}$
mfcc + pitch + tdoa3	0.669 ± 0.019	0.668 ± 0.024
lms + pitch + tdoa	0.759 ± 0.037	0.718 ± 0.022
lms + pitch + tdoa3	0.740 ± 0.026	0.723 ± 0.028

Table 1: Feature selection: for each feature combination, we experiment for 10 times, calculate the mean and standard deviation of corresponding error rates.

mension of $mfcc$. We denote mfcc features with the first dimension as $mfcc0$.

- $pitch$, $tdoa3$ and $tdoa$: We follow [7] to extract the $pitch$, $tdoa3$ and $tdoa$ features. We implement pitch tracking on the thresholded parabolically-interpolated STFT [8] to extract the top pitch value with window length of 2048, ranging from 100 Hz to 4000 Hz. Including the pitch period, we finally get 2-dimension $pitch$ features. By calculating the time difference of arrival (TDOA) of the frequency spectrum, we are intended to capture the difference of localizations of overlapping sound events. We follow [7, 9] to calculate the TDOA of FFT: we first extract the correlation of both channels' FFTs at certain frequency band and time stamp, then find the delay value (the TDOA value) that causes the peak of the correlation. Our TDOA features are calculated across five mel-bands ranging from 0 Hz to 22050 Hz, with three different window lengths: 120, 240 and 480 ms. Thus resulting in 15-dimension features, we refer to them as $tdoa3$ features. Similar to those described in [7], for each sub-band, we take the median value of the TDOA values from three window lengths to overcome the potential noise. This operation results in the 5-dimension features, which we refer to as $tdoa$ features.

2.2. Model

We propose to use a special kind of RNNs, named gated recurrent unit (GRU) networks, to model the temporal evolution of audio features toward SED. Suppose we are given a sequence of input vectors $< \mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^T >$ (concatenations of features in our setting), a GRU unit computes the corresponding hidden activations $< \mathbf{h}^1, \mathbf{h}^2, ..., \mathbf{h}^T >$, and outputs a vector sequence $< \mathbf{y}^1, \mathbf{y}^2, ..., \mathbf{y}^T >$.

Simple RNN usually fails to capture the long-term information due to the gradient vanishing problem. Similar to LSTM, the GRU [3] is also designed to dynamically remember and forget the information flow, which can alleviate the above problem by a large margin. Specifically, let $\odot$ denote the element-wise multiplication of two vectors, the single layer GRU computes the hidden state $\mathbf{h}$ and output vector $\mathbf{y}$ as:

$$\begin{aligned}
\mathbf{r}^t &= \sigma(\mathbf{W}_r \mathbf{x}^t + \mathbf{U}_r \mathbf{h}^{t-1} + \mathbf{b}_r) \qquad (reset\ gate) \\
\mathbf{z}^t &= \sigma(\mathbf{W}_z \mathbf{x}^t + \mathbf{U}_z \mathbf{h}^{t-1} + \mathbf{b}_z) \qquad (update\ gate) \\
\tilde{\mathbf{h}}^t &= tanh(\mathbf{W}_h \mathbf{x}^t + \mathbf{U}_h (\mathbf{r}^t \odot \mathbf{h}^{t-1}) + \mathbf{b}_h) \\
\mathbf{h}^t &= \mathbf{z}^t \odot \mathbf{h}^{t-1} + (1 - \mathbf{z}^t) \odot \tilde{\mathbf{h}}^t \qquad (hidden\ state) \\
\mathbf{y}^t &= \sigma(\mathbf{W}_o \mathbf{h}^t + \mathbf{b}_o)
\end{aligned} \tag{1}$$

where $\sigma(\cdot)$ are element-wise sigmoid functions and $\mathbf{y}^t \in [0, 1]^L$, L is the number of sound events. Each dimension of $\mathbf{y}^t$ means the probability of a certain sound event happening at time t. Suppose the hidden state of GRU has a dimension of d, then $\mathbf{r}^t, \mathbf{z}^t, \mathbf{h}^t, \tilde{\mathbf{h}}^t \in \mathbb{R}^d$. In our experiments, we exploit bidirectional GRU (BGRU) since it makes full use of the context information from both directions. The binary cross entropy loss (BCE loss) is used for end-to-end training:

$$loss(\mathbf{y}^t, \hat{\mathbf{y}}^t) = -\frac{1}{L} \sum_{i=1}^{L} [\hat{y}_i^t log(y_i^t) + (1 - \hat{y}_i^t) log(1 - y_i^t)] \tag{2}$$

where $\hat{\mathbf{y}}^t \in \{0, 1\}^L$ is the binary indicator of sound events.

2.3. Data augmentation

To further address the data scarcity issue of SED, we augment the training data by introducing deformations that have been used in MIR tasks [4] and sound classification tasks [1]. We do not augment the validation or the test sets and it is important to mention that all the following deformations act on the raw audio recordings. We consider the following three types of augmentation strategies:

- *Pitch Shift Deformation*: tune the pitch while keeping the duration unchanged. Each recording is pitch-shifted by 14 times: $\{\pm 0.5, \pm 1, \pm 1.5, \pm 2, \pm 2.5, \pm 3, \pm 3.5\}$ semitones. This augmentation enlarges the dataset by 14 times.
- *Time Stretch Deformation*: slow down or speed up the audio recording while keeping the pitch unchanged. Each sample is time-stretched by 10 factors: {0.71, 0.76, 0.81, 0.87, 0.93, 1.07, 1.15, 1.23, 1.32, 1.41}. This augmentation enlarges the dataset by 10 times.
- *Union Deformation*: for each recording, generate both the pitch shifted counterparts with 14 semitones and the time stretched with 10 factors. This augmentation enlarges the dataset by 24 times.

3. EXPERIMENTS

3.1. Dataset and setup

We experiment on the development dataset of task3 in DCASE2017 challenge [2]. We make use of single layer BGRU with hidden size of 16. Dropout with probability of 0.5 is applied on the output of the hidden activations. We set initial learning rate to 0.001 and optimize the BCE loss with Adam [10]. During training, each recording is splitted into feature sequences of length 25 with hop size of 5. While during testing, recordings are splitted into 25-long sequences without overlap, and we threshold the outputs with a fixed value of 0.5 to mark whether the given sound events are active. We follow the official partition of cross validation and randomly select 5 recordings out of the 18 training recordings for validation, the

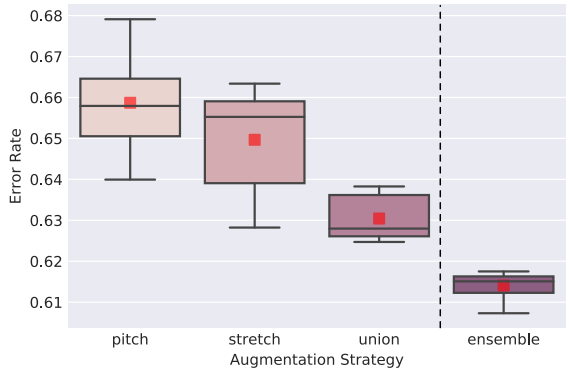

Figure 2: Error rates of different augmentation strategies for the BGRU model.

best model is saved during training process according to the performance on the validation set. Early stopping is applied to prevent over-fitting. Since the output of our model is frame-wise prediction at a resolution of 0.02 seconds, which exhibits noise and instability, we first apply a sliding window to compute the average of the model's outputs and then follow [11] to process the binary predictions with a median filter. By this means, we can partly filter out the noise and smooth the outputs.

3.2. Metrics

We obey the official requirements to evaluate our system with the error rate (ER) calculated on one second long segments [5]. Moreover, as the development dataset has been partitioned into 4 folds for cross validation, results of every testing fold are collected to produce a single evaluation to avoid biases caused by data imbalance across different folds [6].

3.3. Feature selection

We intend to use various types of features. However, the combinations of different features may have diverse performances. Consequently, we experiment with different kinds of feature combinations to figure out the most appropriate one. It is worth to mention that experiments in this section are carried out without data augmentation, we only augment the training data after we find the best feature combination. We conduct experiments with many types of combinations ranging from single feature to combinations of triple features. For each feature combination, we run the experiments for 10 times and calculate the mean and standard deviation of their error rates. Results are shown in Table.1. We can easily figure out that the $mfcc$ features always outperform lms features by a large margin, and when we integrate $mfcc$ and $pitch$ features, the best performance is achieved; Another important conclusion we can draw from Table.1 is that the BGRU model always performs better than the BLSTM model. Thus, we use BGRU as our final model for task3 in DCASE2017.

Table 2: Error rates comparison of our proposed system and other top systems in task3 of DCASE2017.

Methods	Evaluation	Development
Convolutional RNN [12]	$\mathbf{0.7914}$	$\mathbf{0.25}$
Multiple-Input CNN [13]	0.8080	0.51
Ours	0.8251	0.614 ± 0.003
Multi-Channel LSTM [14]	0.8526	0.66
CNN [15]	0.8575	0.81
Baseline [16]	0.9358	0.69

3.4. Data augmentation

We make use of the BGRU model with $mfcc$ and $pitch$ feature combination. The error rates of different data augmentation strategies are presented in Fig.2. Complying with previous experiments, we also run each experiment for 10 times and draw a boxplot to show the error rates. As can be seen, the best single model is obtained by applying the union data augmentation strategy, with an error rate of $\mathbf{0.631 \pm 0.008}$.

Our final system makes an ensemble of the above setting: we train four models with the same setting except that the validation datasets are randomly selected from the training set. During testing, the frame-wise probability prediction of the four models are averaged before post-processing. By this means, we decrease the error rates to a new level and makes the prediction quite stable. The final error rate reported is $\mathbf{0.614 \pm 0.003}$, which exceeds the officially published baseline by $\mathbf{11}\%$.

We exhibit error rates of the top 5 systems in task3 of DCASE2017, their performances on both the evaluation set and development set are shown in Table. 2. We ranked No.3 among all the 13 teams which reflect the effectiveness of our proposed system. The No.1 system makes use of convolutional RNN to make use of CNN's feature learning capacity [12] while the No.2 system incorporates features from multiple scales to improve the performance [13]. These works are meaningful inspirations for our future research.

4. CONCLUSIONS

In this paper, we proposed to exploit the bidirectional GRU for SED in real life audio recordings. By exploring the most appropriate feature combination and applying data augmentation techniques, we achieved a prominent performance improvement on the development dataset of task3 in DCASE2017 and ranked No.3 on the final evaluation set. In future work, we will focus on the design of new structures that better capture the evolving characteristics of sound event data.

5. REFERENCES

[1] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017.

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

[3] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[4] B. McFee, E. J. Humphrey, and J. P. Bello, "A software framework for musical data augmentation." in *ISMIR*, 2015, pp. 248–254.

[5] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016.

[6] G. Forman and M. Scholz, "Apples-to-apples in cross-validation studies: Pitfalls in classifier performance measurement," *SIGKDD Explor. Newsl.*, vol. 12, no. 1, pp. 49–57, November 2010.

[7] S. Adavanne, P. Pertilä, and T. Virtanen, "Sound event detection using spatial features and convolutional recurrent neural network," in *Proc. of ICASSP*, 2017.

[8] J. O. Smith, *Spectral Audio Signal Processing*. `http://-ccrma.stanford.edu/~jos/sasp/`, online book, 2011 edition.

[9] E. Cakır, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *arXiv preprint arXiv:1702.06286*, 2017.

[10] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[11] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multi label deep neural networks," in *Neural Networks (IJCNN), 2015 International Joint Conference on*. IEEE, 2015, pp. 1–7.

[12] S. Adavanne and T. Virtanen, "A report on sound event detection with different binaural features," DCASE2017 Challenge, Tech. Rep., September 2017.

[13] I.-Y. Jeong, S. Lee, Y. Han, and K. Lee, DCASE2017 Challenge, Tech. Rep., September 2017.

[14] J. Zhou, "Sound event detection in multichannel audio LSTM network," DCASE2017 Challenge, Tech. Rep., September 2017.

[15] Y. Chen, Y. Zhang, and Z. Duan, "DCASE2017 sound event detection using convolutional neural network," DCASE2017 Challenge, Tech. Rep., September 2017.

[16] T. Heittola and A. Mesaros, "DCASE 2017 challenge setup: Tasks, datasets and baseline system," DCASE2017 Challenge, Tech. Rep., September 2017.