CONVOLUTIONAL RECURRENT NEURAL NETWORKS FOR RARE SOUND EVENT DETECTION

Emre Çakır and Tuomas Virtanen

Tampere University of Technology
Finland
emre.cakir@tut.fi

ABSTRACT

Sound events possess certain temporal and spectral structure in their time-frequency representations. The spectral content for the samples of the same sound event class may exhibit small shifts due to intra-class acoustic variability. Convolutional layers can be used to learn high-level, shift invariant features from time-frequency representations of acoustic samples, while recurrent layers can be used to learn the longer term temporal context from the extracted high-level features. In this paper, we propose combining these two in a convolutional recurrent neural network (CRNN) for rare sound event detection. The proposed method is evaluated over DCASE 2017 challenge dataset of individual sound event samples mixed with everyday acoustic scene samples. CRNN provides significant performance improvement over two other deep learning based methods mainly due to its capability of longer term temporal modeling.

Index Terms— Sound Event Detection, Convolutional Neural Network, Recurrent Neural Network, Machine learning

1. INTRODUCTION

The aim of sound event detection (SED) is to temporally locate and label the sound event class(es) present in an acoustic signal. For an SED task, a set of target sound event classes should be determined. For instance, an SED task can be defined as the detection of dog barking, door bell, and baby crying sounds for any given acoustic signal. Recently, SED has been utilized in application areas such as wildlife bird audio monitoring [1, 2], audio surveillance [3], and multimedia event detection [4].

Recently, the research on SED has been mainly shifted from traditional classifier approaches such as Gaussian mixture models (GMM) - hidden Markov models (HMM) to deep learning based methods such as feed-forward neural networks (FNN) [5, 6], convolutional neural networks (CNN) [7], recurrent neural networks (RNN) [8], and convolutional recurrent neural networks (CRNN) [2, 9]. Feed-forward neural networks have the benefit of higher expressional capability over nonlinear functions compared to GMM-HMMs. However, their drawback is the fixed connections (each weight is connected to a fixed pair of neurons) which makes them less robust to slight spectral shifts in the acoustic features of the same sound event class. These slight shifts are a major factor in the inherent acoustic variability of sound event classes. This problem has mainly been overcome with the introduction of CNNs for SED, however the temporal context that can be modeled with CNNs is rather short. CRNN combines the long-term modeling capabilities of gated recurrent unit (GRU) [10] layers and the robustness of CNN to small spectral shift variations.

There are several difficulties on developing SED systems to be utilized in real-life environments. Some of these can be listed as the inherent acoustic variability of the sounds belonging to the same event class, overlapping (simultaneously occurring) sound events, environmental noise, variability in the acoustic characteristics of the background acoustic scene, and rarely occurring sound events.

The main problem encountered with the detection of the rare sound events using neural networks is the data imbalance. To elaborate, in an SED task, the classifier is trained to learn the relationship between the target class and its input representation, which is composed of acoustic features extracted in short time frames of an acoustic signal. During training, the classifier makes an estimation for the class presence probabilities for each frame, and calculates the error in the estimation through a loss function (which will be used to update the classifier parameters). In a rare SED task, the target class is not present in a significantly higher portion of time frames of each signal. Unless the training procedure of the classifier is adjusted correspondingly, the classifier will be biased on predicting ”non-present” for all the frames, because it will reach low error even if it fails to detect the frames where the target class is present. Data imbalance is a very common problem in machine learning and methods such as data augmentation using time stretching and block mixing [8], oversampling [11] and synthesizing new samples through generative methods [12] have been previously proposed to limit the negative effect of data imbalance.

In this work, we propose to utilize CRNNs for combined single-class, rare SED in the presence of a real-life acoustic scene in the background. The convolutional layers of CRNN are used to extract shift invariant features from the input time-frequency representation. The gated recurrent layers are especially effective in detecting rare sound events, because they can reset and update their hidden/cell state to distinguish the features from a small number of consecutive time frames (corresponding to a rare target event) which are noticeably different from the features from the rest of the acoustic signal (corresponding to the background). The proposed CRNN method has been previously shown to provide state-of-the-art accuracy in both real-life and synthetic SED datasets [9] and QMUL bird audio detection challenge 2017 [2]. We follow the similar CRNN architecture and procedure as in [9], with the exception that we train separate CRNNs for each class due to the combined single-class approach. In addition, we slightly adjust the training procedure according to the evaluation metric of the given SED task (see Section 3.2). This work has a companion website at [1].

The rest of the paper is organized as follows. The acoustic features and the proposed CRNN method is explained in Section 2. In Section 3, the acoustic material, evaluation metric and the eval-

[1] www.cs.tut.fi/~cakir/DCASE2017

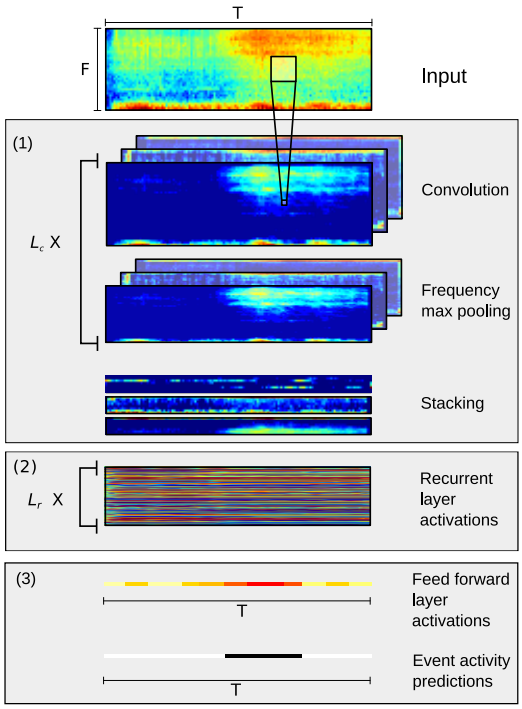

Figure 1: Overview of the proposed CRNN. (1): Multiple convolutional layers with max pooling in frequency axis, and stacking of the features over frequency axis (2): Gated recurrent layers, (3): feed-forward layer produces the event activity probabilities which are then binarized in evaluation/usage case.

uation results of the proposed method compared with the baseline methods is presented. Finally, our conclusions on this work are presented in Section 4.

2. METHOD

2.1. System Overview

The used SED approach consists of sound representation and frame-wise classification stages. In the sound representation stage, frame-level acoustic features are extracted for each time frame in the acoustic signal to obtain a feature matrix $\mathbf{X} \in \mathbb{R}^{F \times T}$, where $F \in \mathbb{N}$ is the number of features per frame and $T \in \mathbb{N}$ is the number of frames in the acoustic signal. In the classification stage, the task is to estimate the probabilities $p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta})$ for target output vector $\mathbf{y} \in \mathbb{R}^T$, where $\mathbf{y}$ denotes the probability of the target event in each frame and $\boldsymbol{\theta}$ denotes the parameters of the classifier. Once the method is to be evaluated or utilized in a usage case, the event activity probabilities are typically binarized by thresholding, *e.g.* over a constant, to obtain binary event activity predictions $\hat{\mathbf{y}} \in \mathbb{R}^T$.

The classifier parameters $\boldsymbol{\theta}$ are trained by supervised learning, and the target outputs $\mathbf{y}$ are obtained from the onset-offset annotations of the sound event class. If the sound event class is present during frame t, then $\mathbf{y}_t$ will be set to 1, and 0 otherwise.

In this work, SED is conducted in combined single-class manner, so the stages below are repeated separately for each class.

2.2. Acoustic Features

The acoustic features used in this work are log mel-band energies, as they have been shown to provide good performance on SED with deep neural networks [2, 6, 9]. Each audio sample is divided into 40 ms frames with 50% overlap and 40 log mel-band energy features are extracted from the magnitude spectrum of each frame. Each feature is then normalized independently to zero mean and unit standard deviation by using statistics calculated from the training data.

2.3. CRNN Architecture

The CRNN architecture used in this work consists of three main blocks: (1) convolution block, (2) recurrent block, and (3) classification block. The illustration of the architecture is given in Figure 1. The input for the CRNN are the acoustic features (log mel-band energies). In the convolution block, the input is fed to L_c consecutive convolutional layers with linear activation functions. Each convolutional layer is followed by batch normalization per feature map [13], a rectified linear unit (ReLU) activation function, a dropout layer [14], and a frequency domain max-pooling layer. At the end of the convolutional block, the extracted features over the CNN feature maps are stacked along the frequency axis.

Convolutional layers provide robustness to frequency shifts in the input features due to shared weight connections and max-pooling operation, and this is crucial to overcome the problem of intra-class acoustic variability for SED. However, as it has been shown previously in other works [9, 15], convolutional layers perform the the best when the filter size is small, and this means the temporal context used in these layers is very short (typically less than two hundred miliseconds).

In the recurrent block, these stacked features are fed to L_r GRU layers where tanh and hard sigmoid activation functions are used for update and reset gates, respectively. Each recurrent layer produces outputs for each frame by using both the features extracted by the convolutional layers (or the previous recurrent layers) and the previous frame activations as input. Dropout is applied on both the inputs and the hidden state outputs of the recurrent layer [16].

GRU layers control the information flow through a gated unit structure. For frame t, the total activation of GRU layer is a linear interpolation of previous activation h_{t-1} and the candidate activation $\hat{h}_t$ as

$$h_t = u_t \cdot h_{t-1} + (1 - u_t) \cdot \hat{h}_t \tag{1}$$

where u_t denotes the update gate. Candidate activation $\hat{h}_t$ is a function of h_{t-1}, the GRU layer's input x_t and the reset gate r_t. GRU activation is mainly controlled by reset gate when the GRU layer's input x_t is significantly different than in previous frames. When reset gate is closed ($r_t = 0$), the candidate activation does not include any contribution from h_{t-1}. Fast response to the changes in the input and the previous activation information is crucial for high performance in rare SED, where the task is to detect a small of consecutive time frames where target event is present.

In the classification block, a feed-forward layer of single unit with sigmoid activation function is used as the classification layer. While computing the output of the classification layer, the same weight and bias values are used over the recurrent layer outputs for each frame. The contributions of GRU's previous and candidate activations to the classification output, namely c_{t-1} and $\hat{c}_t$, can be computed as

$$\begin{aligned} c_{t-1} &= w \odot (u_t \cdot h_{t-1}) \\ \hat{c}_t &= w \odot ((1 - u_t) \cdot \hat{h}_t) \end{aligned} \tag{2}$$

Table 1: CRNN hyper-parameters for each target class.

Hyper-parameters	Baby cry	Glass break	Gun shot
L_c	3	3	3
pool size	(5,4,2)	(5,4,2)	(5,4,2)
L_r	1	3	1
# filters/units	96	160	32
# Parameters	520K	1750K	59K

where w is the weight vector that connects GRU layer and the classification layer, and $\odot$ denotes element-wise multiplication. The outputs of the classification layer are regarded as the presence probabilities of the target class in each frame of the audio sample.

If the model is to be evaluated or utilized in a usage case, the presence probabilities are binarized with a constant threshold of 0.5 to get the binary presence predictions. These predictions are further post-processed with a median filter of length 540 ms.

3. EVALUATION

3.1. Acoustic Material

For the acoustic material, DCASE2017 challenge dataset has been used and detailed information on the dataset can be found in Section 4 of [17]. The dataset consists of samples from 15 different everyday acoustic scenes (park, home, street, cafe, train etc.), some of which are mixed with isolated recordings from at most one of the three different target sound event classes: baby crying, glass breaking and gun shot. The isolated recordings are divided into segments based on the signal energy levels, and the segments relevant to the target class are selected by a human annotator. Mixing is done by adding a segment to the 30-second long background acoustic scene sample with a random time offset. The mean duration of the isolated target sound event recordings is below 2.25 seconds for all three classes and each isolated event is present at most once for each mixed sample, making them active for only a short period of time (hence the task name rare sound event detection).

For the development set, 2973 training, 298 validation and 1496 test samples (4767 total) are generated through the code repository provided as a part of the DCASE challenge [18]. Although the probability of including isolated recordings in each mixed sample is set to 0.5 as default in the code provided by the challenge, we increase the probability of including target events from default 0.5 to 0.99 for training and validation samples. This change increases the percentage of the frames labeled as including a target event from 5% to 8% in the training data, which helps to ease the problem of data imbalance. This probability is kept at 0.5 for the test samples, as suggested by the challenge organizers, to be able to compare the development set results over the same conditions with other participants. In the evaluation set, the training and validation samples of the development set are combined into a single training set, test samples are used as the validation set, and the system is evaluated against an unseen set of 1500 samples (500 for each target class).

3.2. Procedure and Final Configuration

The CRNN is trained using Adam method for gradient based optimization [19]. Cross-entropy is used as the loss function. The network is trained for a maximum of 200 epochs. After each epoch of training, validation set is evaluated for the event-based error rate (see Section 3.4) and the model at the epoch with the lowest error rate is saved in the memory. This way, we aim to align the training procedure with the evaluation metric of this work. If the error rate does not decrease for 25 consecutive epochs, the training is stopped and the last saved model is selected as the final model.

In order to decide the architecture to be used in the evaluation, we run a hyper-parameter grid search and pick the architecture with the lowest event-based error rate on the test set of the development data. The fixed hyper-parameters for each experiment is as follows. We use 5-by-5 size feature maps in convolutional layers, and dropout with probability 0.25 for both convolutional and recurrent layers. The grid search covers the number of convolutional feature maps (filters) / RNN hidden units (both are set to the same value) {32, 96, 160}; the number of recurrent layers {1, 2, 3, 4}; and the number of CNN layers {1, 2, 3 ,4} with the following frequency max-pool sizes after each convolutional layer {(8), (4, 2), (2, 2, 2), (5, 2, 2), (5, 4, 2), (5, 2, 2, 1), (5, 2, 2, 2)}. The best performing CRNN hyper-parameters for each target class are listed in Table 1.

3.3. Baseline

In this work, we compare the performance of CRNN with two baseline methods using deep learning with the same input features. The first baseline method is a deep FNN with two hidden layers of 50 units, which is also the official baseline method for the challenge. The input features differ slightly in the sense that the extracted 40 log mel-band energy features are concatenated for five consecutive frames to gather temporal context, creating a feature vector with 200 entries. The second baseline method is the CNN. While selecting the CNN architectures to be used in evaluation, a very similar grid search procedure has been applied as explained in Section 3.2, the only difference being that the recurrent layers of the CRNN are replaced with the feed-forward layers to obtain CNN architecture.

3.4. Evaluation Metric

The official evaluation metric used in DCASE2017 challenge task 2 is the event-based error rate (ER) with onset tolerance of 500 ms. ER is the sum of insertion, deletion and substitution rates. ER is calculated as explained in detail in [20].

3.5. Results

The models used in the evaluation (hence the challenge submission) have been selected as the following. As a part of hyper-parameter grid search, 84 experiments have been run on development data for CNN and CRNN each. The evaluation models are then selected based on ER on test set of development data. We present four different CRNN methods for the rare SED challenge which are labeled as:

- CRNN-1: the architecture with the lowest ER on average over three classes. This model also happens to have the lowest ER on the "baby cry" class, and its parameters are given in the corresponding column of Table 1.
- CRNN-2: the ensemble of the seven best architectures with the lowest ER on average over three classes.
- CRNN-3: the architecture with the lowest ER for each of the three classes. The parameters for each of the architectures are given in Table 1.

Table 2: Event-based error rate for the baseline FNN and CNN methods and the proposed CRNN on the test set of evaluation data. Method indices are explained in Section 3.5.

Method	Evaluation			
	Baby cry	Glass break	Gun shot	Average
FNN	0.80	0.38	0.73	0.64
CNN-1	0.46	0.13	0.58	0.39
CNN-2	0.38	0.15	0.53	0.35
CNN-3	0.46	0.14	0.55	0.39
CNN-4	0.42	0.14	0.53	0.36
CRNN-1	0.27	**0.07**	**0.20**	0.18
CRNN-2	**0.18**	0.10	0.23	**0.17**
CRNN-3	0.27	0.14	0.47	0.29
CRNN-4	0.21	0.11	0.24	0.19

- CRNN-4: the ensemble of seven best architectures with the lowest ER for each class.
- CNN methods have been obtained in the same fashion to CRNN methods as explained above.

The ensemble method is conducted as follows. Among the seven selected architectures, if four or more predict that the target class is not present in a given sample, then the final decision on the sample is that the target class is not present. Otherwise, the onset and offset values of the target class are selected as the median of the predicted onset and offset values for the sample. This ensemble method is used in order to get more reliable predictions over the onset and offset values and to filter the outlier predictions among the architectures with lowest ER.

The event-based ER results for the proposed and baseline methods have been presented in Table 2. CRNNs clearly provide better performance compared to both baseline methods for all target classes. In addition, by utilizing ensemble methods for both CNN and CRNN, the performance can be further improved, however this comes with an increased computational cost due to running several architectures in parallel. With the experiments for CRNN-1 method, we aimed to show if it is possible to find a single architecture that performs well for all three classes. For the development data, CRNN-1 provides comparable performance (0.16 vs 0.14 ER) with CRNN-3, where the best architecture is selected for each target class. For the evaluation data, as presented in Table 2, CRNN-1 performs even better than CRNN-3.

Regardless of the method, highest performance is obtained for glass break. Although the best performing architectures for each class differ significantly in the number of parameters (see Table 1), the median number of parameters among the seven best architectures are 687K, 806K and 774K for baby cry, glass break and gun shot, respectively. Therefore it is not possible to draw any direct conclusions on the relationship between the target class and the best performing architecture size.

3.6. An insight on GRU layer of CRNN

A case-study demonstration of the effect of GRU layers on the CRNN outputs is given in Figure 2. The CRNN architecture used to create this illustration consists of three convolutional layers and one GRU layer with 32 filters/units each, followed by a single unit classification layer. In panel (a), we can see that multiple GRU units respond to the change of input features around 4.5 second mark and trigger the candidate activation, as the target event starts to appear in the audio signal. After that, the GRU contribution is mainly controlled by the previous activations while the target event is still present, as shown in panels (b) and (c). Finally, the CRNN produces almost perfect detection of onset and offset for the given target event, as shown in panel (d).

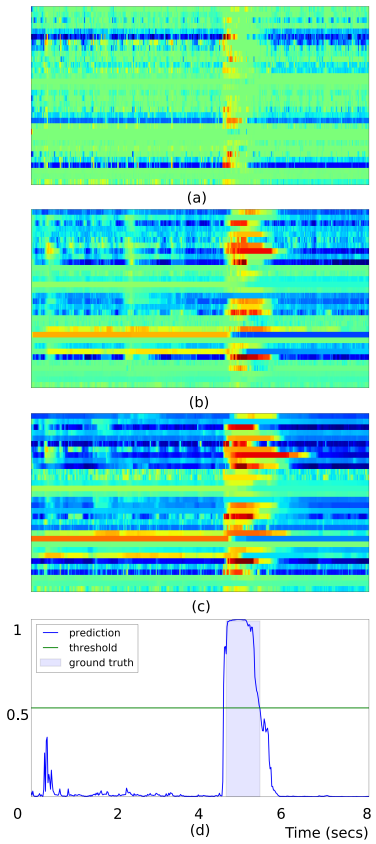

Figure 2: (a): contribution of current (candidate) activation $\hat{c}_t$, (b): contribution of previous activation c_{t-1}, (c): total contribution of GRU layer to the classification activation; (d): event activity probabilities vs. time for the first eight seconds of sample *devtest_babycry_001_1128b63726e9ed59ddc1bb944b3f22ce.wav*.

4. CONCLUSIONS

In this paper, CRNN has been proposed for rare SED. CRNN has provided significantly improved performance over FNNs and CNNs for every target sound event class in DCASE 2017 challenge dataset. It is shown that the performance can further be improved using ensemble methods. For future work, improved ways to incorporate the evaluation metric into training procedure as the objective function can be considered. For instance, instead of aiming to directly match the target output and the predicted output for each frame, the objective function can be calculated over a window of frames, especially for the case when the onset and offset times can be tolerated to a certain degree.

5. REFERENCES

[1] D. Stowell and D. Clayton, "Acoustic event detection for multiple overlapping similar sources," in *2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2015, pp. 1–5.

[2] E. Cakir, S. Adavanne, G. Parascandolo, K. Drossos, and T. Virtanen, "Convolutional recurrent neural networks for bird audio detection," in *European Signal Processing Conference (EUSIPCO)*, 2017, accepted.

[3] P. Foggia, N. Petkov, A. Saggese, N. Strisciuglio, and M. Vento, "Reliable detection of audio events in highly noisy environments," *Pattern Recognition Letters*, vol. 65, pp. 22–28, 2015.

[4] Y. Wang, L. Neves, and F. Metze, "Audio-based multimedia event detection using deep recurrent neural networks," in *2016 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 2742–2746.

[5] I. Choi, K. Kwon, S. H. Bae, and N. S. Kim, "DNN-based sound event detection with exemplar-based approach for noise reduction," DCASE2016 Challenge, Tech. Rep., September 2016.

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

[7] T. Lidy and A. Schindler, "CQT-based convolutional neural networks for audio scene classification and domestic audio tagging," DCASE2016 Challenge, Tech. Rep., September 2016.

[8] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," in *2016 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 6440–6444.

[9] E. Cakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1291–1303, 2017.

[10] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8)*, 2014.

[11] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: a new over-sampling method in imbalanced data sets learning," *Advances in intelligent computing*, pp. 878–887, 2005.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[13] 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*, 2015, pp. 448–456.

[14] 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.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[16] Y. Gal, "A theoretically grounded application of dropout in recurrent neural networks," *Advances in neural information processing systems*, 2016.

[17] 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 (DCASE2017)*, November 2017, submitted.

[18] T. Heittola. (2016) Dcase2017 baseline system. [Online]. Available: github.com/TUT-ARG/DCASE2017-baseline-system

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

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