ACOUSTIC SCENE CLASSIFICATION USING CONVOLUTIONAL RECURRENT NEURAL NETWORKS

Hugo Jallet, Emre Çakır, Tuomas Virtanen

Tampere University of Technology
Korkeakoulunkatu 1
33720, Tampere, FINLAND
{hugo.jallet, emre.cakir, tuomas.virtanen} @tut.fi

ABSTRACT

This paper presents an application of a convolutional recurrent neural network (CRNN) for the task of acoustic scene classification (ASC). Convolutional layers of CRNN are used as high-level feature extractors and gated recurrent layers are used to model the long term temporal context of the acoustic samples. The developed methods are evaluated using the 2017 edition of the "Detection and Classification of Acoustic Scenes and Events" (DCASE) challenge task 1 and consequently tested on the datasets provided for the task of ASC. In this paper, we use two CRNN-based methods which score an overall accuracy of 78.9% and 80.8% compared to baseline feed-forward neural network with 74.8% accuracy.

Index Terms— acoustic scene classification, deep neural networks, convolutional neural networks, recurrent neural network

1. INTRODUCTION

Acoustic scene classification (ASC) is a research area which refers to the recognition of an audio context from a recording. Audio context can be defined as the ensemble of sound events and background noises associated to a particular environment, e.g. a beach or a train. ASC has applications for example in context awareness devices. An example of such technology could be the automatic adjustment of a functionality (phone ringtone etc.) depending on the context.

A vast majority of approaches in ASC used to be based on hand-crafted features [1], in order to facilitate discrimination between all the acoustic classes involved. However, such features are an obstacle for ASC performance and it should be admitted that derived features should be more powerful. Deep learning methods have brought great advances in the field of statistical pattern recognition [2], thus becoming a strong alternative to the hand-crafted features. This methods offer great tools to automatically learn features from raw input data, and are consequently fit for ASC. Convolutional neural network (CNN) has been proposed previously for ASC [3, 4]. In addition, methods combining CNNs with I-vector representations [5] and also source separation based methods such as non-negative matrix factorization (NMF) [6] has been proven to be very effective in ASC.

In this paper, we propose to use convolutional recurrent neural networks (CRNN) for ASC. Convolutional layers of CRNN extract local, small shift-invariant features from a time-frequency representation of the acoustic sample. Meanwhile, gated recurrent layers of CRNN utilize the cues extracted from the features of the previous frames that are relevant to the given task. ASC often requires this sort of long term temporal modeling, as the acoustic scenes are defined by the collection of several sound events happening during different time periods of the acoustic samples. This CRNN method has not been utilized before for ASC but has proven its efficiency for other tasks as sound event detection [7].

The remaining of the article is organized as follows. The acoustic features used to represent the signals and the CRNN architecture is addressed in Section 2. The Section 3 presents the acoustic material, the evaluation metrics and the evaluation results compared to the baseline system of the DCASE 2017 ASC challenge. Last, the conclusions of the authors are exposed in Section 4.

2. METHOD

This method has been developed for the task 1 of DCASE 2017 challenge. The goal of this task was to classify a test recording into one of the 15 predefined class (beach, bus, office etc.) that characterize the environment where it was recorded.

2.1. Acoustic features

The acoustic features used in this work are log mel-band energies, similar to the SED task described in [7]. Each audio sample is divided into 40 ms frames with 50% overlap and 40 log mel-band energy features are extracted for 501 frames per acoustic sample.

2.2. CRNN Architecture

The CRNN architecture chosen for this work is composed of three main blocks : (1) convolution block, (2) recurrent block, (3) classification block. An illustration of the architecture is presented in Figure 1. The input of the network is the acoustic features (log-mel band energies here) of an audio sample.

In the convolution block, the input is fed to L_c consecutive convolutional layers with 5-by-5 feature maps and linear activation functions. We use *same* convolution in each layer, i.e. the input features are padded with the zeros to the length of the feature map, so that the size of the convolutional layer outputs is not reduced due to convolution operation. Each convolutional layer is followed by batch normalization [8] per feature map, a rectified linear unit (ReLU) activation function, a dropout layer [9] with dropout probability of 0.25, 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, i.e. the

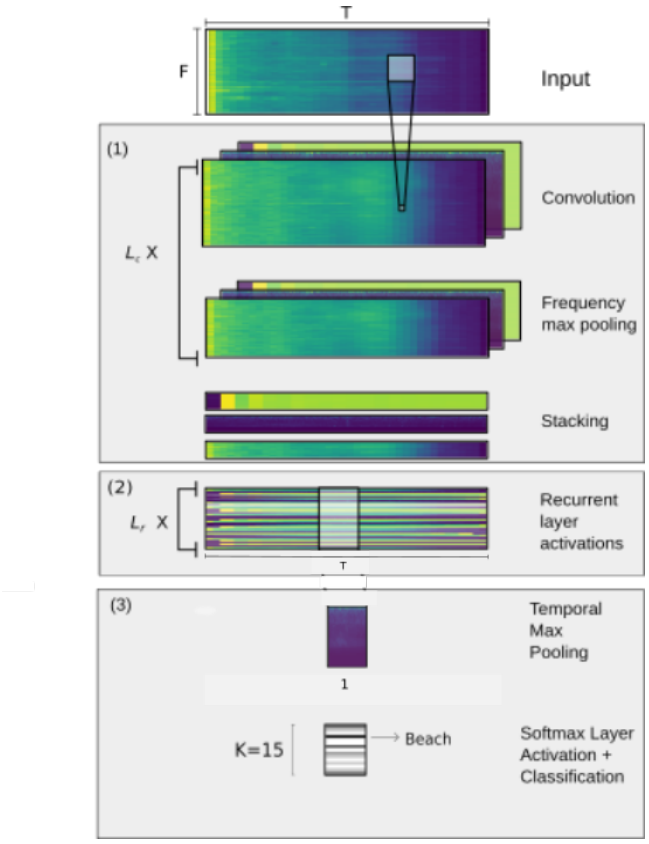

Figure 1: Overview of the submitted CRNN method. (1): Multiple convolutional layers with max-pooling and stacking of the features over frequency axis, (2):Multiple stacked recurrent layers, (3):Temporal max-pooling and Softmax layer permits to classify the audio sample into one class, just with one vector.

features extracted by all the feature maps of the last convolutional layer are concatenated into a single feature vector for each frame.

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. Dropout with probability 0.25 is applied on both the inputs and the recurrent outputs of the recurrent layer.

Finally, in the classification block, a temporal max-pooling layer is used to obtain one output per last GRU layer hidden unit that represents all the frames in the acoustic sample. The output of this layer is then fed to a feed-forward network of 15 units (1 unit per class) with softmax activation function as the classification layer. The outputs of the classification layer are regarded as the probabilities of the sample to belong to a class (given that one unit corresponds to one class). If the method is to be evaluated or utilized in a usage case, the acoustic sample is estimated to belong to the class with the highest probability.

2.3. Procedure and Final Configuration

The CRNN is trained using Adam method for gradient based optimization [10]. Binary cross-entropy is used as the cost function, and the network is trained for a maximum of 200 epochs. Early stopping is used with a patience of 25 epochs and a delta of 0.001, which means that the training is prematurely stopped if the binary accuracy did not improve at least for 0.1% during 25 consecutive epochs. We keep the model from the epoch with the maximum validation accuracy after training.

Table 1: Hyper-parameters of the model (CRNN-1) used for the challenge, determined with the grid search.

Parameters	Chosen Model
L_c	3
L_r	2
Pool Size	(2, 2, 2)
#Filters/Hidden Units	96
#Parameters	688K

In order to decide which architecture to use for our challenge submission, we run a hyper-parameter grid search and pick the architecture which gives the highest classification accuracy in the test set of the development data (see Section 3.1). The grid search covers the number of CNN feature maps / GRU hidden units $\{32,96,160\}$ (both are set to the same value); the number of convolutional and recurrent layers $\{1,2,3,4\}$ (L_c can be different from L_r); and the frequency max-pool sizes after each convolutional layer $\{(8)$; (5, 4); (2, 2, 2); (5, 2, 2); (5,4, 2); (5, 2, 2, 1); (5, 2, 2, 2)$\}$. The final hyper parameters of the network are listed in Table 1. This method is referred in this paper as CRNN-1.

We also decided to pick the seven best architectures (in terms of test accuracy on the development set) and to train these networks with the whole data from development set. Then, for each sample of the test set for the challenge, we do majority voting over test set outputs of these seven models and get the scene label for the sample. This method is referred as CRNN-2.

3. EVALUATION

3.1. Acoustic Material

The acoustic material used for this task consists of 10-second segments. These segments have been created by splitting 3-5 minutes long segments recorded in different scenes. The dataset is divided into two subsets : development set and evaluation set. The development set is actually the complete DCASE 2016 dataset for the same task. The evaluation dataset is made of newly recorded segments. Segments obtained from the same original recording are in the same set. Each class is provided with 312 of these segments. The development dataset is thus composed of 4680 segments, split into 3075 training, 435 validation and 1170 test segments. More details about the dataset can be found in [11].

3.2. Baseline

In this work, we will compare the performance of our two CRNN based methods with two baseline methods using deep learning with the same input features. The first method is a feed-forward neural network (FNN) with two hidden layers of 50 neurons. This is also the official baseline method of the challenge. The second baseline method is a CNN whose architecture has been selected through the

Table 2: Average scene accuracy over four folds for the baseline FNN, the two CRNN based methods and a simple CNN on the development dataset.

	Accuracy			
Scene	FNN	CNN	CRNN-1	CRNN-2
beach	75.3	71.2	72.4	**77.9**
bus	71.8	91.0	95.8	**96.8**
cafe/restaurant	57.7	67.9	64.1	**70.2**
car	**97.1**	89.1	91.3	92.9
city center	90.7	93.3	**93.6**	**93.6**
forest path	79.5	**98.1**	94.9	96.8
grocery store	58.7	84.3	82.7	**86.5**
home	**68.6**	63.8	**68.6**	65.7
library	57.1	**77.2**	72.1	73.4
metro station	91.7	89.7	**93.6**	93.3
office	**99.7**	87.8	89.4	88.5
park	**70.2**	63.8	64.7	69.6
residential area	64.1	**72.8**	64.4	68.6
train	58.0	46.5	**61.2**	55.1
tram	81.7	79.8	74.7	**83.3**
Overall accuracy	74.8	78.4	78.9	**80.8**

same grid search approach as explained in Section 2.3, while replacing the recurrent layers with feed-forward layers that apply the same set of weights on the features in each frame.

3.3. Metrics

The official metric used in the ASC challenge is accuracy, which is defined as the ratio between the number of correct system outputs and the total number of outputs. The system is always tested with the provided four fold cross-validation setup.

3.4. Results

The results on the development set can be found in Table 2. The authors acknowledge that the conclusions/comments in this section have been made based on the assumption that the results over the test sets of development and evaluation datasets would show similar characteristics, and this section will be updated once the evaluation results are published.

The two CRNN methods have the best accuracy of the four methods, but just shows a little improvement compared to the FNN (4.1% and 6% respectively), and what we can call an incremental improvement when compared to the CNN (0.5% & 2.4%). Consequently, the CRNN-2 method is more accurate than the CRNN-1, scoring the best overall accuracy of 80.8%. Moreover, the CRNN-1 method does not make the best scores for every classes, even being overtaken by the FNN which make the best score for four of the classes (three for the CNN and four for the CRNN-1).The CRNN-2 also make the best score for six classes (different from the FNN's best classes).

4. CONCLUSIONS

In this paper, the authors have proposed two different ways of dealing with the task of Acoustic Scene Classification using CRNN networks. This methods show an accuracy improvement compared to a two-layers FNN and a simple CNN. However, these improvements are less important than what we expected, the CRNN method being proven very efficient in other audio related tasks. In the future, enhancements could be made by changing some parameters of the training or by finding an alternative to a simple network, as our proposed method CRNN-2.

5. ACKNOWLEDGMENT

The authors would like to thank Toni Heittola for creating a baseline code repository for the challenge. The research leading to these results has been conducted with the funding from the European Research Council under the European Unions H2020 Framework Programme through ERC Grant Agreement 637422 EVERYSOUND. The authors also wish to acknowledge CSC-IT Center for Science,Finland, for computational resources.

6. REFERENCES

[1] D. Barchiesi, D. Giannoulis, D. Stowell, and M. D. Plumbley, "Acoustic scene classification: Classifying environments from the sounds they produce," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 16–34, 2015.

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[3] M. Valenti, S. Squartini, A. Diment, G. Parascandolo, and T. Virtanen, "A convolutional neural network approach for acoustic scene classification," *2017 International Joint Conference on Neural Networks (IJCNN)*, may 2017.

[4] D. Battaglino, L. Lepauloux, and N. Evans, "Acoustic scene classification using convolutional neural networks," DCASE2016 Challenge, Tech. Rep., September 2016.

[5] H. Eghbal-Zadeh, B. Lehner, M. Dorfer, and G. Widmer, "CP-JKU submissions for DCASE-2016: a hybrid approach using binaural i-vectors and deep convolutional neural networks," DCASE2016 Challenge, Tech. Rep., September 2016.

[6] V. Bisot, R. Serizel, S. Essid, and G. Richard, "Supervised nonnegative matrix factorization for acoustic scene classification," DCASE2016 Challenge, Tech. Rep., September 2016.

[7] E. Çakır, 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, jun 2017.

[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*, 2015, pp. 448–456.

[9] 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.

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

[11] 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.