INVESTIGATING TEMPORAL AND SPECTRAL SEQUENCES COMBINING GRU-RNNs FOR ACOUSTIC SCENE CLASSIFICATION

Technical Report

Eleftherios Fanioudakis

Greece
eleftherios.fanioudakis@gmail.com

Anastasios Vafeiadis

Greece
anasvaf@gmail.com

ABSTRACT

This report describes our contribution to Task 1A of the 2020 Detection and Classification of Acoustic Scenes and Events (DCASE) challenge. We investigated the use of bi-directional Gated Recurrent Unit (GRU) - Recurrent Neural Networks (RNNs) in order to capture the spectral and temporal information of the input signal. The GRU-RNNs are used as an ensemble during training, having equal weights for the time and the frequency sequences. Our architecture is based on a Convolutional Recurrent Neural Network (CRNN), where the short-time Fourier magnitude spectrogram is used as an input to the network. By exploiting the mixup augmentation technique, randomly selecting the mixup coefficient α for every sample, and down-sampling the original signal from 44.1 kHz to 4 kHz, we achieved an average class accuracy of 65.4%. Since most of the information of the environmental sound signals was found in the lower frequencies, a CRNN model ensemble was performed, combining 4 and 8 kHz as the sampling frequencies. The latter system's accuracy was boosted to 67.3%, a 24.4% increase over the development set baseline.

***Index Terms*—** Acoustic scene classification, convolutional recurrent neural networks, STFT spectrograms

1. PROPOSED ACOUSTIC SCENE CLASSIFICATION SYSTEMS

Task 1A of the DCASE 2020 challenge [1] is not only related to the basic problem of acoustic scene classification, in which an audio recording is required to be classified into an acoustic scene class, but it focuses on the generalization property of systems across a number of different devices. The main challenge of this task is concerned with the different acoustic properties that a recording device can have. Some of them are about the frequency response of the microphones, especially when comparing a professional binaural microphone and a mobile device. Additionally the dynamic range compression, added to the simulated devices, can significantly affect the spectrum of the recording and in consequence the classification accuracy of the developed system.

Our proposed systems use a simple CRNN architecture, where we take advantage of combining two bi-directional GRU-RNNs that focus on the spectral and temporal characteristics of the input signal. The selected sampling frequencies are 4 and 8 kHz for the single CRNN models and an ensemble of the two single models, where a sample-based average with weights is taken from each one.

1.1. Audio signal pre-processing

As a first step, the input signal was down-sampled to 4 and 8 kHz respectively for the single CRNN models. The main reason for down-sampling the original signal was that when examining the audio recordings, by plotting their linear spectrograms, most of the audio signal energies were found at the lower frequencies.

Furthermore, we split the 10 s recordings into 4 segments of 2.5 s, for the case of the 4 kHz sampling frequency, and into 8 segments of 1.25 s, for the case of the 8 kHz sampling frequency. This allowed to keep the same shape for both the frequency and time axis of the short-time Fourier transform (STFT) spectrogram and to decrease the computational cost of the network. The length of the Fast Fourier Transform (FFT) was 512, with a hop length of 128. We selected the Hanning window for the FFT and the resulted spectrogram was a matrix, consisting of un-normalized FFT values, with shape 257×79.

1.2. Data augmentation

For our proposed systems, two data augmentation techniques were applied. The first one was the mixup augmentation [2]. Regarding the mixup augmentation, random α values were selected for every batch. This resulted in random mixes between the classes that were equal to 1/2 of the selected batch size.

Additionally, random time-shifts were performed, given the 10 s recording. In our approach, random 2.5 s segments were used as an input to the network.

1.3. Network description

The neural network architecture that was selected for this task was a 2D CRNN with two bi-directional GRUs, for the time and frequency sequences respectively, as shown in Figure 1. Five convolutional layers were selected, each have a kernel size of 3×3 and the number of filters started from 32 and ended at 512 on the fifth convolutional layer. Each convolutional layer was followed by a batch normalization layer and a max-pooling layer with 2×2 kernel and the same stride size. The rectified linear unit (ReLU) [3] activation function was used by the convolutional layers.

After the final max-pooling operation, the frequency and time sequences were fed to two different bi-directional GRU with 512 units each. This resulted in a naive ensemble during the training network that could learn the spectral and temporal characteristics of the input signal, similarly to the one proposed by Deng et al. [4]. The hyperbolic tangent activation function was used by the GRU-RNN layers.

Finally the two separate outputs of the bi-directional GRU were each used as input to two fully connected layers with 1024 units. The activation function used by the fully connected layers was the linear activation function. Since the mixed up samples were equal to 1/2 (=32) of the selected batch size (=64) for each iteration, a combined log-loss for the two heads was calculated for our final system.

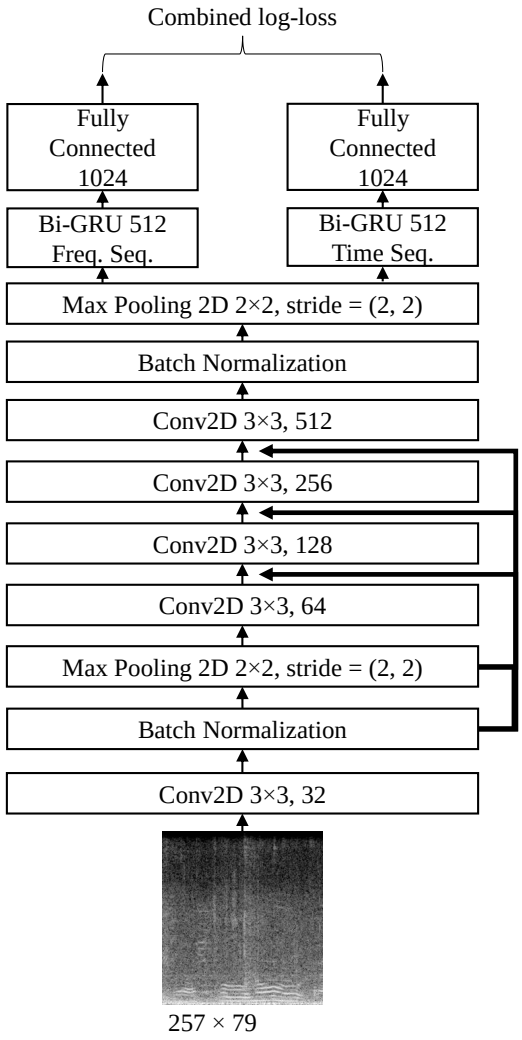

Figure 1: The proposed CRNN architecture.

For the proposed systems (single 2D CRNN with 4 kHz sampling frequency, single 2D CRNN with 8 kHz sampling frequency, ensemble of both), the Adam [5] optimizer with an initial learning rate l_r=0.001 which was reduced by a factor of 0.1, when there was no class-wise macro-average accuracy improvement for 20 consecutive epochs. The networks resulted in 20,477,140 trainable parameters and were trained on a single GTX 1080 Ti.

2. EXPERIMENTAL RESULTS

The systems were evaluated using the proposed single fold split between the training and evaluation parts of the development dataset. The class-wise and devise-wise accuracy results are presented in Table 1 and Table 2, respectively, and they are compared against a baseline [6] that uses the OpenL3 [7] embeddings as features and two fully connected layers, as the network architecture.

Our best system, which was an ensemble of 2D CRNNs on 4 and 8 kHz sampling frequencies, achieved a 24.4% increase over the baseline. We noticed that as we increased the sampling frequencies, the class-wise accuracy on the development set was significantly worse. The classes *pedestrian street* and *public square* and the simulated device six (S6) were the hardest to be classified.

Table 1: Class-wise accuracy results on the development set.

Label	**Accuracy (%)**			
	Baseline	**CRNN 4 kHz**	**CRNN 8 kHz**	**CRNN Ensemble 4&8 kHz**
Airport	45.0	57.6	58.6	59.0
Bus	62.9	88.2	73.7	85.2
Metro	53.5	67.0	61.6	70.0
Metro station	53.0	56.2	49.2	56.6
Park	71.3	82.5	84.2	83.8
Public square	44.9	33.7	42.8	37.7
Shopping mall	48.3	61.2	59.3	65.3
Street, pedestrian	29.8	46.5	39.1	47.8
Street, traffic	79.9	84.2	76.4	83.8
Tram	52.2	77.0	82.3	85.1
Average	**54.1**	**65.4**	**62.8**	**67.3**

Table 2: Device-wise accuracy results on the development set.

Device	**Accuracy (%)**			
	Baseline	**CRNN 4 kHz**	**CRNN 8 kHz**	**CRNN Ensemble 4&8 kHz**
A	70.6	76.4	74.5	77.0
B	60.6	66.7	65.2	69.1
C	62.6	70.6	67.3	71.2
S1	55.0	63.0	65.5	67.0
S2	53.3	60.3	63.3	63.9
S3	51.7	70.3	67.9	70.6
S4	48.2	64.2	55.8	66.4
S5	45.2	62.0	61.0	65.9
S6	39.6	57.0	47.5	58.2

3. CONCLUSIONS

In this report, we introduce our experimental results for the Task 1A of acoustic scene classification in the DCASE 2020 challenge. By ensembling two simple 2D CRNN models on two different sampling frequencies, our system can outperform the baseline system [6] by 24.4%.

4. REFERENCES

[1] http://dcase.community/challenge2020/.

[2] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[3] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 807–814.

[4] J. Deng, B. Schuller, F. Eyben, D. Schuller, Z. Zhang, H. Francois, and E. Oh, "Exploiting time-frequency patterns with lstm-rnns for low-bitrate audio restoration," *Neural Computing and Applications*, vol. 32, no. 4, pp. 1095–1107, 2020.

[5] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference for Learning Representations (ICLR-15)*, 2014.

[6] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in dcase 2020 challenge: generalization across devices and low complexity solutions," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, 2020, submitted. [Online]. Available: https://arxiv.org/abs/2005.14623

[7] J. Cramer, H.-H. Wu, J. Salamon, and J. P. Bello, "Look, listen and learn more: Design choices for deep audio embeddings," in *IEEE Int.˜Conf.˜on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, May 2019, pp. 3852–3856. [Online]. Available: https://ieeexplore.ieee.org/document/8682475