ACOUSTIC SCENE CLASSIFICATION USING PARALLEL COMBINATION OF LSTM AND CNN

Soo Hyun Bae, Inkyu Choi and Nam Soo Kim

Seoul National University
Department of Electrical and Computer Engineering and INMC
Gwanak P.O.Box 34, Seoul 151-744, Korea
{shbae, ikchoi}@hi.snu.ac.kr, nkim@snu.ac.kr

ABSTRACT

Deep neural networks(DNNs) have recently achieved a great success in various learning task, and have also been used for classification of environmental sounds. While DNNs are showing their potential in the classification task, they cannot fully utilize the temporal information. In this paper, we propose a neural network architecture for the purpose of using sequential information. The proposed structure is composed of two separated lower networks and one upper network. We refer to these as LSTM layers, CNN layers and connected layers, respectively. The LSTM layers extract the sequential information from consecutive audio features. The CNN layers learn the spectro-temporal locality from spectrogram images. Finally, the connected layers summarize the outputs of two networks to take advantage of the complementary features of the LSTM and CNN by combining them. To compare the proposed method with other neural networks, we conducted a number of experiments on the TUT acoustic scenes 2016 dataset which consists of recordings from various acoustic scenes. By using the proposed combination structure, we achieved higher performance compared to the conventional DNN, CNN and LSTM architecture.

Index Terms— Deep learning, sequence learning, combination of LSTM and CNN, acoustic scene classification

1. INTRODUCTION

Acoustic scene classification aims to recognize the environmental sounds that occur for a period of time. Many approaches have been proposed for acoustic scene classification including feature representation, classification models, and post-processing. The support vector machine (SVM) was one of the most successful learning model in a number of scene classification tasks. As SVM is a binary classifier, some additional methods must be combined to apply them to the multi-class problems, such as the use of tree or clustering schemes [1, 2]. Furthermore, many machine learning-based scene classification techniques were proposed in the detection and classification of acoustic scenes and events (DCASE) challenge 2013 [3, 4, 5].

However, as deep learning techniques have been widely used on various learning tasks, researchers have started to apply them to acoustic scene classification as well [6, 7]. In [8], a DNN-based sound event classification algorithm was performed with several image features.

Deep neural networks (DNNs) are powerful pattern classifier which enables the networks to learn the highly nonlinear relationships between the input features and output targets. Though the DNNs work well in the classification task, they cannot be used to map sequences to sequences because of their structural limitations. To overcome this shortcoming, recurrent neural networks (RNNs) and long short-term memory (LSTM), which is a special type of RNN, have been applied to sequence learning [9].

DNNs can only map from present input vector to output vector, whereas LSTM can map from sequence to output sequence or vector. Therefore, LSTM can learn the temporal information through consecutive input vectors. The authors in [10] and [11] proposed sound event detection techniques based on bi-directional LSTM which yielded higher performance compared to the DNNs. Unlike sound events which occur in a short time frame, acoustic scenes are maintained for relatively longer range. Thus, applying RNNs to the acoustic scene classification will improve the performance.

Other approaches were proposed to use convolutional neural networks (CNNs) with spectrogram image features (SIF) [12]. In [13], the authors addressed the importance of spectro-temporal locality and proposed a CNN-based acoustic event detection algorithm.

In this paper, we propose to combine the LSTM and CNNs in parallel as lower networks in order to exploit sequential correlation and local spectro-temporal information. In the LSTM layers, sequences of Mel-frequency cepstral coefficients (MFCCs) features are utilized as input in order to extract the sequential information. The CNN layers learn the spectro-temporal locality from SIF, and SIF clips are set to have the same length with the timestep of LSTM inputs. The outputs of the two separated layers are combined by the connected layers which are able to learn complementary features of LSTM and CNN. To compare the performance of the proposed method with various neural networks, we conducted a number of experiments on the TUT acoustic scenes 2016 dataset [14]. The results revealed that the combination of LSTM and CNN outperforms the conventional DNN, CNN and LSTM architecture with respect to classification accuracy.

2. LONG SHORT-TERM MEMORY

The key idea of RNN is that the recurrent connections between the hidden layers allow the memory of previous inputs to retain internal state, which can affect the outputs. However, RNN mainly have two issues to solve in the training phase: vanishing gradient and exploding gradient problems [15]. When computing the derivatives of activation function in the back propagation process, long-term components may go exponentially fast to zero. This makes the model hard to learn the correlation between temporally distant inputs. Meanwhile, when the gradient grows exponentially during training, the

exploding gradient problem occurs. In order to solve this problem, the LSTM architecture was proposed [16]. LSTM layers are composed of recurrently connected memory blocks in which one memory cell contains three multiplicative gates. The gates perform continuous analogues of write, read and reset operations which enable the network to utilize the temporal information over a period of time.

3. PARALLEL COMBINATION OF LSTM AND CNN

In this section, we describe our approach to improve the classification accuracy of acoustic scene. The schematic of the proposed neural networks structure can be seen in Figure 1.

3.1. Feature extraction

In the proposed system, different types of neural networks are combined in parallel. Thus, each network accept different form of input feature. The LSTM layers utilize sequence of acoustic feature, but the CNN layers use spectrogram images. As inputs for the CNN layers, the SIF are extracted from the sound spectrogram [8, 12, 17]. Firstly, a spectrogram is generated by short-time Fourier transform. Given audio frame $s(n)$ segmented by length N and Hamming window $w(n)$, the short time spectral column $\mathbf{F}(f,t)$ at time t is computed as,

$$\mathbf{F}(f,t) = \left| \sum_{n=0}^{N-1} s(n)w(n)e^{\frac{-j2\pi nf}{N}} \right| \quad (1)$$

for $f = 0, ..., N/2$. In order to generate a spectrogram image which has K-bin frequency resolution, down sampling is performed by using a window of length $W = N/2K$ as follows:

$$\mathbf{F}_{down}(f,t) = \sum_{i=0}^{W-1} \mathbf{F}(f+i,t)/W, \quad (2)$$

for $f = 0, ..., (K-1)$. Finally, a simple de-noising method is performed by subtracting each minimum frequency bin value in a frame-wise manner as follows:

$$\mathbf{F}_{dn}(f,t) = \mathbf{F}_{down}(f,t) - \min_t\{\mathbf{F}_{down}(f,t)\} \quad (3)$$

for $f = 0, ..., (K-1)$. In the proposed system, the extracted SIF has size of $K \times \tau$, where τ represents the time resolution which is also identical to the timesteps in the LSTM layers.

3.2. LSTM layers

The hidden layers of LSTM have self-recurrent weights. These enable the cell in the memory block to retain previous information. In the proposed system, τ vectors are used for sequential learning. The lower part in Figure 1 depicts how the sequences are trained through the LSTM layers. Previous $\tau - 1$ vectors and one present vector are forwarded to the recurrent layer sequentially. If the MFCC vectors from $x_{t-\tau+1}$ to x_t are used as the present inputs, vectors from $x_{t-\tau+2}$ to x_{t+1} will be used as the next input sequence. The output vector z_t^{LSTM} is extracted from input MFCC sequence x_t^{LSTM} through the LSTM layers, where $x_t^{LSTM} = [x_{t-\tau+1}, ..., x_t]$.

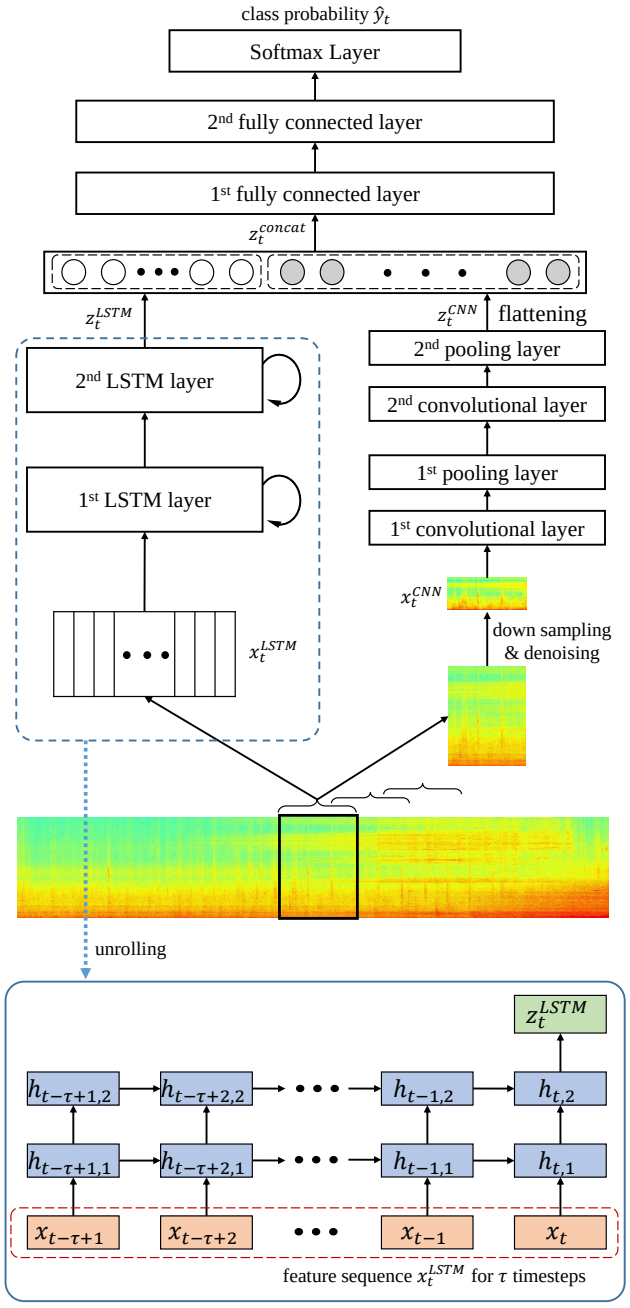

Figure 1: Neural network structure for the proposed technique.

3.3. CNN layers

From Section 3.1, SIF x_t^{CNN}, which is a $F \times \tau$ matrix, are extracted. The convolutional layer performs 2-dimensional convolution between the spectrogram image and the pre-defined linear filters. To enable the network to extract complementary features and learn the characteristics of input SIF, a number of filters with different functions are used. Thus, if we apply K different filters to the spectrogram image, K different filtered images are generated in the convolutional layer. The filtered spectrogram images are forwarded to the pooling layer which conducts down sampling. Especially,

max pooling divides the input image into a set of non-overlapping sub-regions and selects the maximum value. By reducing the spatial size of representation via pooling, the most dominant feature in the sub-region is extracted. The pooling layer operates independently on every filtered image and resizes them spatially. In the last pooling layer, the resized outputs are rearranged in order to fully connect with the upper layer. The flattened output vector z_t^{CNN} is extracted from x_t^{CNN} through the CNN layers

3.4. Connected layer of LSTM and CNN

In [18], long-term recurrent convolution network (LRCN) model was proposed for visual recognition. LRCN is a consecutive structure of CNN and LSTM. LRCN processes the variable-length input with a CNN, whose outputs are fed into LSTM network, which finally predicts the class of the input. In [19], a cascade structure was used for voice search. Compared to the method mentioned above, the proposed network forms a parallel structure in which LSTM and CNN accept different inputs separately. Concatenated vector z_t^{concat} is forwarded to the fully connected layer, where $z_t^{concat} = [z_t^{LSTM}, z_t^{CNN}]$. The connected layers can train the complementary information of LSTM and CNN. These enable the proposed model to learn the sequential information and spectro-temporal information, simultaneously. Finally, the class probability $\hat{y}_t$ is predicted through the softmax layer.

4. EVALUATION

To assess the performance of the proposed method, we conducted a number of experiments on the TUT acoustic scenes 2016 dataset which consists of recordings from various acoustic scenes. The dataset contains 1170 recordings of total 9.75 hours with 15 different classes. Audio signals sampled at 44.1 kHz sampling frequency were divided into 40 ms frames with 50% hop size. Experiments were conducted using 4-fold cross validation. The final results were obtained by averaging over all evaluation folds.

We evaluated the classification accuracy using two measures: frame-based accuracy and segment(30s)-based accuracy. Due to the softmax output layer of our networks, probability distributions among the J class labels were obtained individually. Given z_t^{concat}, the predicted class label at t frame was computed by,

$$C_{frame} = \arg\max_j P(\hat{y}_t = j | z_t^{concat}) \quad (4)$$

where j denotes class index. To obtain the class label of the entire audio segment, the likelihood was computed follows as:

$$C_{segment} = \arg\max_j \sum_{t=1}^{T} \log(P(\hat{y}_t = j | z_t^{concat})), \quad (5)$$

where T represents the number of frames in the one audio segment.

4.1. Neural networks setup

All networks in our experiments were trained using mean squared error as the loss function supervised by one-hot encoding class vectors. The randomly ordered mini-batches in each epoch was set to be 256. After a mini-batch was processed, the weights were updated using adadelta [20]. In order to mitigate the over-fitting problem in the training phase, we used the dropout technique which has already proved its regularization capability [21]. The output layer contained 15 softmax nodes identical to the number of scenes.

Table 1: Frame-based classification accuracy (%), averaged over 4-fold cross validation.

Scene	DNN	CNN	LSTM	CNN-LSTM
beach	76.56	65.29	79.86	81.26
bus	44.69	62.61	56.21	60.99
cafe/restaurant	47.79	61.89	57.72	57.12
car	75.49	71.11	85.51	80.57
city center	80.41	79.13	89.26	91.25
forest path	87.24	72.15	91.69	92.22
grocery store	77.19	57.39	83.07	84.71
home	66.28	72.71	52.70	55.39
library	64.07	71.27	69.29	72.55
metro station	85.71	85.76	82.52	82.47
office	83.40	78.93	82.97	89.09
park	38.24	36.11	48.89	43.88
residential area	61.87	51.71	52.54	57.74
train	22.46	38.87	24.42	38.21
tram	73.57	56.82	72.99	76.46
Overall acc	65.66	64.12	68.64	**70.92**

4.1.1. DNN

As a baseline system, we built a DNN which has three hidden layers with 512 hidden units each and used the ReLU activation in the hidden layers. The input features were 60-dimensional MFCC features including both delta and acceleration MFCC coefficients. Input layer was composed of a concatenation of 9 input frames (the current frame and the four previous and four next frames) resulting in 540 input units. To regularize the network, we used dropout with a probability of 40% for all hidden layers.

4.1.2. CNN

The CNN architecture for the baseline system comprised two convolutional layers, two pooling layers and one fully connected layer with softmax layer on the top. The input features were $F \times \tau$ size SIF, where F=40 and τ=40. In the first convolutional layer, the input SIF is convolved with 32 filters of fixed size 5×5. The first pooling layer then reduce the size of filtered SIF. We utilized max-pooling with kernel size 2×2 for all pooling layers. As an activation function, ReLU was applied. The second convolutional layer perform convolution between the output of the pooling layer and 16 filters of fixed size 5×5. After the second pooling is performed, the flattened output is combined with fully connected layer with 512 units. Dropout was only used after the second pooling layer and the fully connected layer with probabilities 30% and 40%, respectively.

4.1.3. LSTM

The network had two hidden layers with 256 LSTM units each and one feed-forward layer with 512 ReLU units. The structure of two LSTM layers is identical to the lower part in Figure 1. The input sequence consisted of 40 frames of 60-dimensional MFCC features. Dropout was applied with a probability of 40% for all layers. The output layer was identical to the mentioned in the previous section.

Table 2: Segment-based (30s) classification accuracy (%), averaged over 4-fold cross validation.

Scene	DNN	CNN	LSTM	CNN-LSTM
beach	84.62	73.08	88.46	88.46
bus	51.28	88.46	67.95	65.38
cafe/restaurant	58.97	73.08	67.95	60.26
car	78.21	73.08	88.46	89.74
city center	92.31	91.03	93.59	97.44
forest path	93.59	82.05	98.72	97.44
grocery store	83.33	71.79	85.90	91.03
home	80.77	89.74	64.10	70.51
library	75.64	83.33	76.92	76.92
metro station	94.87	100.0	92.31	94.87
office	93.59	96.15	87.18	96.15
park	41.03	43.59	57.69	52.56
residential area	87.18	75.64	73.08	74.36
train	25.64	46.15	29.49	43.59
tram	88.46	82.05	88.46	88.46
correct	881	912	905	**926**
Overall acc	75.30	77.95	77.35	**79.15**

4.1.4. Combination of LSTM and CNN

As a proposed system, we built a combined structure of LSTM and CNN in parallel. The network setup and structure of LSTM part and CNN part was identical to the aforementioned networks in Section 4.1.2 and 4.1.3, respectively. To combine and further train the two separated networks, we used fully connected layers. The connected layers were consisted of two hidden layers with 512 ReLU units each.

4.2. Results and discussion

We compared the average accuracies over all scenes for the conventional DNN, CNN, LSTM, and the proposed network. The frame-based classification results are given in Table 1. Table 2 shows the segment-based classification accuracy, where the *correct* represents the number of correctly classified segments among the total 1170 segments. The proposed method achieved higher accuracy than other networks in both frame-based and segment-based classification.

Though the combined neural network achieved higher performance on average, it did not give the best classification results across all scenes. In the *bus* case, CNN outperformed other networks. In the *park* case, LSTM had better result. In the *residential area* case, DNN achieved higher performance. This can be interpreted that the proposed network cannot fully train some acoustic scenes, and these scenes may not contain enough temporal information. Future research will deal with a more robust network architecture to extract distinct features of acoustic scenes.

Finally, the proposed method was found to improve classification performance and achieved an average accuracy of 79.15%. The baseline accuracy of audio scene classification task in DCASE 2016 challenge [14], which was based on MFCCs and GMMs, was 72.5%. Our method improved the performance by relative 6.7%.

5. CONCLUSION

In this paper, in order to enhance the classification accuracy of acoustic scenes, we proposed a novel neural network structure which achieved higher performance compared with the conventional DNN, CNN and LSTM architecture in terms of both frame-based and segment-based accuracy. In the segment-based classification results, the proposed technique obtained improvement of 3.85%, 1.2% and 1.8% in comparison with DNN, CNN and LSTM architecture, respectively. By combining different networks in parallel, the proposed method was able to learn complementary information of LSTM and CNN. Future works will study other neural network architectures in order to extract distinct features of acoustic scenes.

6. ACKNOWLEDGEMENT

This research was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (NRF-2015R1A2A1A15054343), and by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2016-H8501-16-1016) supervised by the IITP(Institute for Information & communications Technology Promotion).

7. REFERENCES

[1] L. Lu, H.-J. Zhang, and S. Z. Li, "Content-based audio classification and segmentation by using support vector machines," *Multimedia systems*, vol. 8, no. 6, pp. 482–492, 2003.

[2] A. Temko and C. Nadeu, "Classification of acoustic events using svm-based clustering schemes," *Pattern Recognition*, vol. 39, no. 4, pp. 682–694, 2006.

[3] D. Giannoulis, E. Benetos, D. Stowell, M. Rossignol, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: An ieee aasp challenge," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2013, pp. 1–4.

[4] 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.

[5] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, 2015.

[6] Z. Kons and O. Toledo-Ronen, "Audio event classification using deep neural networks," in *INTERSPEECH*, 2013, pp. 1482–1486.

[7] O. Gencoglu, T. Virtanen, and H. Huttunen, "Recognition of acoustic events using deep neural networks," in *European Signal Processing Conference (EUSIPCO)*, 2014, pp. 506–510.

[8] I. McLoughlin, H. Zhang, Z. Xie, Y. Song, and W. Xiao, "Robust sound event classification using deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,*, vol. 23, no. 3, pp. 540–552, 2015.

[9] A. Graves, "Supervised sequence labelling," in *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer, 2012, pp. 5–13.

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

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

[12] H. Zhang, I. McLoughlin, and Y. Song, "Robust sound event recognition using convolutional neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 559–563.

[13] M. Espi, M. Fujimoto, K. Kinoshita, and T. Nakatani, "Exploiting spectro-temporal locality in deep learning based acoustic event detection," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2015, no. 1, pp. 1–12, 2015.

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

[15] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," *arXiv preprint arXiv:1211.5063*, 2012.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] H. Phan, L. Hertel, M. Maass, and A. Mertins, "Robust audio event recognition with 1-max pooling convolutional neural networks," *arXiv preprint arXiv:1604.06338*, 2016.

[18] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2625–2634.

[19] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4580–4584.

[20] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

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