TELKOMNIKA Telecommunication, Computing, Electronics and Control
Vol. 18, No. 5, October 2020, pp. 2587~2596
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v18i5.14246

Sound event detection using deep neural networks

Suk-Hwan Jung, Yong-Joo Chung
Department of Electronics, Keimyung University, Korea

Article Info

Article history:

Received Oct 1, 2019
Revised Apr 23, 2020
Accepted May 8, 2020

Keywords:

Convolutional neural network
Convolutional recurrent neural network
Deep neural networks
Feedforward neural network
Recurent neural network
Sound event detection

ABSTRACT

We applied various architectures of deep neural networks for sound event detection and compared their performance using two different datasets. Feed forward neural network (FNN), convolutional neural network (CNN), recurrent neural network (RNN) and convolutional recurrent neural network (CRNN) were implemented using hyper-parameters optimized for each architecture and dataset. The results show that the performance of deep neural networks varied significantly depending on the learning rate, which can be optimized by conducting a series of experiments on the validation data over predetermined ranges. Among the implemented architectures, the CRNN performed best under all testing conditions, followed by CNN. Although RNN was effective in tracking the time-correlation information in audio signals, it exhibited inferior performance compared to the CNN and the CRNN. Accordingly, it is necessary to develop more optimization strategies for implementing RNN in sound event detection.

Corresponding Author:

Yong-Joo Chung,
Department of Electronics,
Keimyung University, Korea.
Email: yjjung@kmu.ac.kr

1. INTRODUCTION

Automatic sound event detection is a pattern recognition technique that automatically identifies various sound events occurring daily, such as glass breaking, baby crying, people screaming and car horning. In addition to identifying the label of sound events, it detects their onset and offset times. Automatic sound event detection has recently gained popularity owing to its numerous possible applications, including surveillance, urban sound analysis, information retrieval from multimedia content, health care, bird call detection, and autonomous vehicles [1-6].

To encourage research in the general area of sound signal classification including sound event detection, the "Detection and Classification of Acoustic Scenes and Events (DCASE)" challenge was held in 2013, 2016, 2017, 2018, and 2019 [7-11]. It includes two different categories: acoustic scene classification and sound event detection. In the former, the type of acoustic environment is determined using a long segment of audio signals, whereas, in the latter, specific sound events occurring in an acoustic scene are recognized. In this study, we only focus on sound event detection using two public databases from DCASE 2016 and [12].

Before the emergence of mehods based on deep neural networks, Gaussian mixture models (GMMs) were widely used in sound event detection. In fact, a GMM was used as a baseline recognizer in the DCASE 2016 challenge for Task 1 (acoustic scene classification) and Task 3 (monophonic sound event detection). The simple GMM-based bag-of-frames approach was adopted in the baseline system [13], where the mel-frequency cepstral coefficients (MFCCs), which have been widely employed in speech recognition, were used as acoustic features for the GMM. In addition to GMMs, traditional machine learning methods, such

as support vector machines (SVMs) [14] and non-negative matrix factorization (NMF) [15], were also widely used in sound event detection before their inferiority to recent deep learning-based methods was demonstated.

During the last decade, deep neural networks have achieved great success in image classification, speech recognition and machine translation [16-20]. Currently, deep neural networks exhibit state-of-the-art performance in all these domains. In sound event detection, FNNs have achieved better performance compared with GMMs and SVMs and it appears that they have replaced the traditional methods in the sound event detection. Owing to their simple architecture, FNNs have advantages over other deep neural networks. Specifically, fewer parameters and less computational time are required. Several frames of neighboring audio features (usually, log-mel filterbank (LMFB) energies) are concatenated in the time-domain so that they can be used as input to the network. Subsequently, they are multiplied by weight matrices and pass through nonlinear functions, and, hence they are forward propagated. However, the structure of an FNN cannot effectively compensate for the translational variances occurring in image signals owing to the fixed connections between the input and hidden units. Similar problems occur in sound event detection because the variations in the time-frequency domain of the audio signal may not be well modeled by the FNN. Another problem is that the temporal context is restricted to short-time windows of the input audio; therefore, it is difficult to model long-term correlations in the audio signals.

Compared with FNNs, CNNs can address the problem of time-frequency domain variations more efficiently. However, CNNs cannot effectively model long-term context correlations in the time-domain. Recurrent neural networks (RNNs) have been quite successful in modeling temporal context information in speech recognition. However, owing to their shortcomings in capturing the invariance in the time-frequency domain, RNNs are unable to outperform CNNs in sound event detection. Several approaches have been proposed for combining CNNs and RNNs to take advantage of both networks. Recently, convolutional recurrent neural networks (CRNNs), a combination of CNNs and RNNs in a single network, have been proposed for sound event detection, speech recognition and music classification [12, 21-24].

In this paper, we propose the use of a CRNN in polyphonic and scene-independent sound event detection and suggest optimal hyper-parameters and training strategies. Thus, the advantage of CRNNs over CNNs and RNNs is expected to be maximized. We evaluated the performance of the CRNN on recent datasets, including from the DCASE 2016 challenge. We also compared the performance of the CRNN with a CNN, an FNN and an RNN so that the advanatges of the CRNN may be better understood. The remainder of this paper is organized as follows; in section 2, we present feature extraction method and deep neural architectures used in this study. In section 3, we present and discuss the results of various experiments involving the FNN, CNN, RNN and CRNN. Finally, section 4 concludes the paper.

2. FEATURE EXTRACTION AND DEEP NEURAL ARCHITURES

2.1. Feature extraction

In this study, we use LMFB outputs as features for deep neural networks. We first compute the short-time Fourier transform (STFT) of 40-ms audio signals that were sampled at 44.1 kHz. The STFT is computed every 20 ms with 50% overlap. A total of 40 bands of mel-filterbank are extracted from the STFT with the range of 0~22,050 Hz and are log-transformed to obtain a 40-dimensional LMFB for each 20 ms time frame. After computing the LMFBs, we normalize them by subtracting the mean and dividing by the standard deviation computed from the training data.

2.2. FNN

The afoermentioned 40-dimentional LMFBs are used as features. Five successive time frames are concatenated to form 100-dimensional feature vectors as the input to the FNN. Each of the two 2 hidden layers has 1600 hidden units with ReLU activation. One output layer with sigmoid activation has K units where K is the number of sound event classes to be recognized. The outputs of the sigmoid activation are taken as the posterior probabilities for each of the classes, and the binarized outputs are compared with the ground truth table to determine the accuracy of the FNN.

2.3. CNN

The input to the CNN is $T \times 40$ LMFB features, and the overall structure of the network is shown in Figure 1. We use different structures for each of the two selected datasets. The structure in the figure is used for the TUT sound events 2016 dataset. The T frames of the 40-dimensional LMFBs are input to the convolutional layer with 256 feature maps, and each feature map has a two-dimensional 5×5 convolutional filter with ReLU activation. The output of the convolutional layer passes through a non-overlapping max pooling layer to reduce the dimensionality of the data. We compute the max pooling operation only in the frequency domain to retain the temporal information in the LMFBs. This is in contrast to

CNNs used in image classification, where the max pooing operation is performed in both dimensions. Unlike image signals, the time resolution information should be maintained in the audio signals to determine the onset and offset times in the sound event detection. There are three CNN layers, and the output has a dimension of $T \times 1 \times 256$, where the dimension of the frequency domain is reduced to 1, whereas the dimension of the time domain is unchanged, as mentioned previously. The output of the CNN layers is fed into a single feed-forward layer that has 256 units with ReLU activation. The final output layer with K(=number of classes) units of sigmoid activation follows the feed-forward layer and yields the sound event activity probabilities for each sound class at each time frame. Finally, the probabilities are binarized after thresholding over a constant value (0.5), and the activity of a class at a time frame is determined to be active or inactive depending on whether the binarized probability is 1 or 0.

2.4. RNN

The architecture of the RNN used in this study for TUT sound events 2016 is shown in Figure 2. $T \times 40$ LMFB features are used as the input of the GRU in the RNN architecture. We use three layers of GRUs with 256 units, followed by four feed-forward layers with 256 units. The output layer has K units with sigmoid activation. By using multiple feed-forward layers, the CNN and RNN have equally deep levels, thus allowing their performance comparison.

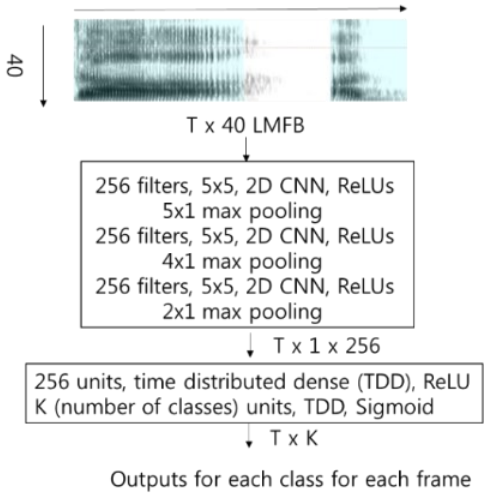

Figure 1. CNN architecture for TUT sound events 2016

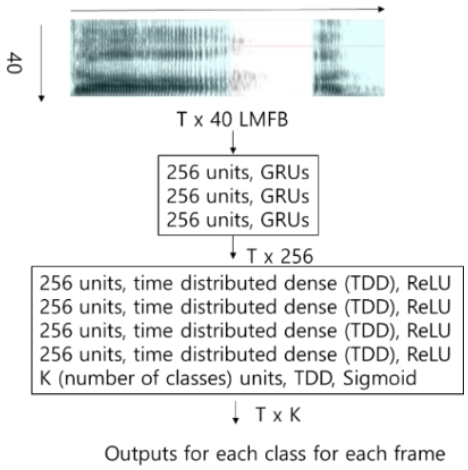

Figure 2. RNN architecture for TUT sound events 2016

2.5. CRNN

The architecture of the CRNN used in this study for TUT sound events 2016 is shown in Figure 3. It consists of convolutional layers in a cascade with recurrent layers followed by an output layer. The convolution layers act as a robust (time- and frequency-invariant) feature extractor. The recurrent layers provide contextual information in the time domain, which is highly important for recognizing sound events. Finally, the output layer generates the activity probabilities for the sound event classes for a given frame. The parameters of the convolutional, recurrent, and feedforward layers are optimized through backpropagation.

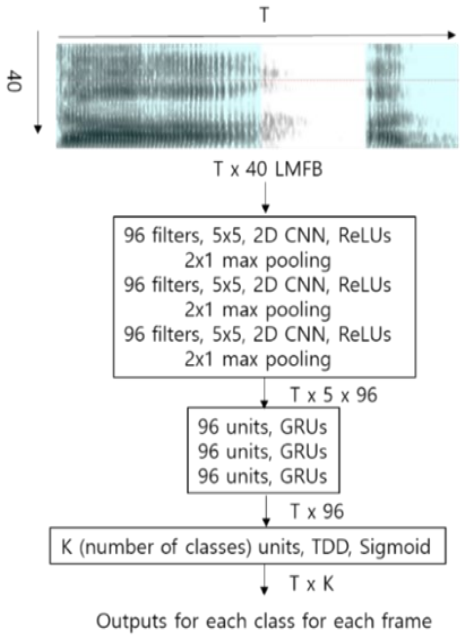

Figure 1. CRNN architecture for TUT sound events 2016

3. EXPERIMENTS

3.1. Databases

We evaluated the deep neural networks on two datasets. One was artificially generated, (TUT sound events synthetic 2016 abbreviated as TUT-SED synthetic), and the other (TUT sound events 2016) was recorded in real environments. The former was selected since the annotations in real audio data are rather subjective; therefore, the ground truth labeling may depend excessively on the annotators, particularly in the presence of polyphonic sound events.

TUT-SED Synthetic was generated by mixing isolated sound events from 16 different classes. A total 100 mixtures were created from 994 sound samples and divided into training, testing and validation data, with proportions 60%, 20%, and 20%, respectively. The total length of the mixture data was 566 min. Segments of length 3-15 s were selected from sound event instances to constitute a mixture, and there were no common sound event instances between training, testing, and validation data.

TUT sound events 2016 consists of recordings in two real environments: residential and home. Each recording was obtained from different locations to ensure large variability. Audio samples with the length of 3-5 min were recorded at each location, and the total length of the audio samples is 78 min. A total of 7 manually annotated classes correspond to the residential environment, whereas 11 annotated sound event classes correspond to the home environment. The four-fold cross-validation approach was adopted in the training and testing procedure to compensate for the small amount of data in this dataset. Twenty percent of the training data were allocated as validation data in the training phase. TUT sound events 2016 was used in the DCASE 2016 challenge, where the two environments were separately evaluated for scene-dependent classification. In this study, the classes from the two were not distinguished, resulting in 18(=7+11) sound event classes to be recognized for scene-independent classification. Therefore, only one classifier is required, rather than two, as was the case in the DCASE 2016 challenge.

3.2. Evaluation metrics

Evaluation methods use either segment- or event-based metrics [25]. In the fortmer, the evaluation of a deep neural network for sound event detection uses the error rate and F-score in a fixed time grid. The binarized outputs of the network are compared with the ground truth Table in 1 s segments. A sound event class is assumed to be detected correctly in a given segment if both the ground truth table and the binarized output corresponding to the class are active throughout the segment. A false positive implies that the ground truth table indicates that a sound event class is inactive, but the binarized output is active. In contrast, a false negative implies that the ground truth table indicates that the class is active, but the output is inactive. Finally, a true positive implies that both the ground truth table and the output indicate that a sound event class is active.

F-score is calculated as follows;

$$\mathrm{P} = \frac{TP}{TP+FP}, \quad R = \frac{TP}{TP+FN}, \quad F = \frac{2PR}{P+R}$$

where TP, FP and FN are counts of true positives, false positives, and false negatives, respectively. Further, P denotes precision, and R is recall.

Another metric is the error rate (ER), which is calculated in terms of insertions, deletions, and substitutions. A substitution error occurs when the binarized output detects a sound event class in a segment, but the label of the detected class is different from that of the ground truth table. A substitution error implies that a false positive and a false negative occur simultaneously in a segment. When only false positives occur in a segment, they are counted as insertions, and when only false negatives occur, they are counted as deletions. The ER is calculated as follows;

$$\mathrm{ER} = \frac{\sum_{k=1}^{K} S(k)+\sum_{k=1}^{K} D(k)+\sum_{k=1}^{K} I(k)}{\sum_{k=1}^{K} N(k)}$$

where, $\mathrm{N(k)}$ is the number of active ground truth events in a segment k and $\mathrm{S(k)}, \mathrm{D(k)}$ and $\mathrm{I(k)}$ denote the number of substitutions, deletions and insertions, respectively . K is the total number of segments.

In event-based metrics, a sound event is assumed to be correctly detected if the binarized output of the network has time-intervals overlapping with those of the correct label in the ground truth table. A 200 ms tolerance is allowed for onset time, and the same amount of time (200ms) or 50% of the duration of the correct label is allowed for the offset time. A false positive occurs when an active binarized output does not correspond to the correct label in the ground truth table within the allowed tolerance. If a sound event in the ground truth table does not correspond to the binarized output with the same label, a false negative occurs.

3.3. Results

We applied batch normalization after the convolutional layers and a dropout rate of 0.25 was applied to the convolutional and recurrent layers. We trained the networks using a binary cross-entropy loss function with the Adam optimizer. Early stopping was used to reduce overfitting. The training was stopped if the value of the loss function did not improve for more than 100 epochs. As the performance of deep neural networks varies with the learning rate, we attempted to select the optimal learning rate for all networks by testing their performance on the validation data. The performance of the CRNN on the TUT-SED Synthetic as the learning rate changes is shown in Table 1.

Table 1. Performance of CRNN on TUT-SED Synthetic as learning rate changes
(bold face numbers represent the best results)

	Validation data		Testing data		
Learning rate	Segment-based (F-score/ER)	Event-based (F-core/ER)	Segment-based (F-score/ER)	Event-based (F-score/ER)	Epoch
10^{-3}	61.69% / 0.52	37.69%/0.96	60.61% / 0.53	37.05%/0.97	16
10^{-4}	**68.75% / 0.45**	**43.49%/0.88**	**64.21% / 0.50**	**40.50%/0.96**	33
10^{-5}	66.44% / 0.49	39.10%/0.96	63.76% / 0.52	36.48%/1.04	157
10^{-6}	44.16% / 0.69	9.83%/1.24	43.38% / 0.71	10.82%/1.27	191

As shown in Table 1, the best performance is obtained when the learning rate is 10^{-4} for all conditions. The optimal learning rate for the validation data is also optimal for the testing data. Accoringly, the selection of the learning rate based on the validation data is quite reasonable. Similar performance variations with the learning rate could also be observed for the FNN, CNN, and RNN. The table shows that as the learning rate decreases, the number of epochs for which we obtain the best results increases. This is due to the slow

convergence of the weight parameters during training. When the learning rate is 10^{-4}, the epoch number is 33, whereas it is 191 when the learning rate is 10^{-7}. The slow convergence also results in poor performance, which is related to underfitting.

The variation of the loss function and accuracy at the output of the CRNN during training when the learning rate varies from 10^{-4} to 10^{-7} is shown in Figure 4. When the learning rate is 10^{-4}, the loss function on the validation data reaches its minimum at approximately 30 epochs (exactly 33); therafter, it fluctuates but never drops below the minimum. However, on the training data, the loss function continues to decrease throughout the duration of the training (we set the maximum number of epochs to 200). As overfitting should be avoided, we stop the iteration at 33 epochs using the aforementioned early stopping algorithm. Meanwhile, we can observe quite different characteristics when the learning rate is 10^{-5}. The loss function on the validation data decreases for a significantly longer period and reaches its minimum at 157. The longer iterations cause performance degradation on both the validation and testing data owing to underfitting. This phenomenon becomes more manifest as we further decrease the learning rate. When the learning rate is 10^{-7}, the loss function does not reach its minimum until the end of the training. A similar trend is observed when we monitor the accuracy instead of the loss function.

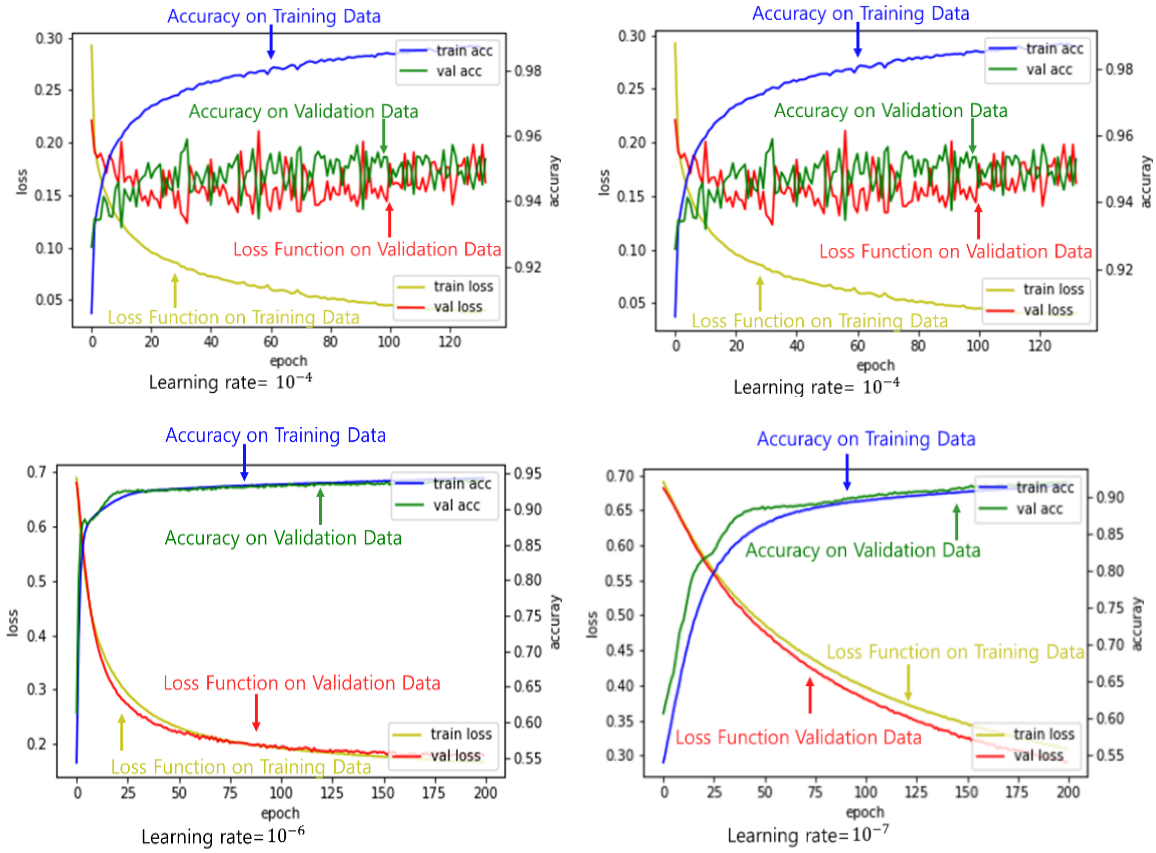

Figure 4. Variation of loss function and accuracy with learning rate
(CRNN on TUT-SED Synthetic dataset)

The performance comparison between the FNN, CNN, RNN, and CRNN on the TUT-SED Synthetic dataset is shown in Table 2. The learning rate is set to 10^{-4} in all networks according to the previous experiments. The evalutaion is represented in both the segment- and event-based methods. As presented in Table 2, the CRNN performs best under all testing conditions. Although the CNN has exhibted quite satisfactory performance in image classification, it is inferior to the CRNN in sound event detection because CNNs lack the ability to model the time correlation information (particularly long-term correlation) in audio signals. However, it can be observed that using the RNN alone could not result in improved performances compared with the CNN. This implies that the time-frequency invariant feature extraction by CNNs is highly important in sound event detection as is e case with invariant features in image classification.

Table 2. Performance comparison between FNN, CNN, RNN and CRNN on TUT-SED synthetic

	Segment-based		Event-based	
	F-score	ER	F-score	ER
FNN	54.57%	0.8	21.45%	3.51
CNN	60.38%	0.66	31.31%	1.87
RNN	47.28%	0.66	28.97%	1.32
CRNN	**64.21%**	**0.5**	**40.50%**	**0.96**

In addition to TUT-SED Synthetic, we investigated the performance of the networks using TUT sound events 2016. The performance of the CRNN on TUT sound events 2016 for varying learning rates is shown in Table 3. Contrary to the result on the TUT-SED Synthetic dataset in Table 1, we can conclude that the best performance is now obtained with a learning rate of 10^{-3} except when the event-based performance is measured on the testing data. This implies that the optimal learning rate varies depending on the training dataset. Moreover, the performance of the CRNN on the TUT sound events 2016 dataset was not as good as that on TUT-SED Synthetic. This may be due to the small amount of training data and the scene-independent classification in TUT sound events 2016. Although the number of weight parameters of the CRNN on TUT sound events 2016 was reduced by approximately 20% compared with that on TUT-SED Synthetic, the performance degradation could not be alleviated.

Table 3. Performance of CRNN on TUT sound events 2016 as learning rate varies (bold face numbers represent the best results)

	Validation data		Testing data		
Leaning rate	Segment-based (F-score/ER)	Event-based (F-score/ER)	Segment-based (F-score/ER)	Event-based (F-score/ER)	epoch
10^{-3}	**58.62% / 0.70**	**5.54%/4.99**	**37.18% / 0.90**	6.58%/3.08	25.5
10^{-4}	50.75% / 0.73	5.32%/4.11	36.41% / 0.88	**7.81%/3.06**	87.5
10^{-5}	1.44% / 0.99	0.00%/1.05	0.20% / 1.00	0.00%/1.00	187.8
10^{-6}	9.59% /10.34	1.13%/82.55	5.88% / 1.08	1.15%/1.85	199.8

The variation of the loss function and accuracy at the output of the CRNN during training on the TUT sound events 2016 dataset (as the learning rate varies from 10^{-3} to 10^{-6}) is shown in Figure 5. A similar trend to that in Figure 4 can be observed. However, with the same learning rate, the loss function on training data converges faster on the sound events 2016 dataset than on the TUT-SED Synthetic dataset. This implies that a smaller learning rate is desirable for the former to prevent overfitting. This is reflected in the performance scores in Table 3, where the best scores are obtained when the learning rate is 10^{-3} except for one case.

The performance comparison between FNN, CNN, RNN, and CRNN on the sound events 2016 dataset is shown in Table 4. The learning rate is set to 10^{-3} in all networks according to the results in Table 3. Table 4 demonstrate that the CRNN performs best in terms of the segment-based F-score and error rate. The CRNN is followed by CNN, and the RNN is the worst. This is in accordance with the results on the TUT-SED Synthetic dataset shown in Table 2. However, regarding the event-based metrics, unexpected results can be observed. Nevertheless, the low F-score and error rate in Table 4 imply that these results are not credible and and may therefore be ignored.

Table 4. Performance comparison between FNN, CNN, RNN and CRNN on the sound events 2016 dataset

	Segment-based		Event-based	
	F-score	ER	F-score	ER
FNN	25.11%	1.32	2.42%	9.81
CNN	35.28%	0.98	**7.54**%	4.49
RNN	24.34%	1.02	4.28%	**2.48**
CRNN	**37.18%**	**0.90**	6.58**%**	3.08

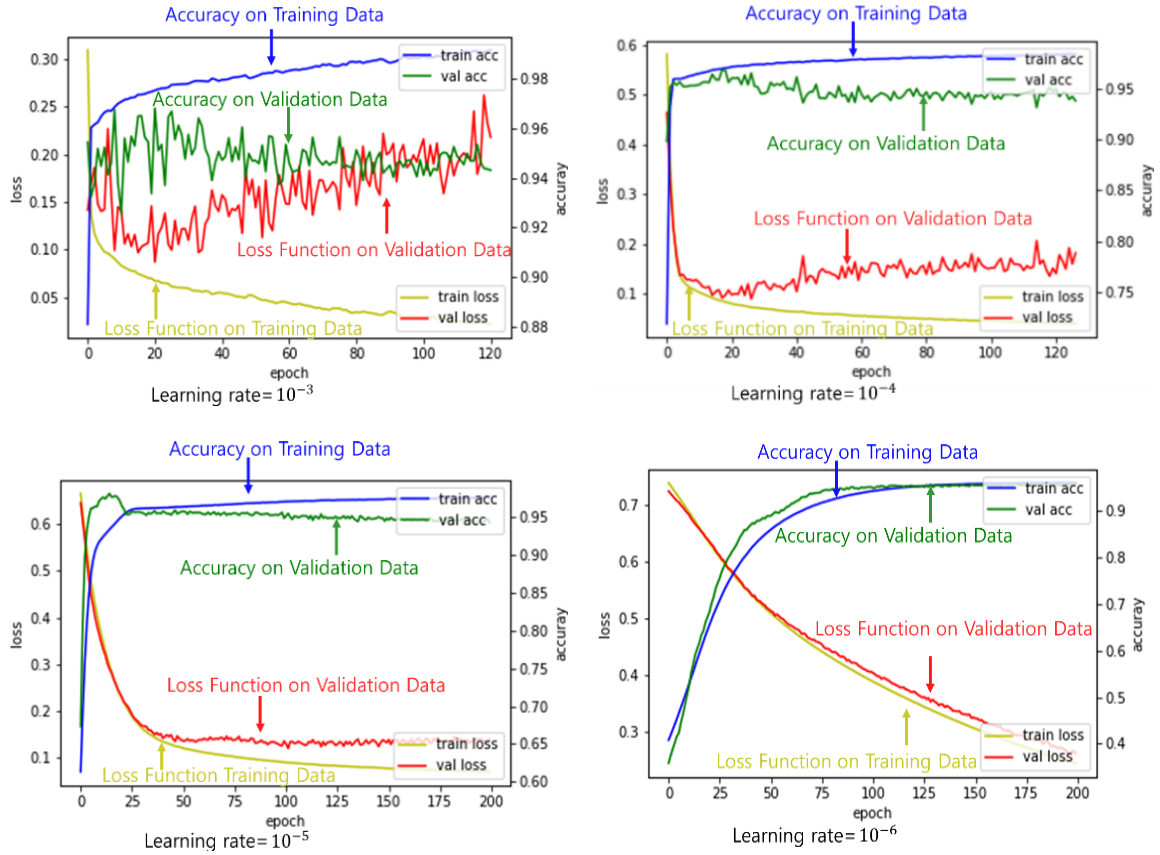

Figure 5. Variation of loss function and accuracy as learning rate changes
(CRNN on sound events 2016)

4. CONCLUSIONS

Deep neural networks have been widely used in various areas of pattern recognition. Recently, in sound event detection, numerous approaches based on deep neural networks have been proposed and have exhibted superior performance to among other conventional methods, such as GMMs and SVMs. In this paper, we proposed the application of representative deep neural networks to the sound event detection. We applied an FNN, a CNN, an RNN, and a CRNN to two independent datasets for sound event detection. The result demonstrated that the performances of these networks varied significantly depending on the learning rate. The optimal learning rate was selected based on the loss function on the validation data; this was confirmed to be quite reasonable based on extensive experimental results on the testing data. A small learning rate tends to underfit the networks to the training data, whereas an excessively large learning rate results in overfitting.

It was also demonstrated that the amount of training data and the type of classes considerably affected the performance of the networks. The performance on TUT-SED Synthetic was significantly better than that on sound events 2016, the size of which is approximately one seventh that of TUT-SED Synthetic, which contains audio classes that are difficult to distinguish. Although the number of weight parameters of the networks on sound events 2016 was reduced by 20% to compensate for the small amount of training data, the performance gap was quite large.

Finally, the CRNN outperformed the other networks, among which the CNN was the second most effective. The FNN and RNN performed worse than the CRNN and CNN. The poor performance of the RNN implies that time-frequency invariant features from the CNN are highly important in sound event detection. In future work, we will study a variant of the CRNN architecture that can use the characteristics of the RNN more effectively by considering different methods of coupling with the CNN. In addition, the segment length in the CRNN should be optimized to achieve the appropriate memory length in the GRUs.

ACKNOWLEDGMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by Ministry of Education (No. 2018R1A2B6009328).

REFERENCES

[1] M. K. Nandwana, A. Ziaei, and J. H. L. Hansen, "Robust Unsupervised Detection of Human Screams In Noisy Acoustic Environments*," Proceedings of the International Conference on Acoustics, Speech and Signal Processing*. *Brisbane,* pp. 161-165, 2015.

[2] M. Crocco, M. Christani, A. Trucco, and V. Murino, "Audio Surveillance: A Systematic Review," *ACM Computing Surveys,* vol. 48, no. 4, pp. 1-46, 2016.

[3] J. Salamon and J. P. Bello, "Feature Learning with Deep Scattering for Urban Sound Analysis," *23rd European Signal Processing Conference (EUSIPCO),* pp. 724-728, 2015.

[4] Y. Wang, L. Neves, and F. Metze, "Audio-based Multimedia Event Detection Using Deep Recurrent Neural Networks," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Shanghai,* pp. 2742-2746, 2016.

[5] F. Briggs, B. Lakshminarayanan, L. Neal, X. Z. Fern, R. Raich, S. J. K. Hadley, A. S. Hadley, and M. G. Betts, "Acoustic Classification of Multiple Simultaneous Bird Species: A Multi-instance Multi-Label Approach," *The Journal of the Acoustical Society of America,* vol. 131, no. 6, pp. 4640-4640, 2012.

[6] S. Ntalampiras, et al., "On Acoustic Surveillance of Hazardous Situations," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pp. 165-168, 2009.

[7] A. Mesaros, T. Heittola, and T. Virtanen, "TUT Database for Acoustic Scene Classification and Sound Event Detection*," 24th European Signal Processing Conference (EUSIPCO). Budapest,* pp. 1128-1132, 2016.

[8] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and Classification of Acoustic Scenes and Events," *IEEE Trans. On Multimedia,* vol. 17, no. 10, pp. 1733-1746, 2015.

[9] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "DCASE 2017 Challeneg Setup; Tasks, Datasets and Baseline System*," Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE 2017). Munich,* pp. 1123-1127, 2017.

[10] G. Dekkers, et al., "DCASE 2018 challenge-Task 5: Monitoring of Domestic Activities based on Multi-channel Acoustics," *KU Leuven, Tech. Rep.,* July 2018.

[11] N. Turpault, R. Serizel, A. Shah and J. Salamon, "Sound Event Detection in Domestic Environements with Weakly Labeled Data and Soundscape Synthesis," *Workshop on Detection and Classification of Acoustic Scenes and Events*, New York City, United States, 2019.

[12] E. Cakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional Recurrent Neural Networks for Polyphonic Sound Event Detection," *IEEE/ACM Trans. On Audio Speech and Language Processing,* vol. 26, no. 6, pp. 1291-1303, 2017.

[13] J. J. Aucouturier, B. Defreville, and F. Pachet, "The Bag-of-Frames Approach to Audio Pattern Recognition: A Sufficient Model for Urban Soundscapes but Not for Polyphonic music," *Journal of the Acoustical Society of America,* vol. 122, no. 2, pp. 881-891, 2007.

[14] C. C. Chang, C. J. Lin, "LIBSVM: A Library for Support Vector Machines," *ACM Transactions on Intelligent Systems and Technology,* vol. 2, no. 3, pp. 1-27, 2011.

[15] D. D. LEE, and H. S. Seung, "Learning the Parts of Objects by Non-negative Matrix Factorization," *Nature*, vol. 401, pp. 788-791, 1995.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems,* pp. 1097-1105, 2012.

[17] Graves, A. Mohamed, and G. E. Hinton, "Speech Recognition with Deep Recurrent Neural Networks*," Proceedings of the IEEE Int. Conf. on Acoustics Speech and Signal Processing (ICASSP),* pp. 6645-6649, 2013.

[18] K. Cho, B. Van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations Using Rnn Encoder-Decoder for Statistical Machine Translation*," Proceedings of the 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP),* pp. 1724-1734, 2014.

[19] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel and Y. Bengio, "End-to-end Attention-based Large Vocabulary Speech Recognition," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4945-4949, 2016.

[20] V. Mnih, N. Heess, A. Graves, et al., "Recurrent Models of Visual Attention," *Advances in Neural Information Processing Systems*, pp. 2204-2212, 2014.

[21] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, Long Short-term Memory, Fully Connected Deep Neural Networks*," Proceedings of the 2015 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Brisbane,* pp. 4580-4584, 2015.

[22] K. Choi, G. Fazekas, M. Sandler, K. Cho, "Convolutional Recurrent Neural Networks for Music Classification*," Proceedings of the 2017 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* pp. 2392-2396, 2017.

[23] S. H. Jung, Y. J. Chung, "Audio Event Detection Using CNN and CRNN*," Proceedings of the 7th International Conference on Next Generation Computer and Information Technology, Hokkaido,* pp. 134-137, 2018.

[24] R. Harb, F. Pernkopf, "Sound Event Detection Using Weakly Labeled Semi-supervised Data with GCRNNs, VAT and Self-adaptive Label Refinement," *Workshop on Detection and Classification of Acoustic Scenes and Events*, Surrey, UK, Oct. 2018.

[25] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for Polyphonic Sound Event Detection," *Applied Sciences*, vol. 6, no. 6, pp. 162-178, 2016.

BIOGRAPHIES OF AUTHORS

Sukwhan Jung received his B.Sc. and M.Sc. Degree in Electronics Engineering from Keimyung University, Daegu, South Korea in 2016 and 2018, respectively. He has been with Samju Electroincs Co. since March 2018. His main research interests are audio event detection under noisy environments and deep learning for artificial intelligence.

Yongjoo Chung received his B.Sc. degree in Electronics Engineering from Seoul National University, Seoul, South Korea in 1988. He earned his M.Sc. and PhD degree in Electrical and Electronics Engineering from Korea Advanced Science and Technology, Daejon, South Korean in 1995. He is currently a Professor with the Department of Electronics Engineering at Keimyung University, Daegu, S. Korea. His research interests are in the areas of speech recognition, audio event detection, machine learning and pattern recognition.