Sound Event Detection Based on Convolutional Neural Networks with Overlapping Pooling Structure

Hang Zhu* and Hongjie Wanb,*
Information Engineering Dept, Beijing University of Chemical Technology, Beijing China

azzhuhang@163.com; b,*Wanhj@mail.buct.edu.cn

Abstract. In this paper, a sound event detection measure is proposed. This measure is based on convolutional neural networks with overlapping pooling structure. Different from the traditional GMM-HMM model and DNN-HMM model, the CNN model uses the convolutional layer which can speed up training by reducing training parameters. In this paper, the extracted sound feature is the mel-frequency cepstrum coefficient (MFCC). The dropout layer is added to the convolutional layer. Over-fitting can decrease the accuracy of the detection, dropout layer can prevent the model from over-fitting. Moreover, the overlapping pooling structure is used in CNN, the stride size is smaller than the pooling kernel size. The output of pooling layer has overlapping parameters, which can increase the richness of features. The final experimental results show that the precision of the proposed CNN model more robust than the GMM-HMM model and baseline model.

1. Introduction
Sound event detection is widely used in healthcare monitoring [1], multimedia indexing and retrieval [2], urban analysis, etc. The traditional sound event detection model, such as Gaussian-Hidden-Markov-Model [3-4], is an effective model, it can achieve sound event detection under the partial conditions. Features include the mel-frequency cepstrum coefficient (MFCC) and the zero-crossing rate (ZCR). Deep learning technology has been applied in many parts of sound detection, besides the well-known back-propagation (BP) algorithm is applied to the neural network. The BP algorithm can optimize the model parameters and promote the accuracy rate of the deep neural network (DNN). This article adopts the CNN method to detect sound event signals and improve the robustness of detection system.

2. Framework of sound event detection system
The network model needs to be trained to optimize a model. The multi-sound event detection system is generally divided into three steps: data pre-handling, model building and output evaluating.

2.1. Data pre-handling
According to the requirements of this job, firstly, the source signal is divided into short time frames. The time-frequency transformation performed on the short frame can remove the redundant information, so that the sound feature vector of the corresponding time frame can be obtained.

2.2. Model building
By comparing the accuracy between the output value and the tag value, a reliable network model is obtained. The specific network model structure will be described in detail in the third part. Figure 1 shows the block diagram system:

![Figure 1. Framework of sound event system](image)

2.3. Model evaluation
Supervised learning can learn a limited number of samples to minimize the error probability of classifiers. By comparing the detection probability of each class, model can judge the category of the event and judge the quality of the network model. The calculation formula of accuracy is as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

In the formula, true-positive (TP) predicts positive classes as positive classes. True-negative (TN) predicts negative classes as negative classes. False-positive (FP): false positive prediction of negative class number as positive class number. False-negative (FN): predicts positive class as negative class number.

3. Building the deep neural network model

3.1. Baseline system
In the last decades, DNN plays an important role in solving sound problems such as natural language processing, speech enhancement and speech recognition. Baseline system is an advanced detection system based on DNN. However, the DNN system has many parameters, which will take a lot of time to train, if the features are not suitable for the task, the final result will be poor.

Baseline system is suitable for being used in some simple sound event detection [5]. The baseline network consists of one input layer, two hidden layers (128 neurons per layer) and one output layer (17 classes). The baseline system is easy to build compared with other neural networks, and it is suitable for simple situations. Baseline system has good robustness in the case of fewer types of events.

3.2. Proposed Convolutional Neural Network
The convolutional neural network consists of 3 convolutional layers, 3 pooling layers, a flatten layer and two full connection layers. Figure 2 shows the convolutional neural network model.
The general convolutional neural network includes the following five parts:

1) Convolutional layer: Some low-level features may be extracted in the first convolutional layer, such as edges, lines and angles, while the more higher features can be extracted from the lower features by deeper networks. In this task, it is to extract the information hidden in MFCC. In order to reduce the over-fitting problem, dropout = 0.2 was set. In the convolutional layer, there are 64 neurons, and the convolutional kernel size is 3*3. The activation functions of the full connection layer and the convolutional layer is Elu activation function. The Elu function formula is shown below:

$$ELU(x) = \begin{cases} 
  x, & x > 0 \\
  \alpha(exp(x) - 1), & x \leq 0 
\end{cases}$$

(2)

2) Pooling layer: The size of the parameter matrix can be reduced by Pooling layer efficiently. Pooling layer can speed up the calculation and reduce over-fitting [6-8]. The step size is smaller than that of pooling kernel, this structure can improve the richness of features. In the pooling layer, the step size is 1*1. The pooling size is 2*2.

3) Flatten layer: Flatten layer is added after the convolutional layer. The flatten layer can reduce the dimension of parameters. After dimension reduction, the parameters can be input into the full connected layer for detection and classification. Moreover, flatten layer does not affect the size of the batch.

The flatten layer is shown in the Figure 3.

4) Fully connected layer: Neurons are completely connected and act as a "classifier" in the whole convolutional neural network. The output of flatten layer is input to fully connected neural network. The input of the full connected layer is a two-dimensional parameter, which contains the learned features and tags.

5) Output layer: For each test sample, the output is a one-dimensional array of $N$ elements.

$$M = \max (x_1, x_2, x_3, x_4, \ldots, x_{n-1}, x_N)$$

(3)

$M$ is the sound event. $N$ is the number of sound event categories. Through the maximum posterior probability, the test sample can be detected. $x_0$ is the sound event. $\theta$ is the label of the sound event.

$$P(\theta|x_0) = \frac{P(x_0|\theta)P(\theta)}{P(x_0)}$$

(4)
Finally, accuracy rate can judge the advantages and disadvantages of the neural network model.

4. Experiment

4.1. Data preparation
Daily-Sound-Event is an open-source sound signal database [9]. The sound event signal collected by quaternion microphone validation array. The sample sampling rate was 44.1 KHz, the 20-30ms sound can be regarded as short-time stable, so we set the frame length to 1024 sampling points (23ms).

\[ k = \frac{M}{1000} \times N, \quad 20 \leq N \leq 30 \]  \hspace{1cm} (5)

In the above equation, M is the sampling frequency, N is the signal stationary time, where M=44100, N= 23.

The whole data set contains 2040 data samples, which are divided into training set, validation set and test set. The proportion is classified into three parts, the 72% is the training set, the 8% is the validation set and the 20% is the test set.

The spectrogram of the flushing sample is shown in the Figure 4.

![Spectrogram of the flushing sound event sample](image)

Figure 4. The spectrogram of the flushing sound event sample

The spectrogram of the noise sample is shown in the Figure 5.

![Spectrogram of the noise sample](image)

Figure 5. The spectrogram of the noise sample

From Figure 4 and Figure 5, left figure is the waveform in the frequency domain and right figure is the waveform in the time domain. In the time domain, the noise has a continual disturbance, while the sound signal has a changeable amplitude.

4.2. Get Feature vectors
Extraction steps are mainly divided into two steps: one is to pre-process the sound events; The other is to obtain MFCC features. In this experiment, the operation of smoothing was carried out. For the sample with short signal time, the zero-filling operation is carried out, and the sample with long signal time is clipped. In this paper, the MFCC dimension of each frame is equal to 13.

4.3. Training and Evaluation
During the experiment processing, the MFCC is set as feature to learn. The loss function is CROSS-ENTROPY. This experiment use the Adam gradient-based optimizer [10]. The calculation formula of cross entropy is:

\[ H_{y'}(y) = - \sum_{i} y'_i \log (y_i) \]  \hspace{1cm} (6)
In the experiment process, the hyperparameters are designed as below: the batch-size, epoch and the learning rate is designed as 8, 500 and 1e-3 separately. Meanwhile, in order to avoid over-fitting in the neural network system, the rate of a dropout process is set to 0.2.

5. Detection Results

This experiment has 1468 training samples and 163 verifiable samples. Epochs are set to 500. The network training result is shown in Table 1:

| Epoch   | loss  | acc  | validation_loss | validation_acc |
|---------|-------|------|----------------|----------------|
| 100     | 1.8148| 0.4099| 1.9467         | 0.3824         |
| 200     | 0.8432| 0.7353| 1.2162         | 0.7353         |
| 400     | 0.1213| 0.9945| 1.1946         | 0.8824         |
| 500     | 0.0113| 0.9963| 1.0797         | 0.9265         |

Table 1 shows the training loss is declining. The validation_loss is also declining. Validation_loss has been declining and remained stable, indicating that this system can still fit data. Neural network is not entered into the fitting. And the detection rate of the final validation set remains around 90%.

6. Conclusion

In this passage, the baseline system using full connected neural network, the other is a convolution neural network with overlapping pool structure. The experimental results suggest that the CNN model is more reliable with the same database, and has a better detection rate than the baseline system. The CNN with overlapping pooling structure is used to enhance the richness of features, and the final recognition rate is improved by 17% compared with the baseline system. The experiment also indicates that the dropout layer can improve the precision of the sound event detection.
References

[1] S. Goetze, J. Schröder, S. Gerlach, D. Hollosi, J. Appell, and F. Wallhoff, “Acoustic monitoring and localization for social care,” J. Comput. Sci. Eng., vol. 6, no. 1, pp. 40–50, 2012.

[2] J. Salamon and J. P. Bello, “Feature learning with deep scattering for urban sound analysis,” in Proc. 23rd Eur. Signal Process. Conf., 2015, pp. 724–728.

[3] He, X., Deng, L.: Discriminative Learning for Speech Recognition: Theory and Practice. Morgan and Claypool (2008)

[4] L. Cao, Y. Shen, T. Shan, Y. Xia, J. Wang and Z. Lin, "Bearing Fault Diagnosis Method Based on GMM and Coupled Hidden Markov Model. 2018 Prognostics and System Health

[5] C. Shan et al., "Investigating End-to-end Speech Recognition for Mandarin-English Code-switching," ICASSP 2019 - 2019 IEEE International Conference on Acoustic ICASSP.

[6] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[7] T. Wang, D. Wu, A. Coates, and A. Ng, “End-to-end text recognition with convolutional neural networks,” in International Conference on Pattern Recognition (ICPR), 2012, pp. 3304–3308.

[8] B. B. Le Cun, “Handwritten digit recognition with a back-propagation network,” in Advances in neural information processing systems. Citeseer, 1990.

[9] S. E. D. Ttxphuocapcs. (2017) Daily -Sound-Event (DSE) database. [Online]. Available: https://github.com/txphuocapcs/CS487_SoundEventDetection

[10] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” CoRR, 2014