Sound Event Detection for Real Life Audio DCASE Challenge

Dai Wei, Juncheng Li, Phuong Pham, Samarjit Das, Shuhui Qu, Florian Metze [*]

Robert Bosch Research and Technology Center[1],
Carnegie Mellon University[4],
Stanford University[3],
University of Pittsburgh[2]

ABSTRACT

We explore logistic regression classifier (LogReg) and deep neural network (DNN) on the DCASE 2016 Challenge for task 3, i.e., sound event detection in real life audio. Our models use the Mel Frequency Cepstral Coefficients (MFCCs) and their deltas and accelerations as detection features. The error rate metric favors the simple logistic regression model with high activation threshold on both segment- and event-based contexts. On the other hand, DNN model outperforms the baseline in frame-based context.

Index Terms— Sound event detection, DNN, neural network, deep learning

1. INTRODUCTION

Polyphonic sound event detection (SED) is the task of detecting overlapped sound events from audio stream. Deep neural networks have shown promising results for polyphonic sound event detections [3][2][8]. In this project, we explore DNN to detect polyphonic sound events for the DCASE 2016 challenge (task 3)[7]. Task 3 includes two scenarios, i.e., home (with 11 event classes) and residential area (with 7 event classes).

2. SYSTEM OVERVIEW

Our models use a two-phase frame-based pipeline to detect polyphonic sound events (Figure 1). Phase 1 extracts audio frames to train the models. Our models will output which event is onset (active) in a frame. Phase 2 groups adjacent events into segments and does further post-processing steps.

2.1. Audio Features

For each audio clip, we apply short time Fourier transform (STFT) with 40ms window size, 50% overlap and Hamming window. Mel filterbank with 40 Mel bands is used to extract Mel band energies. Later, 20 Mel Frequency Cepstral Coefficients (MFCCs) features are extracted from each time frame. We also include the delta and acceleration features from 20 MFCCs. In total, 60 features were extracted from each time frame.
In order to make use of the temporal information, we concatenate the feature vector f_t at time t with its previous and preceding time frames. The concatenated feature vector $x_t = [f_{t-1}\ f_t\ f_{t+1}]$ is considered as a data instance for our models.

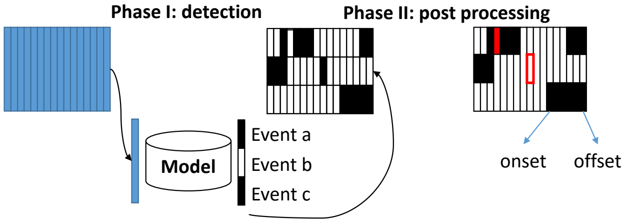

Figure 1: the two-phase frame-based pipeline.

2.2. Deep Neural Network

The DNN is composed of an input layer, multiple hidden layers and an output layer. In each hidden layer, we add batch normalization [5], dropout [9], L2 regularization and the rectified linear unit (ReLU) [9]. Our preliminary results show that without these best-of-practice techniques, and given the size of the challenge's dataset, the DNN quickly saturates, i.e., outputs all 0s.
The output of the network is a binary feature vector y_t where the i^{th} element of y_t is 1 if the i^{th} event is onset (active) at time frame t; otherwise, the i^{th} element of y_t is 0.
The DNN is trained by minimizing the multiclass logloss (categorical cross-entropy). We use Adam, a first-order gradient-based optimization of stochastic objective functions. In our preliminary experiments, Adam showed better performance than RMSProp and Nesterov momentum optimizer. The DNN was built using Keras framework [4] with Theano backend [2].

2.3. Post-processing

Event segments are created by merging adjacent frames together. Furthermore, we also discard any segments that are shorter than 100ms as well as merge segments that are less than 100ms away from each other. The post-processing techniques were suggested by the challenge's baseline system.

[*] Thanks to ABC agency for funding.

		GMM	LogReg	DNN	DNN (high)
Home					
Frame-based	F1	12.2%	2.6%	**14.4%**	9.7%
Segment-based	ER	1.05	**1.02**	2.47	1.51
	F1	11.3%	3.6%	**20.3%**	13.0%
Event-based	ER	1.30	**1.21**	5.97	2.64
	F1	2.2%	0.2%	**3.6%**	3.5%
Residential area					
Frame-based	F1	20.5%	5.8%	**23.3%**	19.2%
Segment-based	ER	1.04	**0.97**	1.48	0.98
	F1	20.2%	7.1%	**28.1%**	23.1%
Event-based	ER	1.99	**1.35**	6.7	2.88
	F1	0.7%	0.2%	**1.4%**	0.4%

Table 1: Class-wised performance metrics of Home and Residential area scenes.

3. EXPERIMENTAL RESULTS

We evaluate the DNN model on DCASE 2016 challenge task 3's development dataset. The challenge uses both overall and class-wised metrics. However, we only focus on class-wised metrics because overall metrics would be biased to the majority classes.

The main evaluation metric of the challenge is error rate (ER), which is calculated as

$$ER_{segment-based} = \frac{\max(N_{ref}, N_{sys}) - TP}{N_{ref}}$$

$$ER_{event-based} = \frac{FN + FP}{N_{ref}}$$

where TP is the true positive, FN is the false negative, FP is the false positive and N_{ref} is the number of ground truth, N_{sys} is the number of model's output, $ER_{segment-based}$ is the aggregated 1s segment error rate, and $ER_{event-based}$ is the event onset, offset error rate. However, both segment- and event-based metrics include post-processing steps. In order to evaluate the models' performance without any post-processing effects, we also use a frame-based metric (F1 only) in this report.

The baseline Gaussian Mxiture Model (GMM) from the DCASE challenge is used for evaluation. Although we have tried with multiple configurations of DNNs, i.e., number of hidden layers and number of hidden units, only the best DNN structure is reported here. The multi-class DNN has 3 hidden layers and 700 hidden units per layer, dropout rate is 0.5 and L2 regularization coefficient is 10^{-3}. All *silent* frames (no annotated events) are discarded. This helps the model focusing more on events' characteristics rather than be distracted by the silent frames.

We have an assumption that the ER metrics focus more on incorrect detections than correct detections. In other words, the metrics prefer high precision to high recall. For example, if a model never gives any positive detection, the $ER_{segment-based}$ is 1. On the other hand, $ER_{event-based}$ does not directly depend on TP or TN. Therefore, we evaluate a simple Logistic Regression classifier (LogReg) with a high activation threshold, i.e. an event is only detected in a frame when the positive probability is higher than 0.9. Besides LogReg, we also evaluate the DNN (DNN-high) with high threshold, i.e. 0.95, in this project.

We do a 4-fold cross validation with the same split as the challenge's baseline model. Table 1 shows class-wised performance of experimental models in frame-based, segment-based, and event-based metrics.

In Table 1, LogReg with the high activation threshold trick can outperform the GMM baseline in all ER metrics. However, LogReg achieves low F1 scores because the models predicted almost nothing except the frame that they feel most confident. As the result, most of the detections belong to the majority classes. We think this type of LogReg would not benefit a real use case because it only focuses on a few number of classes. On the other hand, DNNs get higher F1 than the GMM in all cases. However, the ERs are not good because DNNs made many FPs and FNs. As expected, the DNNs (high) have better ERs than DNNs but worse F1s because the DNNs (high) are more sensitive. However, the DNNs (high) cannot outperform the GMM in ERs. One possible reason is the dataset is quite insufficient. Thinking of LogReg as a Multi-class DNNs without hidden layers, adding more structure to LogReg would hurt its performance given a modest dataset.

4. CONCLUSIONS

We explore multi-class DNNs on the dataset described in [7]. The experimental multi-class DNNs achieve better than the GMM baseline in terms of F1 scores but not in ERs because they still have many FNs and FPs. On the other hand, the LogReg with high activation threshold in favor of ERs can achieve the best ER among all experimental models. We believe that with better structure and sufficient data, the multi-class DNNs can work well on this problem.

5. ACKNOWLEDGMENT

We would like to thanks to ABC for funding

6. REFERENCES

[1] Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I., Bergeron, A., Bouchard, N., Warde-Farley, D., and Bengio, Y. (2012). Theano: new features and speed improvements. arXiv preprint arXiv:1211.5590.

[2] Cakir, E., Heittola, T., Huttunen, H. & Virtanen, T (2015). Multi-label vs. combined single-label sound event detection with deep neural networks. In 23rd European Signal Processing Conference 2015 (EUSIPCO 2015).

[3] Cakir, E., Heittola, T., Huttunen, H. & Virtanen, T (2015). Polyphonic sound event detection using multi label deep neural networks. In International Joint Conference on Neural Networks 2015 (IJCNN 2015)

[4] Chollet, F. (2015). keras. https://github.com/fchollet/keras.

[5] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.

[6] Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[7] Mesaros, A. (2016). 2016 dcase challenge. http://www.cs.tut.fi/sgn/arg/dcase2016/.

[8] O. Gencoglu, T. Virtanen, and H. Huttunen, “Recognition of acoustic events using deep neural networks,” in Proc. 22nd European Signal Processing Conference (EUSIPCO), 2014, pp. 506–510.

[9] Dahl, G.E., Sainath, T.N. and Hinton, G.E., 2013, May. Improving deep neural networks for LVCSR using rectified linear units and dropout. In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on (pp. 8609-8613). IEEE.