A survey of Deep Learning for Polyphonic Sound event detection

An Dang, Toan H. Vu, Jia-Ching Wang
Department of Computer Science and Information Engineering, National Central University
andtt.cit@gmail.com, toanvuhong@gmail.com, jcw@csie.ncu.edu.tw

***Abstract*— Deep learning has achieved state of the art in various machine learning problems, such as computer vision, speech recognition, and natural language processing. Sound event detection (SED), which is about recognizing audio events in real-life environments, has attracted a lot of attention recently. Many works have been successful when applying deep learning techniques for the SED problem as can be seen in Detection and Classification of Acoustic Scenes and Events (DCASE) challenge 2016-2017. In this paper, we present a review of the SED problem and discuss different deep learning approaches for the problem.**

Keywords-deep learning; neural networks; convolutional neural networks; recurent neural networks; sound event detection

I. Introduction

The development of deep learning - which draws inspiration from the structure and function of the brain - has dramatically improved the state-of-the-art in complex problems such as speech recognition [1, 2], image recognition [3], and natural language processing [4]. Deep learning refers to artificial neural networks that are composed of many layers learning representations of data with multiple levels of abstraction. Deep learning demonstrated efficiency in learning complex structure in large data set by using the back propagation approach to indicate how to change internal parameters of inputs signals to produce an expected output signal.

The tasks of sound event detection (SED) involves locating and classifying sound events in an audio from real life environments - such as baby crying, people walking, and dog barking. In other words, the goal of SED is to estimate start time and end time of each event and provide a textual descriptor for each event within an audio recording. SED has two main tasks including monophonic sound event detection and polyphonic sound event detection. The monophonic sound event detection is required to detect the most prominent sound events at each time, while the polyphonic sound event detection identifies overlapped sound events along with the single sound events in a scene. Compared to the monophonic SED, the polyphonic SED presents much more challenge due to the recordings of polyphonic SED exist a large set of overlapping sound events in the same time. It is a challenge to detect all the events happening at a time. Fig.1 describes the task of polyphonic SED, which is mainly topic presenting in this paper.

In the past decade, different approaches have been proposed for SED problem. For example, several SED systems have addressed polyphonic detection using Gaussian mixture models with hidden Markov models (GMM-HMM) [5, 6] and a couple non-negative matrix factorization [7]. More recently, numerous deep learning approaches have been proposed for SED problem and considered the cutting-edge method for SED problem. In [8], a deep neural network architecture was used (DNN) and improved over previous approaches by a large margin in term of accuracy. However, this architecture - artificial neural networks with many intermediate hidden layers, is not inherently well suited to represent time series input such as audio, video or text due to temporal information is limited to short time windows. This problem has been addressed by using two more powerful neural networks are convolutional neural networks (CNNs) and recurrent neural networks (RNNs). They have been applied for SED and presented excellent performance in SED task [8, 9, 10, 11, 12, 13, 14, 15, 16]. Moreover, a combination of CNN and RNN in a model [17, 18] shown to outperform previous SED approaches.

In this paper, we present an overview of the use of deep learning for SED problem. Several SED surveys have been published such as the work by Stowell et al. [19]. However, they just summarized of the use of HMM and NMF for detecting sound events. They lack of presenting the use of deep learning approaches for SED problem. In this work, we provide a review of approaches that use deep learning for SED and discuss their performance.

The remainder of the paper is organized as follows. In section II, we review background of the most of existing approaches of deep learning using for SED problem. A summary about deep learning approaches is described in section III. Finally, conclusions are given in section IV.

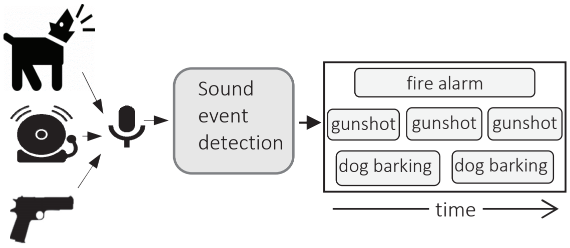

Figure 1: An example of polyphonic sound event detection task.

II. Deep learning

In this section, we review of background about deep learning including neural networks (NNs), convolutional neural networks (CNNs) and recurrent neural networks (RNNs).

A. Neural Networks

Neural networks (NNs), also known as artificial neural networks, are inspired by mimicking the way the human brain operates. The fundamental structure of a NN is organized in layers. Layers are constituted of a number of interconnected neurons which contain activation functions. The connections between the neurons are weighted. Each neuron basically consists of inputs which are multiplied by connection weights and then computed by a mathematical function which determines the activation of the neuron. For a traditional neural network, neurons within a layer are

978-1-5386-3276-5$31.00 ©2017 IEEE

no share connections, but neurons between two adjacent layers are fully pairwise connected. Such layers are called fully-connected layers. Most applications require networks that contain at least the three normal types of layers - input, hidden, and output. The input layer is composed of D neurons, where D is the dimensionality of the input data. Hidden layers are between the input and output layer, and perform intermediate computations of the network. A NN is called a deep neural network (DNN) when it has several stacked hidden layers, where the depth increases with the number of hidden layers. To train NNs, a backpropagation algorithm is used that optimize the performance of the network by adjusting the weights.

Although NNs have been adopted in many applications, it shows some shortcomings when applying for spatial and temporal structure data such as images, audio, speech, and text. Firstly, the design of NNs are fully-connected layers that have a large number of parameters and the number of parameters rapidly increases during training process; it leads to slow learning for spatial and temporal structure data. Secondly, every pair of neurons between two layers of NNs has their own parameters, preventing the network exploits features correlations in high-dimensional spatial and temporal context. CNNs and RNNs can address the limitation by sharing parameters between neurons.

B. *Convolutional Neural Networks*

The convolutional neural networks (CNNs) (LeCun, 1989) [20] are type of neural networks architecture, were developed to overcome drawbacks of NNs when dealing with spatial structure data. CNNs were inspired based on the processing visual cortex of humans. A CNN consists of three basic components – convolutional layers, pooling layers, and fully-connected layers.

A convolutional layer firstly performs convolution operations to produce a set of linear activation, then each linear is fed into a non-linear activation function such as the ReLU or *tanh*. Convolutional networks have a local connectivity to exploit the spatially-local correlation between neurons of adjacent layers. The size of this connectivity is controlled by a hyper-parameter called receptive field. The receptive field is small tensor, with the dimensions $[w \times h \times depth]$, which is applied across the entire previous layer using the same parameters. The parameter sharing is used in convolutional layers to reduce the number of parameters in the whole network and make the computation more efficient.

Pooling layers are usually used after each convolutional layer to reduce representation size of convolutional output and computational burden on the next layers. A pooling function partitions its input into a set of rectangles and each sub-region yields a summary statistic value of the nearby inputs. The use of pooling is very useful to extract the most effective information from an area. There are different kinds of pooling functions such as max pooling, average pooling, weighted average pooling, and L2 norm pooling, in which max pooling the most of popular function is used in pooling layers.

After several convolutional layers and pooling layers, fully connected layers are adopted at the end of a CNN. A fully-connected layer in CNNs is similar the layer in a standard neural network where neurons adjacent layers are fully pairwise connected and neurons in the same layer share no connection.

C. *Recurrent Neural Networks*

Recurrent neural networks (RNNs) [21] are family of NNs for processing sequential or temporal data. While the advantages of CNNs are that can effectively process to spatial structure data with large width and height, the powerful point of RNNs are that can scale to much longer sequences. In recurrent networks, a hidden layer with self-connections acts as memory that accumulates information over time from an input sequence. Most RNNs can handle sequences of variable length.

Given a sequence of input vectors $x = (x_1, x_2, ..., x_\tau)$, a standard RNN calculates a sequence of hidden activations $h = (h_1, h_2, ..., h_\tau)$ and target vectors $\hat{y} = (\hat{y}_1, \hat{y}_2, ..., \hat{y}_\tau)$ as

$$h_t = f(W_{xh} h_t + W_{hh} h_{t-1} + b_h) \qquad (1)$$

$$\hat{y}_t = g(W_{h\hat{y}} h_t + b_{\hat{y}}) , \qquad (2)$$

for all time steps $t = 1,2,...,\tau$, where the matrices $W_{xh}, W_{hh}, W_{h\hat{y}}$ denote weights connecting layers – input layer and hidden layer, hidden layer and hidden layer, hidden layer and output layer, respectively , $b_h, b_{\hat{y}}$ are bias term, and f and g are activation functions. For a deep RNN with several stacked hidden layers, except the first hidden layer receives inputs vector as its inputs, the each hidden layer later receives the output of previous hidden layer as its input.

In a standard RNN, recurrent connections between hidden units allow the network can remember information from previous time steps. RNNs, therefore, are very well suited to adopt sequential inputs. However, the difficulty of training an RNN to capture long-term dependencies problem that gradients propagated over many stages tend to either vanish or explode. The exploding gradient can be easy treat by clipping the gradient – clipping the parameter gradient from a mini-batch element-wise just before the parameter update or clipping the norm of the gradient just before the parameter update [22], while the vanishing gradients problem is more challenging to solve because of the long-term dependencies problem. Several techniques have been proposed to combat the vanishing gradients problem such as long short-term memory (LSTM) [23] and gated recurrent units (GRU) [24]. LSTM and GRU architectures extend the standard RNN by replacing the simple neurons that have self-connections with units are called memory block to accumulate information, which is better to capture long-term dependencies in the time-series data.

III. Methods

In this section, we review of metrics that used to evaluate the performance of SED system. In addition, we summary approaches and compare the performance of deep learning models that adopted SED problem. Moreover, we describe a short description about benchmarking SED datasets that have been introduced by DCASE challenge [25].

A. *Metrics*

The evaluations were used for SED task including segment-based error rate and segment-based F1-score. The segment-based error rate is measured based on the number of errors in term of insertions (I), deletions (D) and substitutions (S). A substitution is determined if an event is detected in a given segment but gives it a wrong label. After the number of substitutions per segment is count, insertions are the remaining false positives in the system output and the remaining false negatives are considered as deletions. Next, a number of active ground truth events in segment k are calculated. Then, the total error rate is calculated as

$$ER = \frac{\sum_{k=1}^{K} S(k) + \sum_{k=1}^{K} D(k) + \sum_{k=1}^{K} I(k)}{\sum_{k=1}^{K} N(k)} \tag{3}$$

The second evaluation metric is segment-based F1 score is computed based on three statistics - the number of false positive (FP), false negative (FN), and true positive (TP). Specifically, false positive (FP) if an event is detected in the given segment but it does not appear in the same given segment of the annotated data, false negative (FN) if an event is not detected in the given segment but it appears in the same given segment of the annotated data, and true positive (TP) if an event is detected in the given segment and it also appears in the same given segment of the annotated data. These statistics are accumulated over test data. Then, based on these accumulated statistics, precision (P), recall (R) and F-score are computed according to the formula as

$$P = \frac{TP}{TP+FP}, R = \frac{TP}{TP+FN}, F1 = \frac{2PR}{P+R} \tag{4}$$

B. Dataset

Recently, the most widely used benchmarking SED datasets are TUT Sound events 2016 and TUT Sound events 2017 that are provided by Tampere University of Technology in DCASE challenge 2016-2017 [25].

TUT Sound events 2016 dataset consists of stereo recordings from two acoustic scenes: home and residential area. The home scene contains 11 sound events including rustling, snapping, cupboard, cutlery, dishes, drawer, glass jingling, object impact, people walking, washing dishes and water tap running. The residential scene consists of 7 sound events including banging, bird singing, car passing by, children shouting, people speaking, people walking, wind blowing. The recordings were recorded in different streets and different homes. Each recording has the length of 3-5 minutes.

TUT Sound events 2017 dataset includes recordings of different streets with various levels of traffic and other activities. For each location, a 3-5 minute binaural audio recording is captured with sampling frequency of 44,100 Hz and 24 bit resolution. The recordings are manually annotated. To be more precise, there are six annotated sound event classes for street recordings including brakes squeaking, car, children, large vehicle, people speaking, and people walking.

C. Deep Learning approaches

Results for SED problem are summarized in Table 1. All results are reported on three benchmarking SED datasets – the database used in CLEAR 2007 evaluation [26], TUT Sound events 2016, and TUT Sound events 2017. The best performance is acknowledged for deep learning approaches on three datasets.

TABLE 1: F1 SCORE AND ERROR RATE RESULTS OF APPROACHES FOR SED TASK. BOLD NUMBER INDICATES THE BEST PERFORMING METHOD FOR THE GIVEN DATASET.

Methods	F1-score	Error rate
The database used in CLEAR 2007 evaluation [26]		
HMM [5]	28.3	0.87
HMM-GMM [6]	40.1	0.842
NMF [7]	57.8	-
DNN [8]	**63.8**	-
TUT Sound events 2016 evaluation dataset		
GMM [25]	34.3	0.877
RNN [16]	39.6	0.905
RNN [12]	41.9	0.912
RNN [11]	**47.8**	**0.805**
TUT Sound events 2017 evaluation dataset		
CNN [10]	30.9	0.857
CNN [9]	40.8	0.808
RNN [14]	37.3	0.852
RNN [15]	39.6	0.825
CNN-RNN [17]	**41.7**	**0.791**

Deep neural networks: Cakir *et al*. [8] proposed a multi-label feed–forward deep neural network (DNNs) for polyphonic SED task. In particular, the proposed method used three kinds of features, namely MFCCs, mel-band energies, and log mel-band energies and DNNs with two hidden layers of 800 units as a classifier. The model is evaluated with recordings from realistic everyday environments and the achieved overall accuracy is 63.8%. The model outperforms HMM-GMM [6] with the large margin of over 20%.

Convolutional neural networks: More recently, the works by Jeong *et al.* [9] shown a good performance when applied CNN for SED task. The work used both short-term and long-term audio signal as input features to feed into ConvNet architectures. They used the 1-dimensional convolution layer with 64 filters. The proposed model obtained better results than the baseline system [25] in both term error rate and f-score.

Recurrent neural networks: Several works have recently used RNNs for addressing SED task such as the works in [11, 12, 13, 14, 15, 16]. Almost the proposed methods are based on bidirectional LSTM or GRU architectures using log mel-band energies as features. Bidirectional LSTM and GRU are powerful techniques to exploit context information in both direction, capture long-term information and reduce over-fitting problems. In comparison to DNNs and CNNs architecture, RNNs work better with the input data has sequential structure.

Convolutional recurrent neural networks: For SED problem, the audio signals are sequential data involving time. RNNs are neural networks that are specialized for processing time-series data. RNNs can integrate information from earlier time windows and even may have connections that go backward in time, presenting a theoretically unlimited context information. However, an audio input presents its features on both time and frequency domains. While RNNs only work well on time domains, CNNs can apply linear convolutional filters in both time and frequency domains of the local features. In order to take advantages of from both approaches, a combination of convolutional networks and recurrent networks often referred as a convolutional neural network (CRNN) have been adopted for SED task [17,18]. CRNN outperforms all previous SED methods.

IV. Conclusions

In this work, we review of the use of deep learning approaches for sound event detection task. In comparison with other surveys, this paper focuses on such a rising topic as deep learning and presents the most advanced and recent SED works using deep learning methods. We also covered the literature of benchmarking SED datasets and metrics that are used to evaluate in SED task.

References

[1] Alex Garve, Abdel-rahman Mohamed, Geoffrey Hinton, “Speech recognition with deep recurrent neural network,” in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2013, pp. 6645–6649.

[2] Yanmin Qian, Philip C Woodland, “Very deep convolutional neural networks for robust speech recognition” in arXiv:1610.00277v1, 2016.

[3] Alex Krizhevsky, Ilya Sutskever, Geoffrey E Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in Neural Information Processing Systems (NIPS) 2012, pp. 1097–1105.

[4] Alexis Conneau, Holger Schwenk, Loïc Barrault, Yann Lecun, “Very Deep Convolutional Networks for Text Classification,” in arXiv:1606.01781, 2016.

[5] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, “Acoustic event detection in real-life recordings,” in Proc. 18th Eur. Signal Process. Conf., Aalborg, Denmark, Aug. 2010, pp. 1267–1271.

[6] T. Heittola, A. Mesaros, A. J. Eronen, and T. Virtanen, “Context-dependent sound event detection,” EURASIP J. Audio, Speech, Music Process., vol. 1, pp. 1– 13, 2013.

[7] Annamaria Mesaros, Toni Heittola, Onur Dikmen, Tuomas Virtanen, “Sound event detection in real life recordings using coupled matrix factorization of spectral representations and class activity annotations,” in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2015, pp. 606–618.

[8] Emre Cakir, Toni Heittola, Heikki Huttunen, Tuomas Virtanen, “Polyphonic Sound Event Detection Using Multi Label Deep Neural Networks,” in IEEE International Joint Conference on Neural Networks (IJCNN) 2015.

[9] Il-Young Jeong, Subin Lee, Yoonchang Han, Kyogu Lee, “Audio event detection using multiple-input convolutional neural network,” in Detection and Classification of Acoustic Scenes and Events 2017, Tech. Rep., 2017.

[10] Yukun Chen, Yichi Zhang and Zhiyao Duan, “DCASE2017 sound event detection using convolutional neural network,” in Detection and Classification of Acoustic Scenes and Events 2017, Tech. Rep., 2017.

[11] S. Adavanne, G. Parascandolo, P. Pertila, T. Heittola, and T. Virtanen, “Sound event detection in multichannel audio using spatial and harmonic features,” IEEE Detection and Classification of Acoustic Scenes and Events workshop, 2016.

[12] T. H. Vu and J.-C. Wang, “Acoustic scene and event recognition using recurrent neural networks,” Detection and Classification of Acoustic Scenes and Events 2016, Tech. Rep., 2016.

[13] G. Parascandolo, H. Huttunen, and T. Virtanen, “Recurrent neural networks for polyphonic sound event detection in real life recordings,” in 2016 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), 2016, pp. 6440–6444.

[14] Rui Lu, Zhiyao Duan, “Bidirectional GRU for sound event detection,” in Detection and Classification of Acoustic Scenes and Events 2017, Tech. Rep., 2017.

[15] Jianchao Zhou, “Sound event detection in multichanel audio LSTM network,” in Detection and Classification of Acoustic Scenes and Events 2017, Tech. Rep., 2017.

[16] Matthias Zohrer and Franz Pernkopf, “ Gated recurrent networks applied to acoustic scene classification and acoustic event detection”, in Detection and Classification of Acoustic Scenes and Events 2016, Tech. Rep., 2016.

[17] Sharath Adavanne, Tuomas Virtanen, “A report on sound event detection with different binaural features,” in Detection and Classification of Acoustic Scenes and Events 2017, Tech. Rep., 2017.

[18] E. Cakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, “Convolutional recurrent neural networks for polyphonic sound event detection,” in IEEE/ACM TASLP Special Issue on Sound Scene and Event Analysis, 2017.

[19] Dan Stowell, Dimitrios Giannoulis, Emmanouil Benetos, Mathieu Lagrange, and Mark D. Plumbley, “Detection and classification of acoustic scenes and events,” in IEEE transactions on multimedia, vol. 17, no. 10, october 2015.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” in Proceedings of the IEEE 86(11): 2278–2324, 1998.

[21] Hinton, G. E. and Sejnowski, T. J. (1986), “Learning and relearning in Boltzmann machines,” in D. E. Rumelhart and J. L. McClelland, editors, Parallel Distributed Processing, vol. 1, chap. 7, pp. 282–317. MIT Press, Cambridge.

[22] Razvan Pascanu, Tomas Mikolov, Yoshua Bengio, “On the difficulty of training Recurrent Neural Networks,” in arXiv:1211.5063, 2013.

[23] Sepp Hochreiter, Jürgen Schmidhuber, “Long short-term memory,” in Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] Kyunghyun Cho, Bart Merriënboer, Dzmitry Bahdanau, Yoshua Bengio, “On the properties of neural machine translation: Encoder-decoder approaches,” in arXiv:1409.1259, 2014.

[25] Annamaria Mesaros; Toni Heittola; Tuomas Virtane, “TUT database for acoustic scene classification and sound event detection,” in 2016 24th European Signal Processing Conference (EUSIPCO), 2016.

[26] A Temko, C Nadeu, D Macho, R Malkin, C Zieger, M Omologo, in Computers in the Human Interaction Loop, ed. by AH Waibel, R Stiefelhagen. Acoustic event detection and classification (Springer, New York), pp. 61–73, 2009.