MULTI-LABEL VS. COMBINED SINGLE-LABEL SOUND EVENT DETECTION WITH DEEP NEURAL NETWORKS

Emre Cakir, Toni Heittola, Heikki Huttunen and Tuomas Virtanen

Department of Signal Processing, Tampere University of Technology, Finland

ABSTRACT

In real-life audio scenes, many sound events from different sources are simultaneously active, which makes the automatic sound event detection challenging. In this paper, we compare two different deep learning methods for the detection of environmental sound events: combined single-label classification and multi-label classification. We investigate the accuracy of both methods on the audio with different levels of polyphony. Multi-label classification achieves an overall 62.8% accuracy, whereas combined single-label classification achieves a very close 61.9% accuracy. The latter approach offers more flexibility on real-world applications by gathering the relevant group of sound events in a single classifier with various combinations.

Index Terms— Sound event detection, deep neural networks, multi-label classification, binary classification, audio analysis

1. INTRODUCTION

Sound event detection (SED) systems aim to recognize and distinguish particular events related to human, nature or machine presence. In realistic environments, there are often multiple sound sources and the sound events originating from them can overlap in time. Birds singing, footsteps, motorbike engine etc. can be given as examples for the sound events in realistic environments. SED systems tackle the problem for two different cases: monophonic and polyphonic detection. In monophonic detection, the aim is to find the prominent event in the sound data. It is used in video keyword tagging [1] and real-life event and context detection [2, 3].

Polyphonic SED is capable of detecting multiple sound events in the same time instance of the sound data. Each instance is associated with a set of labels, *i.e.*, the labels of the sound events that are active in the given instance. The aim is to map each instance with its associated label set. The number of sound events active in an instance is not known *a priori*, which introduces a different level of complexity in detection.

Polyphonic SED systems require multi-label classification, which is not widely experimented in audio information retrieval tasks. Generalized Hough transform (GHT) voting system has been used to recognize overlapping sound events by summing up the local spectrogram keypoint information to produce onset hypotheses [4]. In [5], non-negative matrix factorization (NMF) has been used to first decompose the audio into streams and then recognize a single event from each stream by using prominent stream selection or stream elimination. In our previous work we proposed to use multi-label deep neural networks (DNN) for polyphonic SED and showed that it exceeds the state-of-the-art NMF + hidden Markov model (HMM) based approach [5] in accuracy [6].

DNNs are classifiers that are capable of extracting high level representations of their inputs through the multiple hidden layers. This has been found to provide better discrimination capability in certain pattern recognition tasks. Deep learning methods have recently given state-of-the-art results for many applications in environmental SED [2, 6] and speech recognition [7].

In this paper, we explore the use of DNNs in environmental SED with two different approaches: multi-label (ML) and combined single-label (CSL) methods. The proposed methods are illustrated in Figure 1. First, we train DNNs with multi-label outputs with polyphonic material in a supervised setting. Then, we train several DNNs with single-label outputs again with the same polyphonic material in a supervised setting. We combine the outputs of the single-label DNNs to obtain a multi-label output for each time instance. In [8], it is claimed that decomposing a multi-label classification into several binary classification problems will lose the correlation information between different labels in a single instance. However, the flexibility of making different sets of labels for different applications can be valuable and useful at the expense of slightly decreased accuracy for some applications, especially in SED systems. Moreover, using a set of single-label classifiers allows dynamic inclusion of new labels by training classifiers only for the new sound events instead of re-training the complete framework. To the best of our knowledge, this is the first work that compares these two deep learning approaches on polyphonic SED. Both methods are experimented on realistic sound material with single element.

The organization of this paper is as follows. The problem is stated and the DNN input and target output is explained in Section 2. The methodology, including ML and CSL DNN classification methods, are explained in detail in Section 3. The experiment material, evaluation procedure and experimental results are given in Section 4. Finally, comments and conclusions are given in Section 5.

978-0-9928626-3-3/15/$31.00 ©2015 IEEE

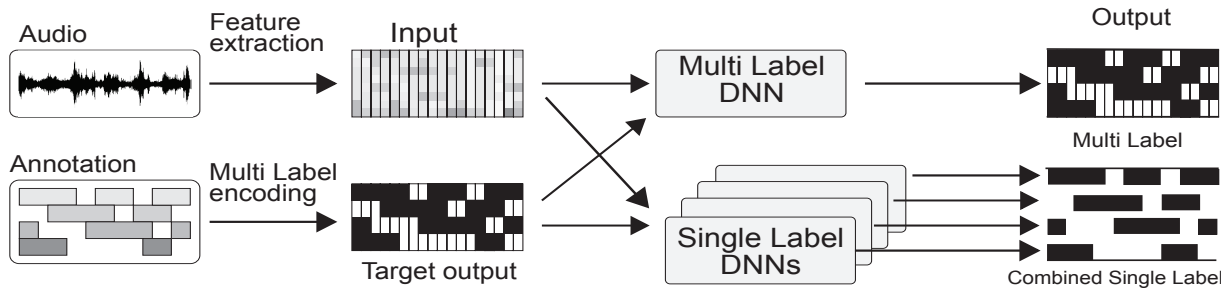

Fig. 1: Framework for the proposed CSL DNN and ML DNN classification methods.

2. PROBLEM STATEMENT

The goal of polyphonic SED is to estimate the start and end times of sound events in an audio signal, and classify the events into their predefined classes. When the audio is processed in short frames, this can be viewed as a multi-label classification problem. Multi-label learning tackles the problems where each instance in the training set can be associated with multiple labels. When it comes to multi-label environmental SED, the sound data typically contains overlapping events, *e.g*, a sound recording taken from a street may contain traffic noise, speech and the rain sound all at the same time.

2.1. Audio Representation

In order to do robust classification on the polyphonic material, one should choose the features that makes a good discrimination over the possible outcomes. Mel band energies are used as audio features in this work, since they have been proved to obtain better performance over the traditional Mel frequency cepstral cofficients (MFCC) in polyphonic SED and speech recognition [6, 7]. The reasoning for this can be that DNNs do not especially require their inputs to be uncorrelated and by applying discrete cosine transform (DCT), MFCCs discard some information from the audio material [9]. The recordings are first amplitude normalized, divided into frames and then short-time Fourier transform (STFT) is applied on 50 ms time frames with 50% overlap. Mel filterbank with 40 Mel bands is used to extract the feature vector $\mathbf{u}_t$ in each time frame, where t denotes the position in the domain.

In order to make use of the dynamic properties of the audio, the feature vectors are concatenated with their two preceding and two succeeding feature vectors. This method is known as *context windowing*. Concatenated feature vector $\mathbf{x}_t$ is obtained as $\mathbf{x}_t = [\mathbf{u}_{t-2}^{\mathrm{T}}\ \mathbf{u}_{t-1}^{\mathrm{T}}\ \mathbf{u}_t^{\mathrm{T}}\ \mathbf{u}_{t+1}^{\mathrm{T}}\ \mathbf{u}_{t+2}^{\mathrm{T}}]^{\mathrm{T}}$, where $\mathbf{u}_t$ denotes the extracted feature vector. The concatenated feature vector $\mathbf{x}_t$ is used as a single training instance for the DNN.

2.2. DNN Target Output

The training of the network is performed in a supervised setting. The start and end times for each sound event in a recording are manually annotated each time they occur in the recording. For each time frame, a target output vector $\mathbf{y}_t$ of length N is obtained, where N is the total number of possible sound events. The elements of the target vector $\mathbf{y}_t$ are binary and determined as

$$y_t(l) = \begin{cases} 1, & \text{if } l^{th} \text{ event is active in frame } t \\ 0, & \text{if } l^{th} \text{ event is not active in frame } t \end{cases} \quad (1)$$

where $y_t(l)$ is the l^{th} entry of target output vector $\mathbf{y}_t$ and $1 \leq l \leq N$.

3. METHODOLOGY

We consider two methods for encoding the presence of simultaneous events in an audio recording. One method is to train a single-label classifier for each label l and then combine the outputs from each classifier to obtain a multi-label output. Second method is to train a multi-label classifier, which produces a multi-label detection output vector.

3.1. Combined Single-Label DNN classification

For the CSL DNN classification, each label l is trained and tested with a different DNN, independent from other labels. The input features are extracted from polyphonic sound signals. The reasons for using polyphonic signals is as follows. Firstly, even for single-label classification, the sound events are hardly ever isolated in a realistic environment and it is difficult to separate signals produced by individual sources. Secondly, using polyphonic data makes the comparison between the CSL DNN and the ML DNN methods easier to interpret and analyze.

CSL DNN provides significant flexibility on real-world applications. To illustrate, if the number of sound events in a database is N, then N different models can be trained and grouped together in various combinations depending which of the classes are of interest in a certain application. Besides, new classes can be easily added to the overall CSL DNN system by training a single-label DNN for the new class with the additional database.

The single-label DNN architecture is composed of an input layer, two or more hidden layers and output layer with a

single output unit. Fully-connected feed-forward DNNs are used in this work. Starting from $\mathbf{h}^1 = \mathbf{x}$, the outputs $\mathbf{h}^k$ of the units for layer k are calculated as

$$\mathbf{g}^k = \mathbf{W}^k \mathbf{h}^{k-1} + \mathbf{b}^k, 2 \leq k < M \quad (2)$$

$$\mathbf{h}^k = f(\mathbf{g}^k) \quad (3)$$

where $\mathbf{W} \in \mathbb{R}^{D \times S}$ is the weight matrix between layers $k-1$ and k, D and S are the number of units for layers $k-1$ and k, respectively, $\mathbf{b} \in \mathbb{R}^S$ is the bias vector for layer k, $f(\cdot)$ is the activation function applied element-wise on the output of each unit in layer k, and M is the total number of layers in the DNN. For the hidden layer activation functions, *maxout* [10] function is used. Instead of applying a conventional non-linearity on the weighted sum $\mathbf{g}^k$, maxout function groups the weighted sums and passes the maximum to $\mathbf{h}^k$, increasing the sparsity of the gradients and preventing the network suffering from the vanishing gradients since the activation outputs are unbounded [11]. For the output layer activation function, the more conventional logistic sigmoid function is chosen. Since the sigmoid activation function output h^M is bounded between 0 and 1, it is possible and logical to interpret the DNN output as the detection probability. For each training instance $\mathbf{x}_t$, the CSL DNN output with single element h^M is used as the source-presence prediction $\hat{y}_t$.

Each single-label DNN is trained with the corresponding target output $y_t(l)$ for label l. In order to estimate the distance between the source-presence prediction and the target output for label l, cross-entropy cost function $C_l(\hat{y}_t, y_t(l))$ is calculated as

$$C_l(\hat{y}_t, y_t(l)) = -y_t(l)\log(\hat{y}_t) - (1 - y_t(l))\log(1 - \hat{y}_t) \quad (4)$$

where $y_t(l)$ is either 0 or 1 and $\hat{y}_t \in [0, 1]$. $C_l(\hat{y}_t, y_t(l))$ is guaranteed to be non-negative and when $y_t(l)$ and $\hat{y}_t$ are closer to each other, it goes closer to zero. Therefore, cross-entropy cost function is to be minimized by updating the weight $\mathbf{W}$ and bias $\mathbf{b}$ vectors. For this purpose, stochastic gradient descent algorithm (SGD) is used. The gradients in each layer are computed using the backpropagation algorithm [12].

When the separate training for each label is finished, the test instances are evaluated by the single-label DNNs and the source-presence predictions $\hat{y}_t$ are obtained. The source presence-predictions from each single-label DNN are combined in the multi-label vector $\hat{\mathbf{y}}_t = [\hat{y}_t(1)\ \hat{y}_t(2)...\hat{y}_t(N)]$. Then, $\hat{\mathbf{y}}_t$ is binarized by thresholding with a certain constant, leading to a binary estimation vector $\mathbf{z}_t$ of length N. The effects of the binarizing threshold is examined in Section 4.

3.2. Multi-Label DNN classification

ML DNN training differs from the CSL DNN training only in the way that the number of units in the output layer is equal to the number of sound events N, therefore we get the source-presence prediction vector $\hat{\mathbf{y}}_t$ of length N for each frame t. This leads to another information to be learned: the correlation structure between the events. Some of the events may appear together in a large number of training instances and it can be a valuable information for the DNN.

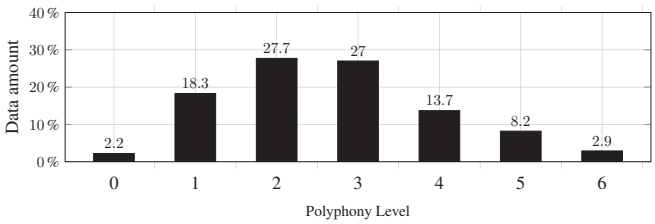

Fig. 2: The percentage of the amount of the test material as a function of the polyphony level.

Instead of calculating the cross-entropy cost function for a single-label, ML DNN computes the cost function as

$$C(\hat{\mathbf{y}}_t, \mathbf{y}_t) = -\mathbf{y}_t \cdot \log(\hat{\mathbf{y}}_t) - (1 - \mathbf{y}_t) \cdot \log(1 - \hat{\mathbf{y}}_t) \quad (5)$$

where the operator $(\cdot)$ denotes the dot product and the logarithm operator is applied element-wise. The cost value is the sum of the costs over each label and therefore depends on the source-presence predictions for each label l.

3.3. Post-processing

Our experiments with realistic audio material showed that environmental sound events typically have some short bursts of less active periods. To illustrate, a dog gives a small break to breathe before each bark, or the footsteps make sounds periodically. Since the annotation of the audio material is done with a rather coarse time resolution, these less active bursts are mapped to a sound event of which they do not possess the spectral characteristics. This results with a rather noisy behaviour in DNN outputs.

A median filtering based post-processing approach is implemented to filter this noise and smoothen the outputs in the testing stage for both CSL DNN and ML DNN. For each frame, the post-processed output $\hat{\mathbf{z}}_t$ is obtained by taking the median of the binary outputs $\mathbf{z}_t$ in a 10-frame window (corresponds to 250 ms of audio) as

$$\hat{\mathbf{z}}_t = \begin{cases} 1, \text{if median}(\mathbf{z}_{(t-9):t}) = 1 \\ 0, \text{otherwise} \end{cases} \quad (6)$$

The method is applied on each label separately and continued by sliding this 10-frame window when every new frame is processed through the DNN.

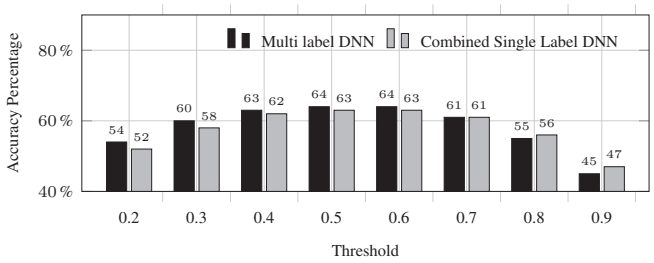

Fig. 3: Detection accuracy vs. binarizing threshold for ML DNN and CSL DNN.

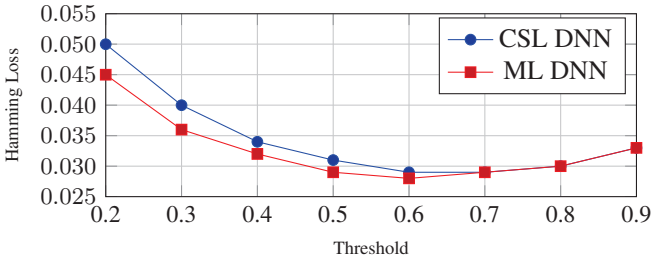

Fig. 4: Hamming loss vs. threshold for CSL DNN and ML DNN classification.

4. EXPERIMENTS AND RESULTS

4.1. Sound Material and Setting

The evaluation of both CSL and ML DNN methods are performed on a sound database collected from highly realistic everyday environments. Recordings from 10 different environments, such as beach, bus, street etc. are used to gather a database of 1133 minutes. From the recordings, 61 most prominent events, such as clapping, dogs barking, cash register beep etc. are selected to be evaluated. The recordings have varying number of active sound events in each instance, *i.e.*, the frames have varying polyphony levels. The amount of test material according to their polyphony levels are illustrated in Figure 2. The label cardinality, *i.e.*, the average number of active events in each frame is 2.55. The database is divided into non-overlapping groups as 60% training, 20% test and 20% validation sets. Validation set is not used in training and it is required to determine the optimum parameters without overfitting the network on the training set. More detailed information on the sound database can be found in [13].

DNN hyper-parameters such as learning rate, hidden unit number, initial weight and bias range etc. are selected by doing a grid search over possible values to get the best accuracy on the validation set. The best performance is obtained with two hidden layers of 800 units for ML DNN and two hidden layers of 400 units for CSL DNN. The learning rates for both methods are 0.02.

4.2. Evaluation Metric

The methods are evaluated by using two different metrics that are commonly used in multi-label evaluation. First one is the block-wise F1 score evaluation metric. A sound event is regarded as correctly detected if it is marked as present in any instance of the time block and if it is also present in the annotations of the time block. Missed and wrongly detected events are calculated in the same manner. This approach fits well with the goal of environmental SED, since it is rather interested in detecting the event rather than the exact start and end times. Precision and recall are calculated according to the number of correctly detected, missed and wrongly detected events. F1 score, the harmonic mean of the precision and recall, is calculated in non-overlapping one second blocks. The final F1 score is calculated by averaging the F1 scores in the one second time blocks of the test dataset and presented as the accuracy percentage.

The second multi-label evaluation metric is Hamming loss [14]. It evaluates how many times a frame is misclassified. It implements exclusive-or (*xor*) operation between binary estimation vector $\hat{\mathbf{z}}_t$ and target output vector $\mathbf{y}_t$ as

$$\text{Hamming loss}(\hat{\mathbf{z}}, \mathbf{y}) = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{N}(\hat{\mathbf{z}}_t \,\Delta\, \mathbf{y}_t) \tag{7}$$

where T is the number of time frames, N is the number of sound events and the operator Δ gives the symmetric difference between $\hat{\mathbf{z}}_t$ and $\mathbf{y}_t$ as

$$\hat{\mathbf{z}}_t(l) \,\Delta\, \mathbf{y}_t(l) = \begin{cases} 0, & \text{if } \hat{\mathbf{z}}_t(l) = \mathbf{y}_t(l) \\ 1, & \text{otherwise} \end{cases} \tag{8}$$

4.3. Results

While converting the DNN outputs $\hat{\mathbf{y}}_t$ into binary form as $\mathbf{z}_t$, several threshold values have been experimented. For various thresholds, the average F1 score is presented as the accuracy percentage for ML DNN and CSL DNN in Figure 3. Both methods provide a huge improvement over the state-of-the-art NMF+HMM-based method in [5], which provides 44.9% accuracy on the same database. For both methods, the accuracy takes its maximum value around threshold 0.5, which indicates that DNN outputs make a balanced probability distribution estimation between 0 and 1. Hamming loss results from Figure 4 also supports this theory. Hamming losses for both methods reach to their minimum around threshold 0.6 (note that they are very close for thresholds above 0.7). ML DNN classification provides a 2-3% better accuracy compared to CSL DNN for low threshold values, but the situation reverses for higher threshold values. This can be explained with the fact that the whole activity of a single frame is bundled in one single DNN output for CSL DNN, whereas in ML DNN, it is distributed in all the events. Therefore, in a polyphonic

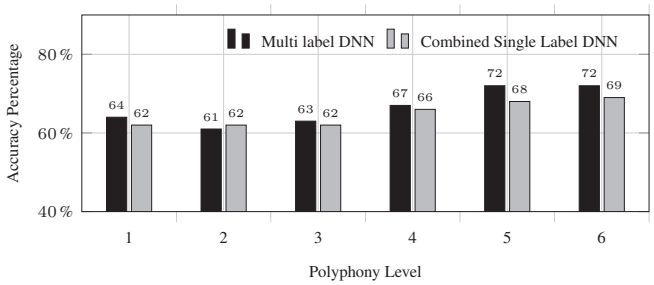

Fig. 5: Detection accuracy vs. polyphony level.

frame, for each active event, the source-presence predictions are higher for CSL DNN than ML DNN and better discrimination is obtained with high threshold values.

The detection accuracy is calculated separately for different levels of polyphony and examined in Figure 5. The binarizing threshold is set to 0.5. When the accuracy is averaged among the polyphony levels according to the data amount for each polyphony level, CSL DNN achieves an overall 61.9% accuracy, while ML DNN achieves 62.8% accuracy. CSL DNN classification provides very similar accuracy compared to ML DNN, regardless of the polyphony level. The results show that decomposing a multi-label sound event classification problem into multiple single-label problems do not suffer from losing the correlation structure between the labels.

5. CONCLUSIONS AND COMMENTS

In this paper, two different deep learning methods are proposed and compared for polyphonic SED in real-life environments. The first method consists of using a multi-label DNN for classification of all possible sound events, whereas the second method uses a single-label DNN for each single sound event and combines the outputs of each DNN for a single time frame. Although the hypothesis was that CSL DNN would be affected from losing the correlation information, it provides very similar accuracy compared to ML DNN. CSL DNN also presents multiple implementation options by grouping different event models together. For the future work, it would be interesting to apply CSL DNN on other multi-label classification problems. Also, context dependent CSL DNN methods can be experimented by grouping the CSL DNN models for the events that are likely to occur together, thus creating a single classifier for a certain context. Finally, an interesting path would be to apply other multi-label learning methods on sound event detection and see if our conclusion for multi-label DNN learning is extensible for other approaches.

REFERENCES

[1] D. Zhang and D. Ellis, "Detecting sound events in basketball video archive," *Dept. Electronic Eng., Columbia Univ., New York*, 2001.

[2] 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.

[3] S. Chu, S. Narayanan, and C-CJ Kuo, "Environmental sound recognition with time–frequency audio features," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1142–1158, 2009.

[4] J. Dennis, H.D. Tran, and E.S. Chng, "Overlapping sound event recognition using local spectrogram features and the generalised hough transform," *Pattern Recognition Letters*, vol. 34, no. 9, 2013.

[5] T. Heittola, A. Mesaros, T. Virtanen, and M. Gabbouj, "Supervised model training for overlapping sound events based on unsupervised source separation," in *Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013, pp. 8677–8681.

[6] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multilabel deep neural networks," in *Int. Joint Conf. on Neural Networks (IJCNN)*, 2015, accepted.

[7] J. Li, D. Yu, J. Huang, and Y. Gong, "Improving wideband speech recognition using mixed-bandwidth training data in CD-DNN-HMM," in *Spoken Language Technology Workshop*, 2012, pp. 131–136.

[8] M. Zhang and Z. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE Trans. Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1338–1351, 2006.

[9] G. Hinton, L. Deng, D. Yu, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[10] I.J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," in *ICML*, 2013.

[11] Pawel Swietojanski, Jinyu Li, and Jui-Ting Huang, "Investigation of maxout networks for speech recognition," in *Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, 2014, pp. 7649–7653.

[12] P. Werbos, *Beyond regression: New tools for prediction and analysis in the behavioral sciences*, Ph.D. thesis, Harvard Univ., Comm. Appl. Math., 1974.

[13] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic event detection in real life recordings," in *Proc. 18th European Signal Processing Conference (EUSIPCO)*, 2010, pp. 1267–1271.

[14] R.E. Schapire and Y. Singer, "Boostexter: a boosting-based system for text categorization," *Machine Learning*, vol. 39, no. 2-3, pp. 135–168, 2000.