DEEP CLUSTERING FOR SINGING VOICE SEPARATION

Yi Luo
Department of Electronic Engineering
Columbia University
yl3364@columbia.edu

Zhuo Chen
LabROSA
Columbia University
zc2204@columbia.edu

Daniel P. W. Ellis
LabROSA
Columbia University
dpwe@columbia.edu

ABSTRACT

This extended abstract describes the system we submitted for the singing voice separation task of MIREX 2016. Our submission here is an extension of the deep clustering network from [1].

1. INTRODUCTION

Deep neural networks have shown to be effective on single channel source separation tasks in recent years. For example, PS Huang et al. first applied deep recurrent neural network (DRNN) on music for singign voice separation task [2], and for speech denoising task (separate noise from speech) [3]. Recently, deep clustering, which is a special type of deep neural network, has shown to perform surprisingly well on the task of single-channel speech separation problem. In this submission, we investigate the effectiveness of deep clustering on the task of singing voice separation.

2. MODEL DESCRIPTION

The basic idea of deep clustering is to generate an embedding for each time-frequency (T-F) unit so that T-F units that belongs to the same source should have similar embeddings (under Euclidean distance), hence we can use clustering methods to cluster the embeddings and generate T-F masks. The structure of deep clustering network is a stack of several recurrent layers, followed by a feedforward layer. The input of deep clustering network is a T-F representation of the raw input signal, and we assume that the T-F representation can be partitioned into sets of T-F bins in which each source dominates. A T-F mask can then be estimated by calculating the dominant sources for each T-F bin, corresponds to the T-F components for each source which is uncorrupted by other sources [1]. The target is a label-assigning matrix $Y \in R^{(T\times F)\times C}$, which C is the number of categories (number of sources) in the input mixture, then $Y_{i,j} = 1$ means T-F component i belongs to category j. We can construct a binary affinity matrix $A = YY^T$, which represents the assignment of the sources in a permutation independent way: $A_{i,j} = 1$ if i and j belong to the same class, and $A_{i,j} = 0$ if not. The network estimates an embedding matrix $V \in R^{(T\times F)\times N}$ for each T-F component, where N is the dimension of the embedding. The affinity matrix of the embedding matrix is then defined as $\hat{A} = VV^T$. The cost function for the network is

$$C(V) = ||\hat{A} - A||_F^2 = ||VV^T - YY^T||_F^2 \quad (1)$$

Although the matrix A and $\hat{A}$ are all extremely large $((T \times F) \times (T \times F))$, we can make use of the low-rank structure of them and greatly decrease the computational complexity.

During test time, we cluster the columns of embedding matrix V using K-means++ algorithm. The resulting cluster assignments are then used as binary masks to estimate the T-F representation of the separated sources.

The original deep clustering model did not use any regularizers, but here we introduce two regularization techniques: batch normalization and dropout.

Batch normalization is an effective way to address problem of covariance shift during training [4], and it has been proved to be very helpful in feedforward and convolutional networks. Suppose $E[x]$ and $Var[x]$ are the mean and variance over a batch respectively, the normalized input is

$$\hat{x} = \frac{x - E[x]}{\sqrt{Var[x] + \epsilon}} \quad (2)$$

where ϵ is a small positive constant for numerical stability.

A variant to the original batch normalization, which is called sequence-wise batch normalization [5], is designed for recurrent networks. Sequence-wise batch normalization computes the mean and variance statistics over all items in the minibatch over the length of the sequence, instead of averaging over all time-steps. This has been shown to be a success in automatic speech recognition system [6]. Here we only apply batch normalization to the input-to-hidden transform.

Dropout is a famous regularization technique that randomly set some outputs of a layer to zero with certain probability [7]. This technique is proved to be very powerful on improving the robustness of a network. Here we only use the 'standard' form of dropout, i.e. only on the feedforward connections.

3. EXPERIMENT SETUP

We use the DSD100 dataset for SiSEC [8] for training. It contains 100 songs of different styles, and all the songs are professionally-produced and mixed using real professional Digital Audio Workstations. Here we only make use of the Dev set (50 songs) in DSD100 to generate a training set and a development set. To generate enough training data, we first downsample the music from 44.1kHz to 16kHz in order to reduce computational cost, then randomly mix different sources together at 0dB. Note that since different tracks in a song may be correlated with each other in terms of harmonics, beats, etc., so here we always guarantee that there the original mapping of the sources (i.e. the true mixture for each song) would always be in the training dataset. The total length of the training dataset is 15h, and the total length of the development dataset is 0.5h.

The input feature we use is calculated by STFT with 512-point window size and 128-point hop size. Since deep clustering will fail if the dimension of the input feature is too large, here we use a 150-dimension mel-filterbank to reduce the dimension of input feature. We use ideal binary mask calculated on the mel-filterbank spectrogram as target.

We use 4 BLSTM layers of 500 hidden units in each layer and a feedforward layer of 3000 hidden units. The number of hidden units in the feedforward layer is the product of the dimension of the embedding vector (20 here) and the dimension of the input feature (150 here). Dropout layers with probability 0.2 is added between each two feedforward connections, and sequence-wise batch normalization is applied in the input-to-hidden transformation in each BLSTM layer. The learning algorithm we choose is rmsprop, and we set the maximum number of epoch to be 100. The chunk size for the input is set to be 100 frames (roughly 0.8 second). We select the best network according to the performance on the validation set.

In order to make the advantage of curriculum training, after training a network using chunk size 100, we increase the chunk size to 500 and continue training the network.

During test time, we split input feature into several chunks of 500 frames long, and pad zero at the end if the last chunk is less than 500 frames. Each of the chunks is then feeded into the network, and after all chunks are processed, we apply k-means++ clustering on all the embeddings generated, which leads to two T-F masks. We apply the two masks to the mel-filterbank spectrogram, and recover the signal using inverse mel-filterbank and inverse-STFT from them. Finally we resample the recovered signals from 16kHz to 44.1kHz.

4. REFERENCES

[1] Hershey J R, Chen Z, Roux J L, et al. Deep clustering: Discriminative embeddings for segmentation and separation[J]. arXiv preprint arXiv:1508.04306, 2015.

[2] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, *Singing-voice separation from monaural recordings using deep recurrent neural networks*. International Society for Music Information Retrieval, pages 477482, 2014

[3] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, *Joint Optimization of Masks and Deep Recurrent Neural Networks for Monaural Source Separation*. IEEE Transactions on Audio, Speech, and Language Processing, 23.12 (2015):2136-2147

[4] Ioffe S, Szegedy C. *Batch normalization: Accelerating deep network training by reducing internal covariate shift[J]*. arXiv preprint arXiv:1502.03167, 2015.

[5] Laurent C, Pereyra G, Brakel P, et al.*Batch Normalized Recurrent Neural Networks[J]*. arXiv preprint arXiv:1510.01378, 2015.

[6] Amodei D, Anubhai R, Battenberg E, et al. *Deep Speech 2: End-to-End Speech Recognition in English and Mandarin[J]*. arXiv preprint arXiv:1512.02595, 2015.

[7] Pham V, Bluche T, Kermorvant C, et al. *Dropout improves recurrent neural networks for handwriting recognition[C]*. Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on. IEEE, 2014: 285-290.

[8] https://sisec.inria.fr/sisec-2015/2015-professionally-produced-music-recordings/