Improving Music Source Separation based on DNNs through Data Augmentation and Network Blending

SONY

Stefan Uhlich[1], Marcello Porcu[1], Franck Giron[1], Michael Enenkl[1], Thomas Kemp[1], Naoya Takahashi[2] and Yuki Mitsufuji[2]

[1] Sony European Technology Center (EuTEC), Stuttgart, Germany   [2] Sony Corporation, Audio Technology Development Department, Tokyo, Japan

Summary

- Separation of music into instruments ("bass", "drums", "other", "vocals")
- Two network architectures are described: feed-forward and recurrent
  - Each of them yields state-of-the art results on SiSEC DSD100
  - We blend both architectures to further improve performance
    - Linear combination of raw outputs and MWF post-processing
    - Gives the best results that have been reported so far on DSD100
- We study the effect of data augmentation for the recurrent architecture
  - Experiment shows that even simple architectures can overfit
  - Data augmentation during training avoids problem

Introduction

- Music Source Separation (MSS) = Separation of music into instrument tracks
  - Received increasing attention over the last years
  - Many applications require MSS: Karaoke, Upmixing, ...
- Popular MSS contest: Signal Separation Evaluation Campaign (SiSEC)
  - Regularly held source separation contest
  - Most popular separation subtask: Professionally-mixed music
    - Goal: Separate music into "bass", "drums", "other" and "vocals"
    - Train/test dataset consists of 50 songs each
- Contribution of this paper is three-fold
  - Description of our submissions to SiSEC 2016 contest
    - Results for feed-forward approach [1] with MWF post-processing
    - Results for new recurrent network structure (bi-directional LSTM)
  - Proposal of using data augmentation for training
    - Avoids overfitting to training data
    - Esp. important for recurrent nets as they only use DSD100 Dev
  - Blending of two networks before MWF post-processing
    - Considerably improves performance
    - Gives best results on DSD100 Test that have so far been reported

MSS using Deep Neural Networks

- Signal model for music source separation

$$\mathbf{x}(n) = \mathbf{s}_\mathsf{B}(n) + \mathbf{s}_\mathsf{D}(n) + \mathbf{s}_\mathsf{O}(n) + \mathbf{s}_\mathsf{V}(n) = \sum_{i \in \mathcal{I}} \mathbf{s}_i(n)$$

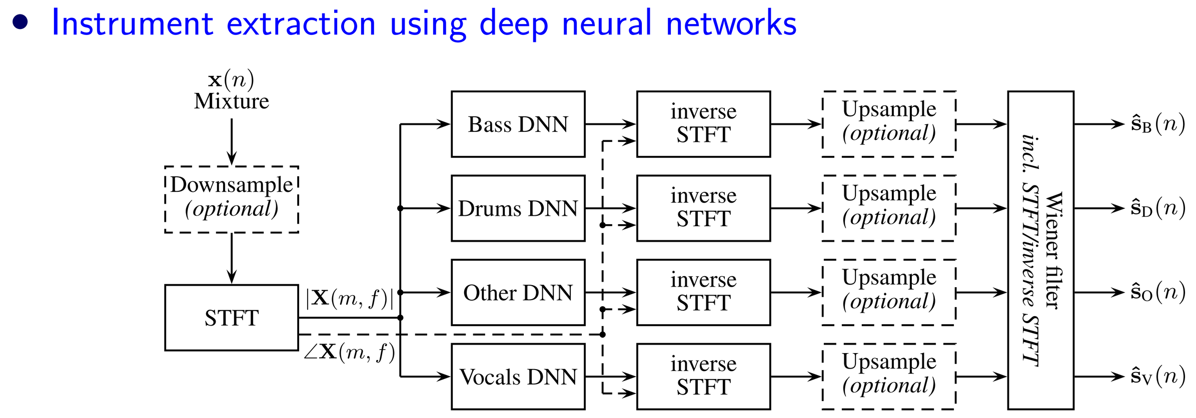
- Notation

$\mathbf{x}(n)$, $\mathbf{s}_i(n)$, $\hat{\mathbf{s}}_i(n) \in \mathbb{R}^2$ ... Stereo mixture/sources/source estimates in time domain with $\mathcal{I} := \{\mathsf{B}, \mathsf{D}, \mathsf{O}, \mathsf{V}\}$ denoting *bass*, *drums*, *other* and *vocals*

$\mathbf{X}(m,f)$, $\mathbf{S}_i(m,f)$, $\hat{\mathbf{S}}_i(m,f) \in \mathbb{C}^2$ ... Stereo mixture/sources/source estimates in STFT domain

General DNN Approach

- Instrument extraction using deep neural networks



- Multi-channel Wiener filter
  - Important post-processing step that improves performance
  - Reduces flanging effects that can appear for single-channel WF
  - Assumed signal model [2]

$$\mathbf{X}(m,f) = \mathbf{S}_i(m,f) + \mathbf{Z}_i(m,f)$$

with $i \in \mathcal{I}$, $\mathbf{Z}_i(m,f) = \sum_{j \in \mathcal{I} \setminus i} \mathbf{S}_j(m,f)$ and $\mathbf{S}_i(m,f) \sim \mathcal{CN}(\mathbf{0}, v_i(m,f)\mathbf{R}_i(f))$ where $v_i(m,f)$ is the power-spectral density (PSD) and $\mathbf{R}_i(f)$ the time-invariant spatial covariance matrix.

  - Hence, we assume a time-invariant, convolutive mixture
  - Reasonable for majority of music mixtures
- MMSE estimator for $\mathbf{S}_i(m,f)$ from $\mathbf{X}(m,f)$ is

$$\hat{\mathbf{S}}_i(m,f) = v_i(m,f)\mathbf{R}_i(f)\left(\sum_{j \in \mathcal{I}} v_j(m,f)\mathbf{R}_j(f)\right)^{-1}\mathbf{X}(m,f)$$

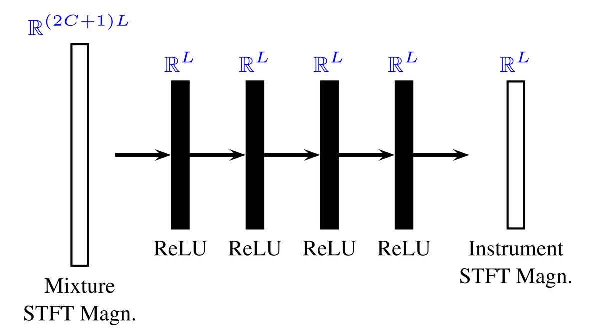
- PSDs and spatial covariance matrices are estimated from M consecutive frames as [2, 3]

$$\hat{v}_i(m,f) = \frac{1}{2}\|\hat{\mathbf{S}}_i(m,f)\|^2, \quad \hat{\mathbf{R}}_i(f) = \frac{\sum_{m=1}^{M} \hat{\mathbf{S}}_i(m,f)\hat{\mathbf{S}}_i(m,f)^H}{\sum_{m=1}^{M} \hat{v}_i(m,f)}$$

  - Weighted version of classical ML estimator
  - More weight put on TF bins for which we expect better SNR

Feed-Forward Networks (FNN)

- First approach: feed-forward architecture [1]



- FNN-1: previous submission to SiSEC 2015
  - Structure: ReLU network with $K = 3$ layers
  - Training material
    - $P = 2 \cdot 10^6$ training samples which are randomly generated
    - Short instrument loops which are independent of DSD100
  - Input: FFT size is 1024, $C = 3$ non-overlapping context frames
  - Post-processing: single-channel Wiener filter
- FNN-2: newly trained network with following changes regarding FNN-2
  - Structure: ReLU network with $K = 4$ layers
  - Training material: $P = 1.2 \cdot 10^7$ samples where we additionally use non-bleeding stems from MedleyDB and stems from SiSEC Dev
  - Input: $C = 8$ overlapping context frames with PCA pre-processing
- Comparison of the two sets of DNNs
  - Lower baseline: $\hat{s}_i(n) = \frac{1}{4}x(n)$
  - Upper baseline: Ideal ratio mask
  - Estimate of accompaniment given by $\hat{\mathbf{s}}_\mathsf{A}(n) = \mathbf{x}(n) - \hat{\mathbf{s}}_\mathsf{V}(n)$

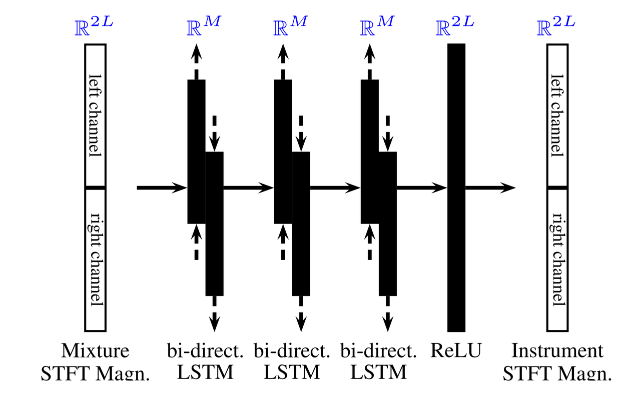
Network	SDR in dB					Comments
	Bass	Drums	Other	Vocals	Acco.	
BL	0.72	0.95	1.43	0.35	6.82	lower baseline: mix as separations
BU	6.26	7.96	7.76	10.16	16.38	upper baseline: ideal ratio mask
FNN-1	2.22	3.08	2.48	3.63	10.19	*UHL1* from SiSEC 2015
FNN-2	2.54	3.75	2.92	4.47	11.12	
BLSTM-1	2.30	3.71	2.98	3.69	10.33	one BLSTM layer
BLSTM-2	2.77	3.78	3.44	4.91	11.35	two BLSTM layers
BLSTM-3	2.89	4.00	3.24	4.86	11.26	three BLSTM layers

Table 1: *FNN and BLSTM networks on Test part of DSD100*

  - Observation: FNN-2 on average 0.6 dB better in SDR (using SiSEC Dev: $+0.4$ dB, using multi-channel WF: $+0.2$ dB)

Bidirectional LSTM Networks

- Second approach: Recurrent architecture with bidirect. LSTM layers
  - Better incorporation of context information than supervectors



- We trained three architectures which differ in #BLSTM layers
  - Each BLSTM layer consists of 250 forward/backward LSTM cells
  - Input: stereo magnitude frames (frame size: 1024, overlap: 50%)
  - Post-processing: multi-channel Wiener filter
- Results: see Table 1

Data Augmentation during Training

- Data augmentation is known to improve performance of DNNs
- We use the following data augmentation techniques on the fly when we create a mini-batch (mini-batch size: 10, sequence length: 500):
  - random swapping left/right channel for each instrument,
  - random scaling with uniform amplitudes from $[0.25, 1.25]$,
  - random chunking into sequences for each instrument, and,
  - random mixing of instruments from different songs.
- In order to prove the effectiveness of the data augmentation, we also trained BLSTM-1 for the extraction of vocals without it

Instr.	Network	SDR in dB (Raw outputs)		
		Dev	Test [All]	Test [New artists]
Vocals	BL	0.91	0.35	0.77
	BLSTM-1 w/o data augm.	7.13	3.37	3.19
	BLSTM-1 with data augm.	6.19	3.59	3.79
Accomp.	BL	6.57	6.82	6.49
	BLSTM-1 w/o data augm.	13.33	9.71	9.53
	BLSTM-1 with data augm.	12.23	9.93	9.64

Table 2: *Effect of data augmentation for BLSTM-1*

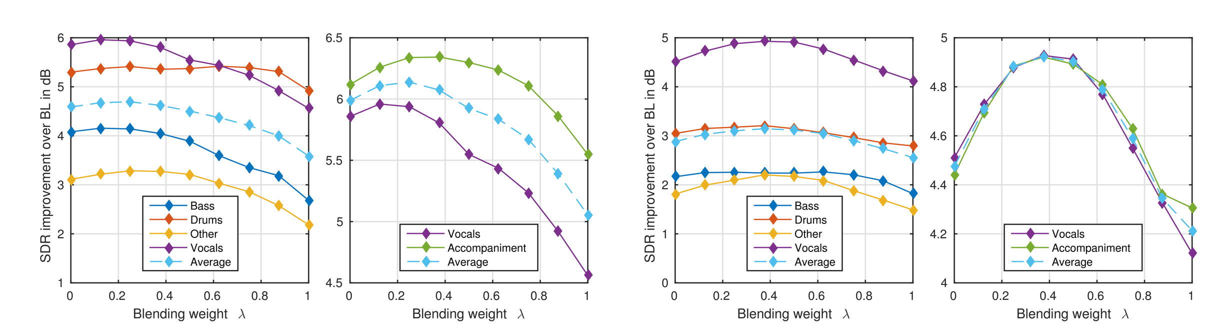
  - Vocals and accompaniment gain through data augmentation
    - on average 0.2 dB if we consider all Test songs
    - on average 0.35 dB if we only consider the subset of Test songs where there is not a song of the same artist in the Dev part

Blending of Networks

- Further improvement: Blending ($+0.2$ dB compared to BLSTM-3)
  - Step 1: Linear combination of raw DNN outputs

$$\hat{\mathbf{s}}_{i,\mathsf{BLEND}}(n) = \lambda\hat{\mathbf{s}}_{i,\mathsf{FNN}}(n) + (1-\lambda)\hat{\mathbf{s}}_{i,\mathsf{BLSTM}}(n)$$

  - Step 2: multi-channel Wiener-filter post-processing



(a) Dev part of DSD100   (b) Test part of DSD100

Figure 3: *Effect of blending ($\lambda = 0$: BLSTM-3, $\lambda = 1$: FNN-2)*

- Our blending scheme is an extension of learned temporal fusion [4]
  - Instead of linearly combining the systems after the MWF, we blend the raw outputs of each DNN and perform afterwards a MWF post-processing
  - Final MWF helps to reduce the interference and achieves better results
  - Comparing the two schemes, we can observe that our fusion is on average 0.1 dB better for SDR and 0.3 dB for th SIR than the *learned temporal fusion* scheme proposed in [4].

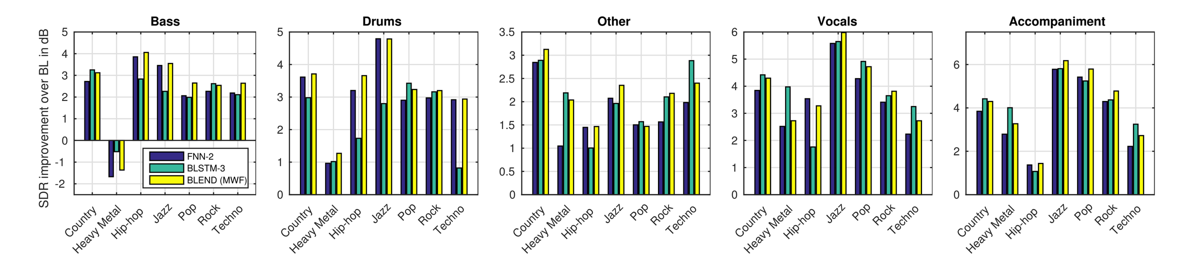


Figure 4: *SDR improvement in dB for different music genres (Test part of DSD100)*

Comparison to Other Approaches

- Comparison of our networks to other methods from literature (NMF, DNN)

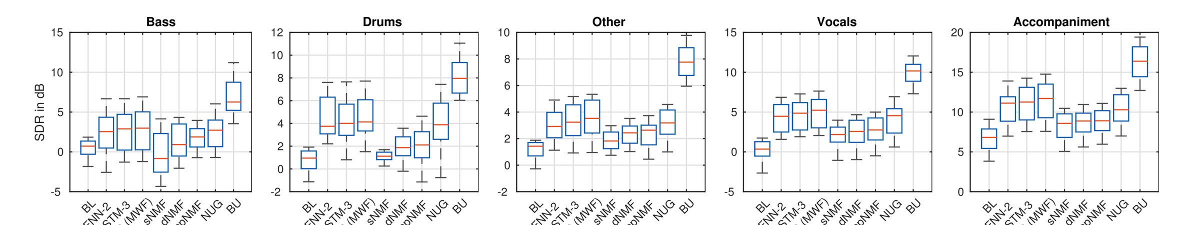


Figure 5: *Comparison of different music source separation algorithms (Test part of DSD100)*

	Approach	SDR in dB					Comments
		Bass	Drums	Other	Vocals	Acco.	
Single-channel methods	BLEND (SWF)	2.76	3.93	3.37	5.13	11.53	$\lambda = 0.25$
	sNMF [5, 6]	-0.84	1.12	1.82	2.17	8.58	$Q = 25$
	dNMF [7]	0.91	1.87	2.43	2.56	8.88	$Q = 25$
	DeepNMF [8]	1.88	2.11	2.64	2.75	8.90	$Q = 25$
Multi-channel methods	BLEND (MWF)	2.98	4.13	3.52	5.23	11.70	$\lambda = 0.25$
	NUG [2]	2.72	3.89	3.18	4.55	10.29	

Table 3: *Comparison on Test part of DSD100*

- Comparison of BLEND (MWF) to FNN-1
  - We could gain 1.1 dB SDR since SiSEC 2015 competition

Separations of all methods that particpated in SiSEC MUS are available
http://sisec17.audiolabs-erlangen.de

References

[1] S. Uhlich, F. Giron, and Y. Mitsufuji, "Deep neural network based instrument extraction from music," in *Proc. ICASSP*, 2015, pp. 2135–2139.

[2] A. A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel music separation with deep neural networks," in *Proc. EUSIPCO*, 2016.

[3] S. Sivasankaran, A. A. Nugraha, E. Vincent, J. A. Morales-Cordovilla, S. Dalmia, I. Illina, and A. Liutkus, "Robust ASR using neural network based speech enhancement and feature simulation," in *Proc. ASRU*, 2015, pp. 482–489.

[4] X. Jaureguiberry, G. Richard, P. Leveau, R. Hennequin, and E. Vincent, "Introducing a simple fusion framework for audio source separation," in *Proc. MLSP*, 2013, pp. 1–6.

[5] J. Eggert and E. Korner, "Sparse coding and NMF," in *Proc. Neural Networks*, 2004, vol. 4, pp. 2529–2533.

[6] C. Févotte, N. Bertin, and J.-L. Durrieu, "Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis," *Neural computation*, vol. 21, no. 3, pp. 793–830, 2009.

[7] F. Weninger, J. Le Roux, J. R. Hershey, and S. Watanabe, "Discriminative NMF and its application to single-channel source separation," in *Proc. Interspeech*, 2014, pp. 865–869.

[8] J. Le Roux, J. R. Hershey, and F. Weninger, "Deep NMF for speech separation," in *Proc. ICASSP*, 2015, pp. 66–70.

ICASSP 2017, March 5 - 9, New Orleans