MULTI-SCENARIO DEEP LEARNING FOR MULTI-SPEAKER SOURCE SEPARATION

Jeroen Zegers and Hugo Van hamme

PSI, ESAT, KULeuven, Belgium

KU LEUVEN

1. Introduction

Multi-speaker Source Separation (MSSS)

- Cocktail party problem
- Speaker independent (unknown sources)
- MSSS using Deep Learning

Research Question

- *How to combine data from different scenario's?*
- Scenario's (tasks): mixtures with different number of speakers
- Advantages:
  - Data optimally used
  - Single model needed

Relevance

- Preprocessing to conversational ASR, meeting transcriptions, ...
- Increased speech intelligibility

2. Permutation Invariant Training (PIT)

Source Separation

Mixture: $\mathbf{Y} = \sum_{s=1}^{S} \mathbf{X}_s$

SS estimate: $\hat{\mathbf{X}}_s = \hat{\mathbf{M}}_s \odot \mathbf{Y}$

Permutation Problem for Single Class SS

- Sources from single class. How to distinguish sources?
- Permutation independent loss function [1]:

$$\mathcal{L}_\theta = \min_{p \in \mathcal{P}_S} \sum_{s=1}^{S} \sum_{t,f} || |\hat{\mathbf{X}}_{\theta,s,tf}| - |\mathbf{X}_{p_s,tf}| ||_F^2$$

- Number of output nodes dependent on number of speakers

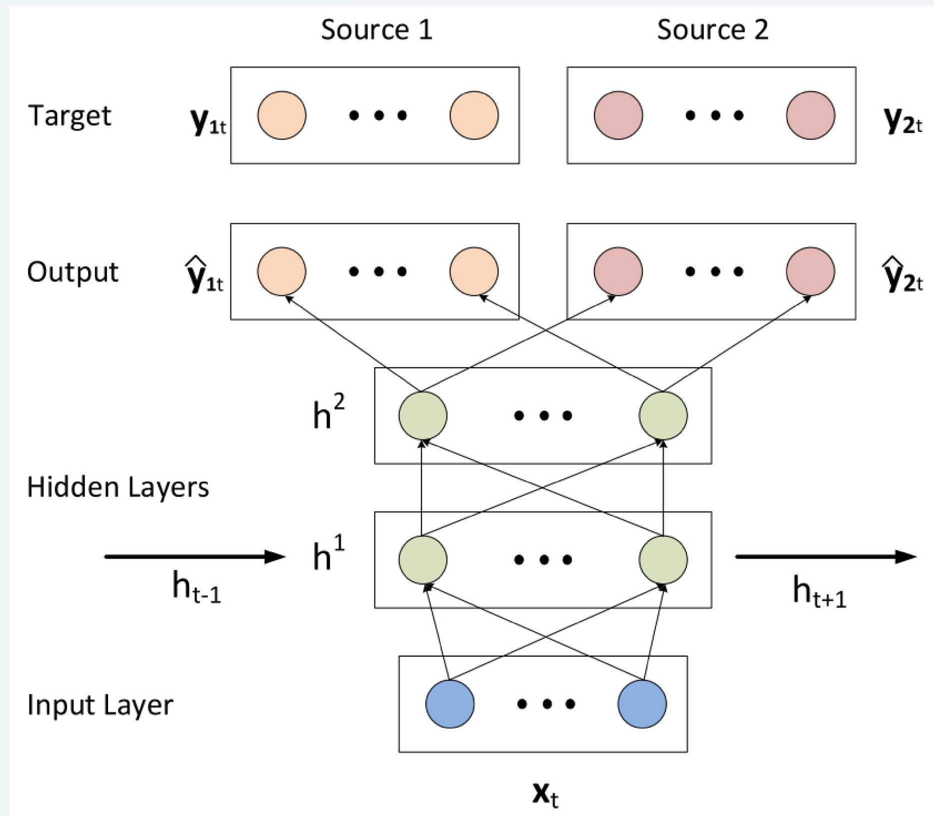

Figure 1: PIT architecture

[1] "Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks", *Kolbæk, Morten et al. 2017*

3. Deep Clustering (DC)

Training

- Map time-frequency bin to embedding. Embeddings closer per speaker

$$\mathbf{v}_{tf} = f_\theta(\mathbf{Y})$$

- Labels: binary indicator of most active speaker in each bin

$$w_{tf,s} = \begin{cases} 1, & \text{if } x_{tf,s} > x_{tf,s'} \\ 0, & \text{otherwise} \end{cases}.$$

- Loss function is permutation independent:

$$\mathcal{L}_\theta = ||\mathbf{V}_\theta \mathbf{V}_\theta^T - \mathbf{W}\mathbf{W}^T||_F$$
$$= \sum_{tf_1, tf_2} (<\mathbf{v}_{tf_1}, \mathbf{v}_{tf_2}> - <\mathbf{w}_{tf_1}, \mathbf{w}_{tf_2}>)$$

Testing

- Cluster embeddings using K-means for C (number of speakers) clusters to form a binary mask (BM) per speaker
- Number of output nodes independent on number of speakers (not for K-means)

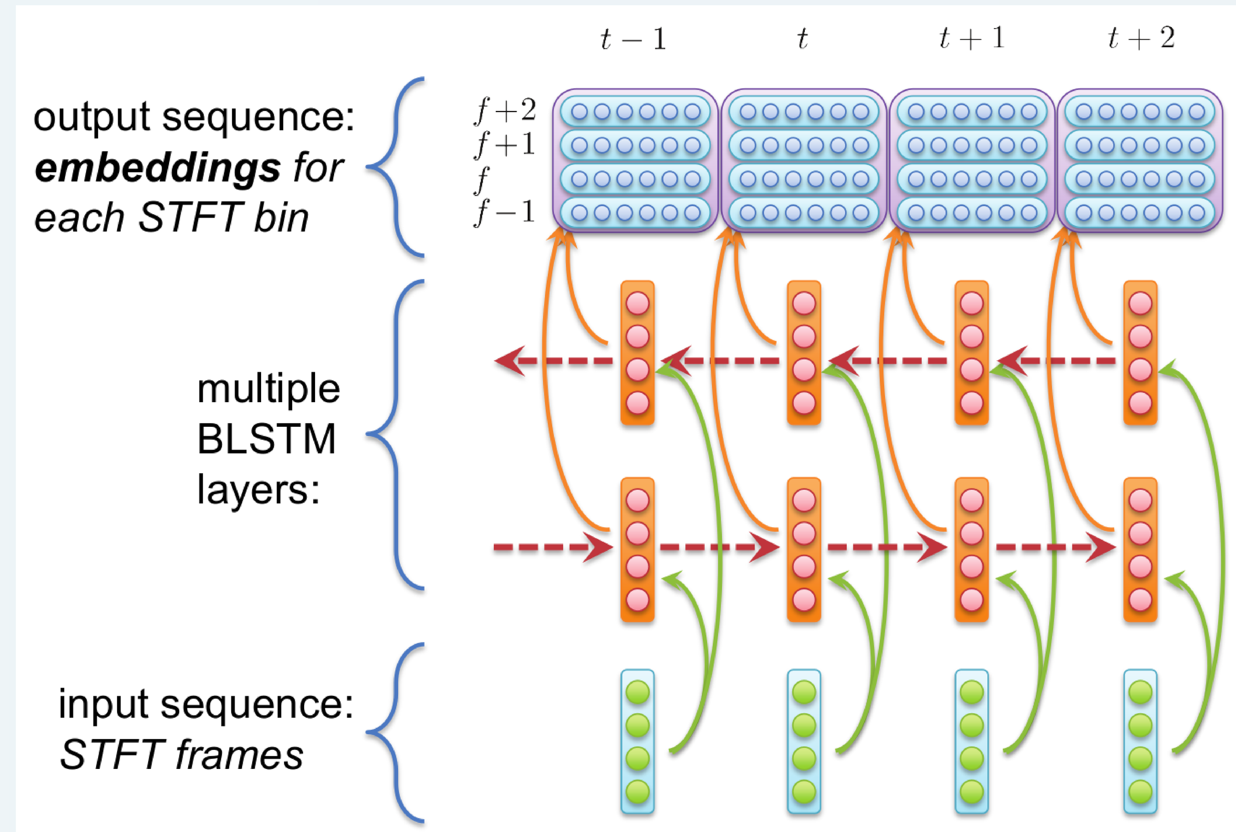

Figure 2: Deep clustering architecture [2]

[2] "Deep clustering: Discriminative embeddings for segmentation and separation", *Hershey, John R et al. 2016*

4. Joint Learning

- Parameter update

$$\Delta\theta_i = g\left(\frac{\partial \mathcal{L}}{\partial \theta_i}\right)$$

$g()$ is stochastic gradient descent, Adam, ...

- Parameter update for Multi-scenario learning:

$$\mathcal{L}_m = \sum_{j=1}^{J} \alpha_j \mathcal{L}_j$$

- Alternative approach:

$$\Delta_m\theta_i = \sum_j^J \Delta_j\theta_i = \sum_j^J g\left(\frac{\alpha_j \partial \mathcal{L}_j}{\partial \theta_i}\right)$$

For Adam $g(\alpha\partial\mathcal{L}/\partial\theta_i) = g(\partial\mathcal{L}/\partial\theta_i)$:

$$\Delta_m\theta_i = \sum_j^J g\left(\frac{\partial \mathcal{L}_j}{\partial \theta_i}\right)$$

5. Experiments

Database and setup

- Artificial mixtures from Wall Street Journal 0 (WSJ0) database
- Bidirectional LSTM (2 layers, 600 units)
- MSSS performance in Signal to Distortion Ratio (SDR) improvements over original mixture
- TensorFlow

Single scenario Experiments

- Train on 2 spk and test on 3 spk: $-4.14dB$
- Train on 3 spk and test on 2 spk: $-0.51dB$

Multi scenario Experiments

- Fully shared model: slight drop
- Separate output layer: slight increase
- Slightly worse if number of training mixtures not increased

Conclusion

- Data from different scenario's are useful
- Only need a single model for multiple scenario's

Results

algorithm	train set	test set 2spk	3spk	2+3spk
DC	2spk	**8.59**	1.95	5.27
	3spk	8.08	**6.09**	**7.20**
	2+3spk	8.20	5.29	6.66
	2+3spk half	8.08	5.12	6.60
DC out sep	2+3spk	**9.38**	**6.38**	**7.88**
	2+3spk half	8.73	5.77	7.25
PIT	2spk	8.21	-	-
	3spk	-	6.25	-
	2+3spk	**8.48**	**6.50**	**7.67**
	2+3spk half	7.97	6.08	7.03

Table: SDR improvement of reconstructed signals in dB. DC=Deep Clustering. PIT=Permutation Invariant Training.