Deep attractor networks for speaker re-identification and blind source separation

Lukas Drude, Thilo von Neumann, Reinhold Haeb-Umbach

Department of Communications Engineering – Paderborn University
Prof. Dr.-Ing. Reinhold Haeb-Umbach
2018-04-17
Table of contents

Introduction/ problem statement

Analysis of latent space (DC, DAN)

Solution: Identification embeddings

Evaluation
Schematic overview: DC/ DAN

- **Deep Clustering (DC)**
  - No assumption about the speaker at test time
  - Encoder network generates embedding vectors
  - Decoder just applies binary mask to observation

- **Deep Attractor network (DAN)**
  - Different loss function allows end-to-end training
  - Decoder calculates soft mask first
  - Developed for short mixtures.

Properties of the embeddings?
- Identify speakers?
Deep Clustering (DC) [Hershey 2016]:
- No assumption about the speaker at test time
- Encoder network generates embedding vectors
- Decoder just applies binary mask to observation

Deep Attractor network (DAN) [Chen 2017]:
- Different loss function allows end-to-end training
- Decoder calculates soft mask first
Schematic overview: DC/ DAN

- Deep Clustering (DC) [Hershey 2016]:
  - No assumption about the speaker at test time
  - Encoder network generates embedding vectors
  - Decoder just applies binary mask to observation

- Deep Attractor network (DAN) [Chen 2017]:
  - Different loss function allows end-to-end training
  - Decoder calculates soft mask first

Properties of the embeddings? Identify speakers?

Developed for short mixtures.
Tasks

Block permutation problem (tracing)

Block $n$  $\rightarrow$  Block $n + 1$
Tasks

Block permutation problem (tracing)

Block $n$ \rightarrow Block $n + 1$

Re-identification problem

Blue speaker \rightarrow Bag of $K$ separation results
Possible approaches

• Use i-vectors?
  → See results.

• Multichannel/ spatial cues?
  → AASP-P11.3:
    Drude et al., Dual Frequency- and Block-Permutation Alignment [...] Friday 13:30 – 15:30
Deep Clustering

- Minimize difference between estimated and true affinity matrices:
  - Embedding vectors of same speaker co-linear
  - Embedding vectors of different speakers orthogonal

Encoder Network → Affinity Loss → loss $\ell_{DC}$

$y_{tf}$ → $e_{tf}$
Deep Clustering – Centroids

- Each dot is an embedding **centroid** for each speaker
- Oracle mask used to visualize centroids
Deep Clustering – Centroids

- Each dot is an embedding centroid for each speaker
- Oracle mask used to visualize centroids

![Diagram showing PCA components and centroids labeled A, B, and C.](Diagram)
Minimize reconstruction loss (MSE)

Intuition:
- Embedding vectors of same speaker in same direction
- Embedding vectors of different speakers in opposite direction
Deep Attractor Network – Centroids/Attractors

- Each dot is an embedding **centroid** for each speaker
- Oracle mask used to visualize centroids

![PCA Component 1 vs PCA Component 2](image)

L. Drude, T. von Neumann, R. Haeb-Umbach
Deep attractor networks for speaker re-identification and blind source separation
Deep Attractor Network – Centroids/Attractors

- Each dot is an embedding **centroid** for each speaker
- Oracle mask used to visualize centroids

![PCA Component vs PCA Component 2](image)

L. Drude, T. von Neumann, R. Haeb-Umbach
Deep attractor networks for speaker re-identification and blind source separation
Deep Attractor Network – Centroids/Attractors

- Each dot is an embedding **centroid** for each speaker
- Oracle mask used to visualize centroids
Solution: Identification loss

- Upper branch: Vanilla DAN
Solution: Identification loss

- Upper branch: Vanilla DAN
- Lower branch: Identification network + loss
  - Loss during training
  - Just use corresponding centroid at test time
- Multi-task learning: \( \ell_{\text{total}} = \ell_{\text{MSE}} + \alpha \ell_{\text{CE}} \)

\[
\begin{align*}
\text{Encoder Network} & \quad \rightarrow \quad e_{tf} \\
\text{Mean} & \quad \rightarrow \quad \mu_k \\
\text{Identification Network} & \quad \rightarrow \quad i_k \\
\text{MSE} & \quad \rightarrow \quad x_{ktf} \quad \text{(reconstruction loss)} \\
\end{align*}
\]
Solution: Identification loss

- Upper branch: Vanilla DAN
- Lower branch: Identification network + loss
  - Loss during training
  - Just use corresponding centroid at test time
- Multi-task learning: $\ell_{\text{total}} = \ell_{\text{MSE}} + \alpha \ell_{\text{CE}}$
Solution: Identification loss

- Location of **identification attractors** tend to form clusters
Source separation performance

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>SDR/dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAN</td>
<td>9.4</td>
</tr>
<tr>
<td>DAN + ID loss</td>
<td></td>
</tr>
<tr>
<td>0.001</td>
<td>10.1</td>
</tr>
<tr>
<td>0.01</td>
<td>9.9</td>
</tr>
<tr>
<td>0.1</td>
<td>9.9</td>
</tr>
<tr>
<td>1</td>
<td>9.7</td>
</tr>
<tr>
<td>10</td>
<td>8.9</td>
</tr>
<tr>
<td>DAN + ID emb.</td>
<td></td>
</tr>
<tr>
<td>0.001</td>
<td>9.9</td>
</tr>
<tr>
<td>0.01</td>
<td>9.8</td>
</tr>
<tr>
<td>0.1</td>
<td>10.0</td>
</tr>
<tr>
<td>1</td>
<td>10.1</td>
</tr>
<tr>
<td>10</td>
<td>9.2</td>
</tr>
</tbody>
</table>
### Permutation/re-identification performance

<table>
<thead>
<tr>
<th>Error Rate / %:</th>
<th>$\alpha$</th>
<th>Permutation</th>
<th>Identification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chance level</td>
<td>50.0</td>
<td>50.0</td>
<td></td>
</tr>
<tr>
<td>i-vector with VAD</td>
<td>8.0</td>
<td>9.7</td>
<td></td>
</tr>
<tr>
<td>DC</td>
<td>7.3</td>
<td>33.4</td>
<td></td>
</tr>
<tr>
<td>DAN</td>
<td>5.8</td>
<td>31.5</td>
<td></td>
</tr>
<tr>
<td>DAN + ID loss</td>
<td>0.001</td>
<td>6.7</td>
<td>32.7</td>
</tr>
<tr>
<td></td>
<td>0.01</td>
<td>6.0</td>
<td>31.1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>5.0</td>
<td>20.1</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>4.0</td>
<td>9.3</td>
</tr>
<tr>
<td>DAN + ID emb.</td>
<td>0.001</td>
<td>4.7</td>
<td>9.9</td>
</tr>
<tr>
<td></td>
<td>0.01</td>
<td>3.7</td>
<td>7.7</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>4.2</td>
<td>8.5</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td><strong>3.1</strong></td>
<td><strong>6.4</strong></td>
</tr>
</tbody>
</table>
Summary

- Embedding topology only valid for one mixture
  - Limitations in changing mixing conditions
  - Limitations for re-identification

- Extract speaker information with same encoder network
  - Multi-task learning helps both objectives

- Ways for speaker tracing/identification...
  - i-vectors
  - Multichannel/spatial cues (Drude et al., Friday, AASP-P11.3)
  - Embedding network provides ID embeddings