Deep Learning Acoustic Model in Microsoft Cortana Voice Assistant

Jinyu Li
AI & Research, Microsoft
Speech Recognition Products
Selected Technologies behind Microsoft Cortana

- Reduce **runtime** cost without accuracy loss
- Adapt to speakers with **low** footprints
- Time-frequency **invariance** modeling
- Enable languages with **limited** training data
- Reduce accuracy **gap** between large and small deep networks
- **New** domain adaptation
- Multi-talker **separation**
Reduce Runtime Cost without Accuracy Loss

[Xue13, Miao16]
Motivation

• The runtime cost of DNN is much larger than that of GMM, which has been fully optimized in product deployment. We need to reduce the runtime cost of DNN in order to ship it.
Solution

• The runtime cost of DNN is much larger than that of GMM, which has been fully optimized in product deployment. We need to reduce the runtime cost of DNN in order to ship it.

• We proposed SVD-based model restructuring to compress the DNN models without accuracy loss.
Singular Value Decomposition (SVD)

\[ A_{m \times n} = U_{m \times n} \Sigma_{n \times n} V_{n \times n}^T = \begin{bmatrix} u_{11} & \cdots & u_{1n} \\ \vdots & \ddots & \vdots \\ u_{m1} & \cdots & u_{mn} \end{bmatrix} \begin{bmatrix} \epsilon_{11} & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & \epsilon_{kk} & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & \epsilon_{nn} \end{bmatrix} \begin{bmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \vdots \\ v_{n1} & \cdots & v_{nn} \end{bmatrix} \]
SVD Approximation

\[
\begin{bmatrix}
a_{11} & \cdots & a_{1n} \\
a_{m1} & \cdots & a_{mn}
\end{bmatrix}
= \begin{bmatrix}
u_{11} & \cdots & u_{1n} \\
u_{m1} & \cdots & u_{mn}
\end{bmatrix}
\begin{bmatrix}
\epsilon_{11} & \cdots & 0 & \cdots & 0 \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
0 & \cdots & \epsilon_{kk} & \cdots & 0 \\
0 & \cdots & 0 & \cdots & \epsilon_{nm}
\end{bmatrix}
\begin{bmatrix}
v_{11} & \cdots & v_{1n} \\
v_{n1} & \cdots & v_{nn}
\end{bmatrix}
\]

\[
\approx \begin{bmatrix}
u_{11} & \cdots & u_{1k} \\
u_{m1} & \cdots & u_{mk}
\end{bmatrix}
\begin{bmatrix}
\epsilon_{11} & \cdots & 0 & \cdots & 0 \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
0 & \cdots & \epsilon_{kk} & \cdots & 0 \\
0 & \cdots & 0 & \cdots & \epsilon_{nk}
\end{bmatrix}
\begin{bmatrix}
v_{11} & \cdots & v_{1n} \\
v_{k1} & \cdots & v_{kn}
\end{bmatrix}
\]

\[
= \begin{bmatrix}
u_{11} & \cdots & u_{1k} \\
u_{m1} & \cdots & u_{mk}
\end{bmatrix}
\begin{bmatrix}
\epsilon_{11} & \cdots & w_{11} & \cdots & w_{1n} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
0 & \cdots & \epsilon_{kk} & \cdots & w_{kk} \\
0 & \cdots & 0 & \cdots & w_{kn}
\end{bmatrix}
\]

- Number of parameters: \(mn \rightarrow mk+nk\).
- Runtime cost: \(O(mn) \rightarrow O(mk+nk)\).
- E.g., \(m=2048, n=2048, k=192\). 80% runtime cost reduction without accuracy loss.
SVD-Based Model Restructuring
SVD-Based Model Restructuring
Directly training from the low-rank structure without doing SVD costs 4% relative WER increase.
Decoding with Frame Skipping

DNN Model

\[ x_{t-1} \quad x_t \quad x_{t+1} \]

LSTM Model

\[ x_{t-1} \quad x_t \quad x_{t+1} \]
LSTM Training with Frame Skipping

*Split training utterances* through frame skipping

- When skipping 1 frame, *odd and even frames* are picked as separate utterances

\[
\begin{array}{cccc}
X_1 & X_2 & X_3 & X_4 \\
\end{array}
\begin{array}{cccc}
X_5 & X_6 \\
\end{array}
\begin{array}{cccc}
X_1 & X_3 & X_5 \\
\end{array}
\begin{array}{cccc}
X_2 & X_4 & X_6 \\
\end{array}
\]

- Frame labels are selected accordingly
Adapt to Speakers with Low Footprints

[Xue14]
Motivation

- Speaker personalization with a deep model creates a storage size issue: It is not practical to store an entire deep models for each individual speaker during deployment.
Solution

• Speaker personalization with a DNN model creates a storage size issue: It is not practical to store an entire DNN model for each individual speaker during deployment.

• We proposed low-footprint DNN personalization method based on SVD structure.
SVD Personalization
SVD Personalization
Adaptation with 100 Utterances

Adapting with 100 utterances

- **FULL-SIZE DNN**: 30
- **SVD DNN**: 7.4
- **STANDARD ADAPTATION**: 18.64
- **SVD ADAPTATION**: 20.86

Relative WER reduction

Number of parameters (M)
Time-Frequency Invariance Modeling

[Li15, Li16]
How DNN and (LSTM-)RNN Process an Utterance

• Independence between LFBs
How DNN and (LSTM-)RNN Process an Utterance

• No impact when two LFBs are switched.
Human Read Spectrum by Using the Correlation across Time and Frequency

• Big impact when two LFBs are switched.
Frequency-LSTM
TF-LSTM Results

Models: trained from the 375hr Cortana task

Test set: Cortana

<table>
<thead>
<tr>
<th>Model</th>
<th>WER (%)</th>
<th>Number of parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-layer T-LSTM</td>
<td>15.35</td>
<td>19.8 M</td>
</tr>
<tr>
<td>TF-LSTM + 3-layer T-LSTM</td>
<td>15.09</td>
<td>17.0 M</td>
</tr>
<tr>
<td>TF-LSTM + 4-layer T-LSTM</td>
<td>14.83</td>
<td>21.6 M</td>
</tr>
</tbody>
</table>
Invariance Properties

Models: trained from the 375hr Cortana task

Test set: Aurora 4

<table>
<thead>
<tr>
<th>Model</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-layer T-LSTM</td>
<td>6.37</td>
<td>14.25</td>
<td>9.14</td>
<td>23.90</td>
<td>17.46</td>
</tr>
<tr>
<td>TF-LSTM + 4-layer T-LSTM</td>
<td>5.45</td>
<td>12.07</td>
<td>8.07</td>
<td>20.69</td>
<td>15.01</td>
</tr>
</tbody>
</table>

14.2% WERR
Enable Languages with Limited Training Data

[Huang13]
Motivation

• Develop a new language in new scenario with small amount of training data.
Solution

• Develop a new language in new scenario with small amount of training data.

• Leverage the resource-rich languages to develop high-quality ASR for resource-limited languages.
Shared Hidden Layer Multi-Lingual DNN

Language 1 senones
Language 2 senones
Language 3 senones
Language 4 senones

Many Hidden Layers

Shared Feature Transformation

Input Layer:
A window of acoustic feature frames

Lang 1
Lang 2
Lang 3
Lang 4
Training or Testing Samples
Adapting to New Language

Input Layer:
A window of acoustic feature frames

Shared
Feature Transformation

Many Hidden Layers

Output Layer

New language senones

New Language

Training or Testing Samples
DNN data reuse: 10-20% WER reduction with data from non-native languages
(WER vs. hours of data)

Target language: zh-CN
Non-native source languages: FRA: 138 hours, DEU: 195 hours, ESP: 63 hours, and ITA: 93 hours of speech.
Reduce Accuracy Gap between Large and Small Deep Networks

[Li14]
To Deploy DNN on Server

- SVD matrices are used to reduce the number of DNN parameters and CPU cost.
- Quantization for SSE evaluation is used for single instruction multiple data processing.
- Frame skipping is used to remove the evaluation of some frames.
To Deploy DNN on Device

• Even with the technologies mentioned above, the large computational cost is still very challenging due to the limited processing power of devices.

• A common way to fit CD-DNN-HMM on devices is to reduce the DNN model size by
  • reducing the number of nodes in hidden layers
  • reducing the number of targets in the output layer
Significant Accuracy Loss when DNN Size Is Significantly Reduced

- Better accuracy is obtained if we use the output of large-size DNN for acoustic likelihood evaluation.

- The output of small-size DNN is away from that of large-size DNN, resulting in worse recognition accuracy.

- The problem is solved if the small-size DNN can generate similar output as the large-size DNN.
Teacher-Student Learning

- Minimize the KL divergence between the output distribution of the student DNN and teacher DNN with large amount of un-transcribed data.
# Learning with Soft Targets

## Teacher-Student Learning [1]

\[
- \sum_f \sum_i P_T(s_i|x_{src,f}) \log P_S(s_i|x_{tgt,f})
\]

Pure soft target learning

**Can use all available untranscribed data**

## Knowledge Distillation [2]

\[
-(1 - \lambda) \sum_f \sum_i P_T(s_i|x_{src,f}) \log P_S(s_i|x_{tgt,f})
\]

\[
-\lambda \sum_f \log P_S(s_i|x_{tgt,f})
\]

Soft target regularized with hard label from transcription

**Limited to available transcribed data**

---


Production Setup

- 2 Million parameter for small-size DNN, compared to 30 Million parameters for teacher DNN.
- The footprint is further reduced to 0.5 million parameter when combining with SVD.
New Domain Adaptation with Parallel Data

[Li17]
Domain Adaptation

• The success of deep learning relies on a large amount of transcribed data
  • The training data is assumed to originate from the distribution as the test data
  • Performance degrades when exposed to test data from a new domain

• It is very expensive to transcribe large amounts of data for a new domain
  • Domain-adaptation approaches have been proposed to bootstrap the training of a new model using an existing well-trained model
    • Supervised adaptation: only limited transcribed data is available in new domain
    • Semi-supervised adaptation: Estimated hypotheses are typically unreliable in the new domain
    • Unsupervised adaptation: does not rely on transcription
How to Train a Good Target Model

- Good accuracy is obtained if we use the output of source-domain DNN with source data for acoustic likelihood evaluation.
- The output of target-domain DNN with target data is away from that of source-domain DNN with source data, resulting in worse recognition accuracy.
- The problem is solved if target-domain DNN with target data can generate similar output as the source-domain DNN with source data.
Teacher-Student Learning with Parallel Data

- The behavior of student DNN with target data should be similar to that of the teacher DNN with source data.

- Objective function: minimize the KL distance between the teacher and student distributions:

$$- \sum_f \sum_i P_T(s_i|x_{src,f}) \log P_S(s_i|x_{tgt,f})$$

- No transcriptions required.
# Application Scenarios

<table>
<thead>
<tr>
<th>Source domain</th>
<th>Target domain</th>
<th>How to simulate?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean speech</td>
<td>Noisy speech</td>
<td>Add noise</td>
</tr>
<tr>
<td>Close-talk speech</td>
<td>Far-field speech</td>
<td>Apply RIR, add noise</td>
</tr>
<tr>
<td>Adults</td>
<td>Children</td>
<td>Voice morphing</td>
</tr>
<tr>
<td>Original speech</td>
<td>Compressed speech</td>
<td>Apply codec</td>
</tr>
<tr>
<td>Wideband speech</td>
<td>Narrowband speech</td>
<td>Downsampling/filter</td>
</tr>
</tbody>
</table>
Experimental evaluation

• Baseline model: 4-layer LSTM trained with 375 hours of Cortana data (Microsoft’s digital assistant available on many platforms)

• Evaluated using 2 new domains
  • Noisy Cortana
  • CHiME-3

<table>
<thead>
<tr>
<th>Task</th>
<th>Test utterances</th>
<th>Parallel data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noisy Cortana task</td>
<td>Simulated noisy speech</td>
<td>clean – simulated noisy speech</td>
</tr>
<tr>
<td>CHiME-3 task</td>
<td>Real far-talk speech</td>
<td>close – far talk speech</td>
</tr>
</tbody>
</table>
## Noisy Cortana Task

<table>
<thead>
<tr>
<th>Train Teacher</th>
<th>Train Student</th>
<th>Noisy WER</th>
<th>Original WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>original 375h</td>
<td>none</td>
<td>18.80</td>
<td>15.62</td>
</tr>
<tr>
<td>noisy 375h</td>
<td>none</td>
<td>17.34</td>
<td>16.58</td>
</tr>
<tr>
<td>original 375h</td>
<td>original + noisy (375h)</td>
<td>16.66</td>
<td>15.32</td>
</tr>
</tbody>
</table>
## Noisy Cortana Task

<table>
<thead>
<tr>
<th>Train Teacher</th>
<th>Train Student</th>
<th>noisy WER</th>
<th>original WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>original 375h</td>
<td>none</td>
<td>18.80</td>
<td>15.62</td>
</tr>
<tr>
<td>noisy 375h</td>
<td>none</td>
<td>17.34</td>
<td>16.58</td>
</tr>
<tr>
<td>original 375h</td>
<td>original + noisy (375h)</td>
<td>16.66</td>
<td>15.32</td>
</tr>
<tr>
<td>original 375h</td>
<td>original + noisy (3400h)</td>
<td>16.11</td>
<td>15.17</td>
</tr>
</tbody>
</table>
## Noisy Cortana Task

<table>
<thead>
<tr>
<th>Train Teacher</th>
<th>Train Student</th>
<th>noisy WER</th>
<th>original WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>original 375h</td>
<td>none</td>
<td>18.80</td>
<td>15.62</td>
</tr>
<tr>
<td>noisy 375h</td>
<td>none</td>
<td>17.34</td>
<td>16.58</td>
</tr>
<tr>
<td>original 375h</td>
<td>original + noisy (375h)</td>
<td>16.66</td>
<td>15.32</td>
</tr>
<tr>
<td>original 375h</td>
<td>original + noisy (3400h)</td>
<td>16.11</td>
<td>15.17</td>
</tr>
</tbody>
</table>

Student network in the **target domain** is approaching performance of teacher network in the **source domain**.
How to Effectively Simulate Data

• Example: Assume we want to use 5X data
• Compare two approaches:
  • Simulate 5 different copies of the transcribed data
  • Simulate 1 copy of 5X larger untranscribed data
Space of Original Transcribed Data

source data
Simulate 5 Copies of the Transcribed Data
Space of Original Transcribed Data

source data
Space of 5x Untranscribed Data
Simulate 1 Copy of 5x Un-transcribed Data
Chime-3 Task

- Test data more severely mismatched to training data
  - Topic/content mismatched (personal assistant vs. WSJ)
  - Noises/conditions mismatched to adaptation data

<table>
<thead>
<tr>
<th>Train Teacher</th>
<th>Train Student</th>
<th>Chime-3 WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>original 375h</td>
<td>none</td>
<td>23.16</td>
</tr>
<tr>
<td>noisy 375h</td>
<td>none</td>
<td>24.51</td>
</tr>
<tr>
<td>original 375h</td>
<td>original + noisy (375h)</td>
<td>23.67</td>
</tr>
<tr>
<td>original 375h</td>
<td>original + noisy (3400h)</td>
<td><strong>19.89</strong></td>
</tr>
</tbody>
</table>

- Increasing the amount of parallel training data helps the student model more of the acoustic space
Chime-3 with Smaller Well-matched Parallel Corpus

- Matched real data significantly improves the performance of T/S learning

<table>
<thead>
<tr>
<th>The noisy data in the pair comes from</th>
<th>Real channel 5</th>
<th>Simulated channel 5</th>
<th>Other real channels</th>
<th>Simulated other channels</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>15.88</td>
</tr>
</tbody>
</table>
Chime-3 with Smaller Well-matched Parallel Corpus

• Matched simulated data also improves the performance of T/S learning

<table>
<thead>
<tr>
<th>The noisy data in the pair comes from</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real channel 5</td>
<td></td>
</tr>
<tr>
<td>Simulated channel 5</td>
<td></td>
</tr>
<tr>
<td>Other real channels</td>
<td></td>
</tr>
<tr>
<td>Simulated other channels</td>
<td></td>
</tr>
<tr>
<td></td>
<td>15.88</td>
</tr>
<tr>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>N</td>
<td>Y</td>
</tr>
</tbody>
</table>
Chime-3 with Smaller Well-matched Parallel Corpus

- With both real and simulated data, T/S learning can get further improved.

<table>
<thead>
<tr>
<th>The noisy data in the pair comes from</th>
<th>Real channel 5</th>
<th>Simulated channel 5</th>
<th>Other real channels</th>
<th>Simulated other channels</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>15.88</td>
</tr>
<tr>
<td></td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>15.73</td>
</tr>
<tr>
<td></td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>13.77</td>
</tr>
</tbody>
</table>
Chime-3 with Smaller Well-matched Parallel Corpus

• More data gives better performance
  • Significantly better than feature mapping and mask learning [3]

<table>
<thead>
<tr>
<th>The noisy data in the pair comes from</th>
<th>Real channel 5</th>
<th>Simulated channel 5</th>
<th>Other real channels</th>
<th>Simulated other channels</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>15.88</td>
</tr>
<tr>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>15.73</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>13.77</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>12.99</td>
</tr>
</tbody>
</table>

When Baseline Model is Trained with 3400hr Transcribed Data

- Evaluated with multiple scenarios – real test utterances

<table>
<thead>
<tr>
<th>Model</th>
<th>Test0</th>
<th>Test1</th>
<th>Test2</th>
<th>Test3</th>
<th>Test4</th>
<th>Test5</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4k hour-transcribed Teacher</td>
<td>62.36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
When Baseline Model is Trained with 3400hr Transcribed Data

- Evaluated with multiple scenarios – real test utterances: T/S learning with simulation works very well for real target-domain speech

<table>
<thead>
<tr>
<th>Model</th>
<th>Test0</th>
<th>Test1</th>
<th>Test2</th>
<th>Test3</th>
<th>Test4</th>
<th>Test5</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4k hour-transcribed Teacher</td>
<td>62.36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T/S with 3.4k hour paired data</td>
<td>17.22</td>
<td>12.78</td>
<td>9.19</td>
<td>14.65</td>
<td>13.89</td>
<td>25.90</td>
</tr>
</tbody>
</table>
When Baseline Model is Trained with 3400hr Transcribed Data

• Evaluated with multiple scenarios – real test utterances: T/S learning with simulation works very well for real target-domain speech

<table>
<thead>
<tr>
<th>Model</th>
<th>Test0</th>
<th>Test1</th>
<th>Test2</th>
<th>Test3</th>
<th>Test4</th>
<th>Test5</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4k hour-transcribed Teacher</td>
<td>62.36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T/S with 3.4k hour paired data</td>
<td>17.22</td>
<td>12.78</td>
<td>9.19</td>
<td>14.65</td>
<td>13.89</td>
<td>25.90</td>
</tr>
<tr>
<td>T/S with 25k hour paired data</td>
<td>15.66</td>
<td>12.35</td>
<td>8.95</td>
<td>12.90</td>
<td>12.23</td>
<td>20.79</td>
</tr>
</tbody>
</table>
New Domain Adaptation without Parallel Data

[Meng17]
Domain-Invariant Training of Acoustic Model: Gradient Reversal Layer Network (GRLN)

- Adversarial Multi-Task Learning
  - Senone-discriminative: \( \min_{\theta_y, \theta_f} L_y(\theta_y, \theta_f) \)
  - Domain-invariant: \( \max_{\theta_f} \min_{\theta_d} L_d(\theta_d, \theta_f) \)
  - Multi-task: \( \max \min \left[ L_y(\theta_y) + \frac{\partial L_y}{\partial \theta_y} \right] \)
- Stochastic Gradient Decent
  - \( \theta_y \leftarrow \theta_y - \mu \frac{\partial L_y}{\partial \theta_y} \)
  - \( \theta_f, \theta_d \leftarrow \theta_f, \theta_d - \mu \left[ \frac{\partial L_y}{\partial \theta_f} - \alpha \frac{\partial L_d}{\partial \theta_d} \right] \)
  - \( \theta_d \leftarrow \theta_d - \mu \frac{\partial L_d}{\partial \theta_d} \)
- Gradient Reversal Layer \( R_\alpha \)
  - Forward pass: \( R_\alpha(f) = f \)
  - Backward pass: \( \frac{\partial L_y}{\partial f} = -\alpha I \)
  - \( I \) is the identity matrix
Private Component Extractor

- Source Private Component Extractor $M^s_p$
- Shared Component Extractor $M_c$
- Target Private Component Extractor $M^t_p$

- Source Difference Loss
- Senone Loss $\hat{y}^s$
- Domain Loss $d^s, d^t$
- Target Difference Loss

- Senone Classifier $M_y$
- Domain Classifier $M_d$

- $f^s_p$, $f^s_c$, $f^t_c$, $f^t_p$

- $x^s$, $x^t$
Reconstructor

- Source Reconstruction Loss
- Source Difference Loss
- Source Private Component Extractor $M_p^s$
- Source Reconstruction $g^s$
- Senone Loss
- Senone Classifier $M_y$
- Senone Loss $d^s, d^t$
- Target Difference Loss
- Target Private Component Extractor $M_p^t$
- Target Reconstruction Loss $g^t$
- Target Reconstruction $g^t$
- Domain Classifier $M_d$
- Shared Component Extractor $M_c$
- Source Private Component $f_p^s$
- Target Private Component $f_p^t$
- Target Difference Loss $d^t$
- Shared $x^s, x^t$
Adversarial Training of Domain Separation Network
ASR Results of DSN for Unsupervised Environment Adaptation

• Test data: CHiME-3 dev set with 4 noise conditions
• WSJ 5K word 3-gram language model is used for decoding

<table>
<thead>
<tr>
<th>System</th>
<th>Data</th>
<th>BUS</th>
<th>CAF</th>
<th>PED</th>
<th>STR</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean</td>
<td>Real</td>
<td>36.25</td>
<td>31.78</td>
<td>22.76</td>
<td>27.18</td>
<td>29.44</td>
</tr>
<tr>
<td>GRL</td>
<td>Real</td>
<td>35.93</td>
<td>28.24</td>
<td>19.58</td>
<td>25.16</td>
<td>27.16</td>
</tr>
<tr>
<td>DSN</td>
<td>Real</td>
<td>32.62</td>
<td>23.48</td>
<td>17.29</td>
<td>23.46</td>
<td>24.15</td>
</tr>
</tbody>
</table>
Multi-talker Separation

[Chen17]
Solving the cocktail problem

- Multi-talker speech separation & recognition
  - Separate and recognize each speaker in highly overlapped environment, e.g. cocktail party
  - The speaker identity and number of speakers are unknown
- Difficulty
  - Tracking multiple speaker largely increase the data & computation complexity
  - Unknown number of speaker is troublesome to neural networks
  - Permutation problem
- Single channel solution
  - Deep clustering/ deep attractor network
  - Permutation Invariation training
- Limitations of single channel processing
  - Performance is still unsatisfactory
  - Difficult to deal with reverberation
  - Multi-channel signal provides spatial clues, which is beneficial for separation
System Architecture

- A fixed set of beamformer
  - 12 fixed differential beamformer, uniformly sample the space
  - A linear operation for beamformer
- Separation network
  - Anchored deep attractor network
  - Pick best two speakers for each beam
  - Additional residual more for noise
- Post selection
  - Selecting each speaker from all 24 outputs
  - Spectral clustering to group the classes
  - Speech quality evaluation to pick best speech for each group
State of the art separation performance

- A new state of the art for multi-talker separation & recognition
  - Similar performance as the ideal ratio mask and the oracle mvdr beamformer
  - Largely improve the single channel system
  - Robustly separating 4 overlapped speakers
  - Significantly improvement for multi-talker speech recognition

- Still a room to further improve
  - Acoustic model retraining/ joint training
  - Mask based beamformer from the separated result

- Example:
  - The sample that has the median performance

  - Mixture:

  - Result:

<table>
<thead>
<tr>
<th></th>
<th>Proposed</th>
<th>IRM</th>
<th>OMVDR</th>
<th>DAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 speaker</td>
<td>+10.98</td>
<td>+11.05</td>
<td>+12.00</td>
<td>+7.82</td>
</tr>
<tr>
<td>3 speaker</td>
<td>+11.54</td>
<td>+11.52</td>
<td>+12.56</td>
<td>+5.16</td>
</tr>
<tr>
<td>4 speaker</td>
<td>+11.19</td>
<td>+12.22</td>
<td>+11.82</td>
<td>+4.23</td>
</tr>
</tbody>
</table>

**Separation result SDR(Db)**

<table>
<thead>
<tr>
<th>Clean model</th>
<th>Mixture</th>
<th>Top 1</th>
<th>Top 2</th>
<th>Top3</th>
<th>Top4</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 speaker</td>
<td>82.29</td>
<td>29.85</td>
<td>31.38</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3 speaker</td>
<td>93.61</td>
<td>31.8</td>
<td>39.21</td>
<td>44.89</td>
<td>-</td>
</tr>
<tr>
<td>4 speaker</td>
<td>95.97</td>
<td>42.31</td>
<td>46.54</td>
<td>53.68</td>
<td>65.67</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Far-field model</th>
<th>Mixture</th>
<th>Top 1</th>
<th>Top 2</th>
<th>Top3</th>
<th>Top4</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 speaker</td>
<td>81.96</td>
<td>23.6</td>
<td>26.38</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3 speaker</td>
<td>94.19</td>
<td>27.95</td>
<td>32.64</td>
<td>40.61</td>
<td>-</td>
</tr>
<tr>
<td>4 speaker</td>
<td>95.91</td>
<td>37.79</td>
<td>40.29</td>
<td>46.1</td>
<td>57.93</td>
</tr>
</tbody>
</table>

**Recognition Result**
Reference


- [Huang13] Jui-Ting Huang, Jinyu Li, Dong Yu, Li Deng, and Yifan Gong, cross-language knowledge transfer using multilingual deep neural network with shared hidden layers, in ICASSP, 2013

- [Li14] Jinyu Li, Rui Zhao, Jui-Ting Huang and Yifan Gong, Learning Small-Size DNN with Output-Distribution-Based Criteria, in Interspeech, 2014.


- [Xue 14] Jian Xue, Jinyu Li, Dong Yu, Mike Seltzer, and Yifan Gong, Singular Value Decomposition Based Low-footprint Speaker Adaptation and Personalization for Deep Neural Network, in ICASSP, 2014