Reverberant speech recognition combining deep neural networks and deep autoencoders

Masato Mimura
Shinsuke Sakai
Tatsuya Kawahara
ACCMS, Kyoto University
The REVERB challenge 2014 workshop
Introduction

• Use **deep learning** in both **frontend** and **backend** of the speech recognizer to handle reverberant speech.
  – Frontend: speech feature enhancement (dereverberation) w/ **deep autoencoder**
  – Backend: acoustic modeling w/ **deep neural networks**
Our submitted results for the challenge and final results on paper

Our submitted results

<table>
<thead>
<tr>
<th></th>
<th>Room1</th>
<th>Room2</th>
<th>Room3</th>
<th>Ave</th>
<th>room1</th>
<th>Ave.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Near</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Far</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Real-time</td>
<td>12.9</td>
<td>13.4</td>
<td>15.9</td>
<td>26.4</td>
<td>18.5</td>
<td>30.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>19.6</td>
<td>52.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>52.3</td>
<td>52.3</td>
</tr>
<tr>
<td>Full batch</td>
<td>13.0</td>
<td>13.3</td>
<td>15.4</td>
<td>24.9</td>
<td>17.9</td>
<td>28.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>28.8</td>
<td>50.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>50.5</td>
<td>50.6</td>
</tr>
</tbody>
</table>

Forgot to include results by full batch adaptation in the paper. Sorry!

Our final results with DAE feature enhancement (and some bug fixes)

<table>
<thead>
<tr>
<th></th>
<th>Room1</th>
<th>Room2</th>
<th>Room3</th>
<th>Ave</th>
<th>room1</th>
<th>Ave.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Near</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Far</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Real-time(a)</td>
<td>10.3</td>
<td>10.6</td>
<td>12.9</td>
<td>21.4</td>
<td>14.1</td>
<td>23.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15.5</td>
<td>49.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>48.1</td>
<td>48.7</td>
</tr>
<tr>
<td>Real-time(b)</td>
<td>14.2</td>
<td>14.2</td>
<td>13.3</td>
<td>19.5</td>
<td>14.0</td>
<td>18.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15.7</td>
<td>45.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>45.2</td>
<td>45.4</td>
</tr>
</tbody>
</table>

Results with DAE enhancement not in time for result submission deadline
Standard procedure for training DNN

unsupervised pre-training

stacking RBMs

supervised fine-tuning

unsupervised pre-training

stacking RBMs

supervised fine-tuning

Target → y

Input → v₁

v₁ = h₁

v₂ = h₂
Hybrid model (DNN-HMM) [Mohamed 12][Dahl 12]

- GMMs for calculating state probabilities replaced by a single DNN
- Other parameters like transition probabilities copied from a well-trained GMM-HMM
Deep autoencoders (DAEs) [06 Hinton] (traditional)

- Deep neural networks used for regression tasks
- Encoder layers generate compact representation for Decoder to recover the input data
- DAE trained as **denoising autoencoder**:
  - Input = corrupted data
  - Target = clean data
Deep autoencoders (DAEs) (our network for dereverberation)

- Since our goal is not generating compact codes, we adopt network structure **without any bottleneck layer** for dereverberation
Our proposed network
(Combination of DNN-HMM and denoising DAE)

**DNN**
- 5 hidden layers × 2,048 nodes
- 40ch FBANK + Δ+ΔΔ × 11 frames (≈ 110 msec)

**DAE**
- 5 hidden layers × 2,048 nodes
- 40ch FBANK + Δ+ΔΔ × 11 frames (≈ 110 msec)

Posteriors of 3,113 HMM states
Speech recognition experiments

• DNN Training
  – input: Multi-condition data  target: Frame-level state labels

• DAE Training
  – input: Multi-condition data  target: Clean data
    • Reverberant speech frames and clean speech frames are adjusted to be time-aligned

• Test data
  – Simulated data: 3264 utts
    • Rooms: Small (T60 = 0.25s), Med (0.5s), Large (0.7s)
    • Mic. distances: Near (= 50cm), Far (= 200cm)
  – Real data: 372 utts:
    • Room: Large (T60 = 0.7s)
    • Mic. distances: Near (= 100cm), Far (= 250cm)
Performance of DNN-HMM for reverberant test data

- GMM(clean)
- GMM(multicond)
- GMM(multicond, MLLR)
- DNN(clean)
- DNN(multicond)

■ vs. ■: DNN-HMMs achieves drastically higher accuracies than adapted GMM-HMMs

■ vs. ■: multi condition training effective for DNN-HMMs as well as GMM-HMMs ( ■ vs. ■ )
Performance of DAE for reverberant test data

By using DAE as frontend, accuracies by clean DNN improved drastically.

Interestingly, performance of clean DNN combined with DAE almost the same as multicond. DNN without DAE.
Example of DAE-enhanced speech feature

Reverberant FBANK feature

→ time

DAE-enhanced FBANK feature

→ time
Effectiveness of combination of multicond. DNN-HMM and DAE

In very adverse conditions, significant improvements obtained by combining DAE with multicond. DNN-HMM

In less adverse conditions, speech “enhancement” by DAE harmful
Conclusion

- **Deep learning effective** for reverberant speech recognition
  - Multi condition training of **DNN-HMMs**
  - Speech feature enhancement by **DAEs**
- Combined DAE and multicond. DNN-HMM achieves larger accuracy improvements in more adverse reverberant conditions.
- **Further error reduction by adapting** DNN-HMMs to the DAE-enhanced features