Bottleneck Features for Speaker Recognition

Odyssey 2012:
The Speaker and Language Recognition Workshop

Sibel Yaman¹, Jason Pelecanos¹, and Ruhi Sarikaya²

¹ IBM T. J. Watson Research Labs, Yorktown Heights, NY
² Microsoft Corporation, Redmond, WA
Roadmap

- Introduction

- Bottleneck feature extraction
  1) A conversation level training criterion
  2) Incorporating a separate system in training

- Experiments

- Summary
The Big Picture

- **In the speech recognition literature:**
  Deep networks are shown to outperform HMMs (Seide 2012, etc.).

- **In the speaker recognition literature:**
Bottleneck Network Architecture

An *information bottleneck* acts as a feature compressor (Konig 1998).

Stacked raw features from 0.5 seconds of speech
Using Neural Networks for Speaker Recognition

- Feature extraction with neural networks traditionally performs relatively poorly.

- We investigate approaches to make the performance comparison the other way around. :-)

![Graph showing EER (%) comparison between MFCC and Traditional NN for same and different mic conditions.]

-40.4 % rel.

-45.1 % rel.

-57.5 % rel.

-46.1 % rel.

1000xmin DCF (2008)
An Overview

We demonstrate two ways of exploiting the expressive power of deep networks:

1) The training is adjusted to the targeted performance evaluation metric.
2) Information from a separate system is incorporated in training.
Roadmap

- **Introduction**

- **Bottleneck feature extraction**
  1) A conversation level training criterion
  2) Incorporating a separate system in training

- **Experiments**

- **Summary**
Frame vs. Conversation Level Training

Frame level training has limitations:
- Learning the speaker is constrained to the context around the current frame.
- A long context would explode the number of free parameters.

Conversation level training offers solutions:
- The frames coming from one conversation are tied together so that a single decision is made.
- The network size can be kept relatively small.
(1) A Speaker Recognition Training Criterion

- A log-likelihood ratio-based training criterion (Brummer 2005) is optimized

\[ J_{LLR}(\Theta) = \alpha \sum_{T:\text{target}} \log(1 + e^{-u_T - c}) + \beta \sum_{N:\text{nontarget}} \log(1 + e^{u_N + c}) \]

Cost associated with target trials
Cost associated with nontarget trials

- There is one target and \((S-1)\) nontarget scores at the output layer.
- We need a global constraint on the decision for the entire recording.
- The scores are averaged at the output layer before the nonlinearity.
(2) Using a Separate System in Training

Scores from a separate system are incorporated in training.

The term

$$u^{(\ell)}(\theta) = W^{(\ell-1)}\sigma^{(\ell-1)}$$

in the training objective is replaced with

$$u'_n(\Theta) = \omega_1 W^{(\ell-1)}\sigma^{(\ell-1)} + \omega_2 u_n^M + \kappa$$
Score Calibration

- The additional scores should have a log-likelihood ratio interpretation.

- The score calibration is achieved by solving

\[
\{\omega_1^*, \omega_2^*, \kappa^*\} = \arg\min_{\omega_1, \omega_2, \kappa} J_{LLR}(\omega_1, \omega_2, \kappa \mid \Theta \text{ fixed})
\]

- The network is trained by solving

\[
\Theta^* = \arg\min_{\Theta} J_{LLR}(\Theta \mid \omega_1^*, \omega_2^*, \kappa \text{ fixed})
\]
The Back-End System

- Bottleneck Feature Extraction
- Universal Background Model Training
- Dimension Reduction
- MAP-Adapted Speaker Modeling
- Probabilistic Linear Discriminant Analysis (PLDA)
- Recognition Scores

STATE-OF-THE-ART SPEAKER RECOGNITION SYSTEM

[Flowchart showing the process with arrows indicating the flow from feature extraction to recognition scores]
Roadmap

- Introduction

- Bottleneck feature extraction
  1) A conversation level training criterion
  2) Incorporating a separate system in training

- Experiments

- Summary
Experiments

- We ran experiments on the same and different microphone tasks of NIST SRE 2010.
- Microphone recordings were used in bottleneck network training.
  - 173 speakers in the training and validation sets
  - 4341 recordings in training and 865 recordings in validation
- Network architecture:
  - 294 dimensional input $\rightarrow$ 1000 x 42 x 500 $\rightarrow$ 173 speakers
Processing of the Input and Output Features of the Network

- Input features are mean and variance normalized to better condition the network.

- The bottleneck features are decorrelated for modeling with diagonal covariance GMMs.
Effect of the Training Criterion

-34.2% rel.

-30.0% rel.

-36.4% rel.

EER (%)
Dependence on Feature Size

EER (%)

Same mic – EER (%)  Different mic – EER (%)

BN – 42 dim  BN – 60 dim  BN – 100 dim

1000xmin DCF (2008)

Same mic – D08  Different mic - D08

BN – 42 dim  BN – 60 dim  BN – 100 dim
Performance when Trained with Information from a Separate System

<table>
<thead>
<tr>
<th>Same mic – EER (%)</th>
<th>Different mic – EER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>+14.0 % rel.</td>
<td>+18.0 % rel.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Same mic – D08</th>
<th>Different mic - D08</th>
</tr>
</thead>
<tbody>
<tr>
<td>+11.0 % rel.</td>
<td>+12.0 % rel.</td>
</tr>
</tbody>
</table>

- **EER (%)**
- **1000xmin DCF** (2008)
- **MFCC**
- **Linear Score Combination**
- **Incorporating a Separate System**
Summary

1) We showed how to train a neural network for use in the front-end of a speaker recognition system.

   – A conversation level training criterion that minimizes a log-likelihood ratio score-based cost function is developed.

2) We also showed how to use neural networks to exploit information from a separate system.
Thank you!