AN IMPROVED DEEP NEURAL NETWORK FOR MODELING SPEAKER CHARACTERISTICS AT DIFFERENT TEMPORAL SCALES

Bin Gu, Wu Guo, Lirong Dai and Jun Du
National Engineering Laboratory of Speech and Language Information Processing
University of Science and Technology of China, China
ICASSP2020

Presented by Bin Gu
2020.4
Outline

• Introduction
• Proposed Method
• Experiments and Analysis
• Conclusion
What is speaker verification?

- Speaker verification (SV) is the task of determining whether the claimed identity of a speaker matches an enrolled identity by using voice characteristics.

How does it work?

- Front-end: low dimensional speaker embedding learning (i-vector, x-vector).

- Back-end: calculate the similarity between speaker embeddings (PLDA).
• i-vector/PLDA methods
  – Incorporating local acoustic variability information into short duration speaker verification (Ma et. al)

• Deep embedding learning
  – use DNNs that are trained as acoustic models for automatic speech recognition (ASR) to enhance the modeling of the i-vectors, including DNN-ivector (Lei et al.) and so on.

  – first deal with frame-level acoustic features, and then use a pooling layer to map features to utterance-level, including TDNN (Snyder et al.), CNN (Kenny et al.), LSTM (Heigold et al.).
• **Comparison of existing methods**

<table>
<thead>
<tr>
<th>Pros</th>
</tr>
</thead>
<tbody>
<tr>
<td>i-vector</td>
</tr>
<tr>
<td>deep embedding learning</td>
</tr>
</tbody>
</table>

• **Motivation**

Exploit context temporal information at different temporal scales

– Since neural network is good at exploit frame-level information efficiently, we could improve its ability.

– Applying utterance-level speaker information in neural network could be useful.
Outline

• Introduction
• Proposed Method
• Experiments and Analysis
• Conclusion
X-vector: a typical SV system framework (Snyder et al.)
• X-vector: a typical SV system framework (Snyder et al.)
Proposed Method – framework

- Multiscale convolution neural network:
  
  - $K$ sets of convolution filters $\{W_{l+1}^1, \ldots, W_{l+1}^K\}$ with various dilation factors are used

  - The output of $l + 1^{th}$ layer $H_l$ consists of $C$ 1-dimentional vectors $[s_{l+1}^1, \ldots, s_{l+1}^C]$

  $$s_{l+1}^c = \text{relu}(W_{l+1}^k * H_l + b), c \in [\lambda(k - 1), \lambda k]$$

  $$H_{l+1} = [s_{l+1}^1, \ldots, s_{l+1}^C]$$
Proposed Method – framework

- BWSA-based statistics pooling:
  - Value:
    \[ h_t^L \]
  - Query:
    \[ q_t = d(h_t^{L-1}) \]
  - Key:
    \[ f_m = \sum \gamma_t(m) x_t / T, m = 1, ..., M \]
    \[ f_m = V_2 \tanh(V_1 f_m + b) \]
    \[ K = [\tilde{f}_1, ..., \tilde{f}_m, ..., \tilde{f}_M, w_1, ..., w_n, ..., w_N]^T \]
    \[ = [\tilde{F}, W]^T \]
Proposed Method – framework

- BWSA-based statistics pooling:
  - Attention weight:
    \[ e_t = f_{BA}(h_{t-1}) = v^T \tanh(Kq_t + b) \]
    \[ \alpha_t = \frac{\exp(e_t)}{\sum_{k=1}^{T} \exp(e_k)} \]
  - Statistics pooling
    \[ \mu = \sum_{k=1}^{T} \alpha_t h_t^L \]
    \[ \sigma = \sqrt{\sum_{t} \alpha_t h_t^L \odot h_t^L - \mu \odot \mu} \]
    \[ c = [\mu, \sigma] \]
Outline

• Introduction
• Proposed Method
• Experiments and Analysis
• Conclusion
• **Training set:**
  – NIST SRE 2004-2010 evaluation set, Switchboard and Mix6 dataset.

• **Testing set:**
  – NIST SRE 2016 (Tagalog and Cantonese)

• **Features:**
  – 23-dimensional MFCCs
  – 25ms windows, 10ms shift
  – mean normalization over a sliding 3s window
  – voice activity detection (VAD)
Experiments and Analysis – experiment setup

• i-vecotr:
  – I-vector baseline system

• x-vector:
  – X-vector baseline system

• SA:
  – System applying self-attention

• IA:
  – System applying i-vector based attention

• BA:
  – System applying Baum-Welch statistics attention

• BA+MS-3L:
  – System applying BWSA and multiscale convolution
Experiments and Analysis – results

- Comparison results of different systems on SRE16

<table>
<thead>
<tr>
<th>Systems</th>
<th>Pooled</th>
<th>Taglog</th>
<th>Cantonese</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EER</td>
<td>DCF\text{min}</td>
<td>EER</td>
</tr>
<tr>
<td>i-vecter</td>
<td>14.08</td>
<td>0.739</td>
<td>17.31</td>
</tr>
<tr>
<td>x-vector</td>
<td>7.99</td>
<td>0.587</td>
<td>11.58</td>
</tr>
<tr>
<td>SA</td>
<td>7.61</td>
<td>0.575</td>
<td>11.04</td>
</tr>
<tr>
<td>IA</td>
<td>7.81</td>
<td>0.586</td>
<td>11.15</td>
</tr>
<tr>
<td>BA</td>
<td>7.29</td>
<td>0.569</td>
<td>10.74</td>
</tr>
<tr>
<td>BA+MS-3L</td>
<td>7.04</td>
<td>0.561</td>
<td>10.34</td>
</tr>
</tbody>
</table>
Experiments and Analysis – results

- Comparison results of different systems applying MSCNN with different system configurations.

<table>
<thead>
<tr>
<th>Systems</th>
<th>$L$</th>
<th>$K$</th>
<th>$N$</th>
<th>EER</th>
<th>DCF$^{\text{min}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-vecter</td>
<td>-</td>
<td>-</td>
<td>512</td>
<td>7.99</td>
<td>0.587</td>
</tr>
<tr>
<td>x-vector*</td>
<td>-</td>
<td>-</td>
<td>756</td>
<td>8.11</td>
<td>0.596</td>
</tr>
<tr>
<td>MS-1L</td>
<td>1</td>
<td>2</td>
<td>512</td>
<td>7.88</td>
<td>0.589</td>
</tr>
<tr>
<td>MS-2L</td>
<td>2</td>
<td>2</td>
<td>512</td>
<td>7.65</td>
<td>0.575</td>
</tr>
<tr>
<td>MS-3L</td>
<td>3</td>
<td>2</td>
<td>512</td>
<td>7.60</td>
<td>0.572</td>
</tr>
<tr>
<td>MS-3L*</td>
<td>3</td>
<td>3</td>
<td>756</td>
<td>7.51</td>
<td>0.571</td>
</tr>
</tbody>
</table>

“$L$” indicates the number of layers applying the MSCNN. “$K$” is the number of convolution filters with various dilation factors. “$N$” denotes the MSCNN layer size.
Outline

• Introduction
• Proposed Method
• Experiments and Analysis
• Conclusion
• Conclusion

  – The information with different granularities at the frame level can be detected by MSCNN.
  
  – BWSA-based statistics pooling could capture utterance-level speaker information very well.
Thank you for your attention!