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

Introduction - background

- 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).

Introduction - existing methods

- 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.)*.

Introduction - motivation

- Comparison of existing methods

	Pros
i-vector	extract high-order statistics from input features and capture long-term speaker characteristics effectively
deep embedding learning	extract speaker representation at small time scales and perform well in short duration conditions

- 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.

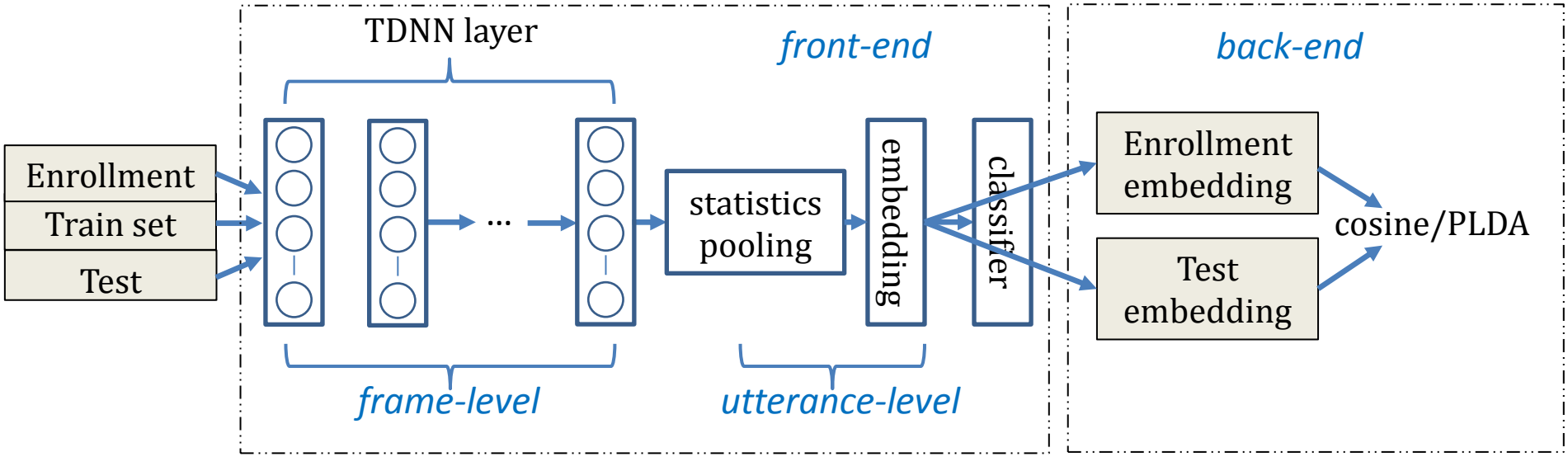
  – Applying utterance-level speaker information in neural network could be useful.

Outline

- Introduction
- Proposed Method
- Experiments and Analysis
- Conclusion

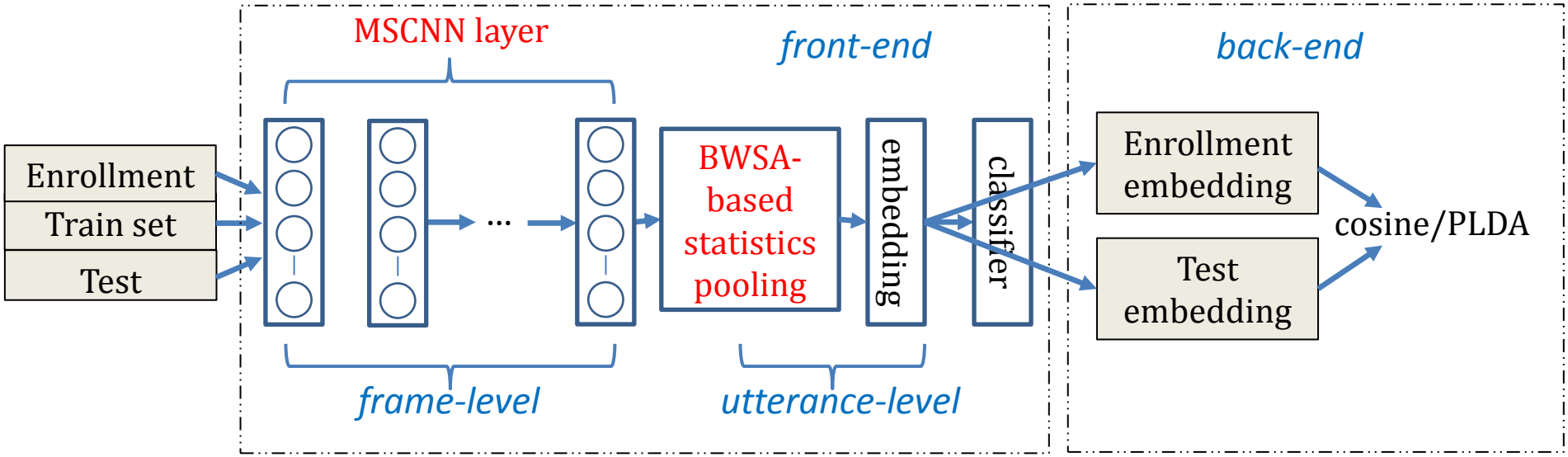
Proposed Method – framework

- X-vector: a typical SV system framework (Snyder et al.)



Proposed Method – framework

- X-vector: a typical SV system framework (Snyder et al.)



Proposed Method – framework

- Multiscale convolution neraul network:
  - *K* sets of convolution filters $\{\mathbf{W}_{l+1}^1, \ldots, \mathbf{W}_{l+1}^K\}$ with various dilation factors are used
  - The output of $l+1^{th}$ layer $\mathbf{H}_l$ consists of *C* 1-dimentional vectors $[\mathbf{s}_{l+1}^1, \ldots, \mathbf{s}_{l+1}^C]$

$$\mathbf{s}_{l+1}^c = \text{relu}(\mathbf{W}_{l+1}^k * \mathbf{H}_l + \mathbf{b}), c \in [\lambda(k-1), \lambda k]$$

$$\mathbf{H}_{l+1} = [\mathbf{s}_{l+1}^1, \ldots, \mathbf{s}_{l+1}^C]$$

Proposed Method – framework

- BWSA-based statistics pooling:
  - Value:

$$\mathbf{h}_t^L$$

  - Query :

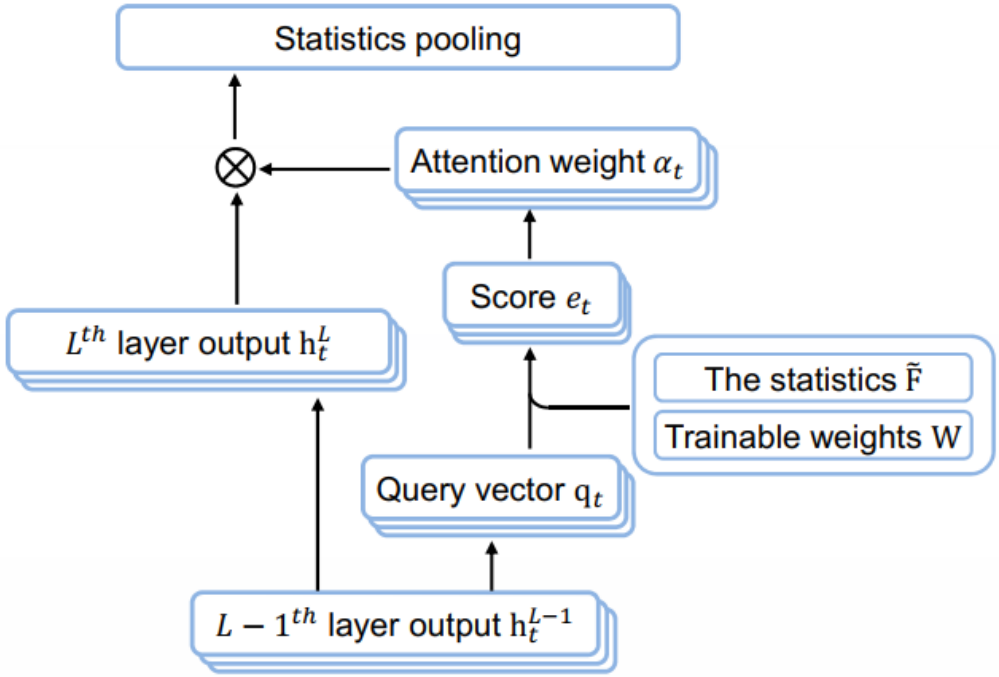
$$\mathbf{q}_t = d(\mathbf{h}_t^{L-1})$$

  - Key :

$$\mathbf{f}_m = \sum_t \gamma_t(m)\mathbf{x}_t/T, m = 1, ..., M$$

$$\mathbf{f}_m = \mathbf{V}_2 \tanh(\mathbf{V}_1\mathbf{f}_m + \mathbf{b})$$

$$\mathbf{K} = [\tilde{\mathbf{f}}_1, ..., \tilde{\mathbf{f}}_m, ..., \tilde{\mathbf{f}}_M, \mathbf{w}_1, ..., \mathbf{w}_n, ..., \mathbf{w}_N]^T$$
$$= [\tilde{\mathbf{F}}, \mathbf{W}]^T$$



Proposed Method – framework

- BWSA-based statistics pooling:
  - Attention weight :

$$e_t = f_{BA}(\mathbf{h}_t^{L-1}) = \mathbf{v}^T \tanh(\mathbf{K}\mathbf{q}_t + \mathbf{b})$$

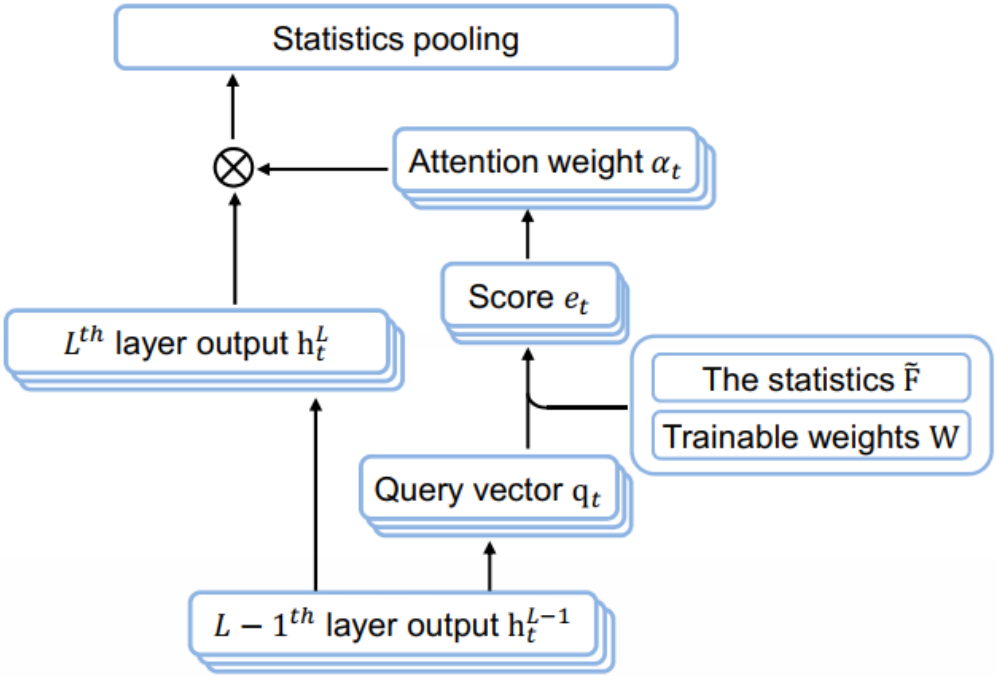
$$\alpha_t = \frac{\exp(e_t)}{\sum_{k=1}^{T} \exp(e_k)}$$

  - Statistics pooling

$$\mu = \sum_{k=1}^{T} \alpha_t \mathrm{h}_t^L$$

$$\sigma = \sqrt{\sum_t \alpha_t \mathrm{h}_t^L \odot \mathrm{h}_t^L - \mu \odot \mu}$$

$$\mathbf{c} = [\boldsymbol{\mu}, \boldsymbol{\sigma}]$$



Outline

- Introduction
- Proposed Method
- Experiments and Analysis
- Conclusion

Experiments and Analysis – dataset

- 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 appling BWSA and multiscale convolution

Experiments and Analysis – results

- Comparison results of different systems on SRE16

Systems	Pooled		Taglog		Cantonese	
	EER	DCF^{min}	EER	DCF^{min}	EER	DCF^{min}
i-vecter	14.08	0.739	17.31	0.864	8.20	0.597
x-vector	7.99	0.587	11.58	0.741	4.26	0.430
SA	7.61	0.575	11.04	0.729	4.23	0.423
IA	7.81	0.586	11.15	0.736	4.54	0.437
BA	7.29	0.569	10.74	0.733	3.88	0.402
BA+MS-3L	**7.04**	**0.561**	**10.34**	**0.725**	**3.77**	**0.398**

Experiments and Analysis – results

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

Systems	*L*	*K*	*N*	Pooled	
				EER	DCF^{min}
x-vecter	-	-	512	7.99	0.587
x-vector*	-	-	756	8.11	0.596
MS-1L	1	2	512	7.88	0.589
MS-2L	2	2	512	7.65	0.575
MS-3L	3	2	512	7.60	0.572
MS-3L*	3	3	756	**7.51**	**0.571**

"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

- 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.

The End

Thank you
for
your attenion!