AntVoice: The Neural Speaker Embedding System for FFSVC 2020

Zhiming Wang

Ant Financial Services Group, Hangzhou, China

{zhiming.wzm}@antgroup.com

Abstract

This paper describes the AntVoice systems, developed by the Ant Financial Service Group, for the tracks of far-field speaker verification from single microphone array in FFSVC 2020. We explore the cross-channel relationship modeling technique and a combination of additive cosine margin softmax loss and equidistant triplet-based loss for metric learning. On both tracks of speaker verifications, our system performances fully surpass the baselines, even just with the single model. Our submissions were a fusion of several state-of-the-art encoding neural network models, that leads to consistent performance improvement.

Index Terms: far-field speaker identification, neural speaker embedding, Squeeze-and-Excitation network, additive cosine margin softmax, equidistant triplet-based loss

1. Introduction

In this paper, we describe the speaker verification systems, AntVoice, developed by the Ant Financial Service Group for the 2020 Far-Field Speaker Verification Competition (FFSVC 2020) [1]. The competition aims to address challenges related to far-field speaker verification. It is separate into tasks for 1) far-field text-dependent speaker verification from single microphone array, 2) far-field text-independent speaker verification from single microphone array, and 3) far-field text-dependent speaker verification from distributed microphone arrays. To simulate real usage scenarios, various background noise is played during recording, and enrollment utterances are collected in close-talking microphones while testing utterances in far-field microphone arrays. As a consequence, participants need to develop systems that are robust to background noise, room reverberations, mismatch between recording channels, *et al*. Our team has participated in tasks 1 and 2, which both involve speaker verification from single microphone array.

The field of speaker verification has advanced significantly due to the development of speaker embeddings using deep neural network [2, 3]. This paradigm has been very effective, achieving state-of-the-art results on speaker verification benchmark datasets. It generally consists of a pooling layer on top of encoded frame-level feature representations, to obtain the segment-level information of speech segments. Various temporal pooling techniques have been investigated [2, 4, 5, 6]. Once the segment-level information is obtained, it is mapped via feed forward network classifier to corresponding speaker ids. Although the effectiveness of this paradigm was demonstrated for speaker verification in close-talking microphones [7], we found in this paper that more techniques need to be developed for far-field speaker verification.

In this paper, we report our approach as follows. Section 2 describes the front-end and data augmentation that increases diversity of the training data. In Section 3, we describe the neural speaker embeddings we used in this competition. Results are reported in Section 4. Finally, in Section 5 we conclude the paper.

2. The Front-end and Data Augmentation

Input Feature. Audios are resampled to 16,000 Hz. 80-dimensional logarithm mel filter banks are generated within a 25ms sliding window with a hop size of 10ms, cepstral mean normalization(CMN) is first performed within a 3 second sliding window and then cepstral mean and variance normalization(CMVN) applied whinin the whole utterance. Energy based voice activity detector(VAD) is employed to remove silence frames; specially, we use adaptive threshold per utterance, that is, 1.0325 times the average energy of the head-and-tail respective 30 frames. Other acoustic feature configurations are the same as the default in Kaldi [8].

During training, chunks of $1 \sim 4.5$ second long audio segments are randomly sampled from recordings; specially, in a mini-batch, the frame length L is uniformly distributed within a certain interval range , e.g., [100, 256] in the first task(for shorter utterances) and [200, 450] in the second task. Then a chunk of segments give a batch of acoustic features of size $B \times L \times 80$, where B is the mini-batch size.

Speech Enhancement. To enhance speech quality, the algorithm of weighted prediction error(WPE)[9, 10] is used to reduce signal dereverberation for enrollment and testing recordings.

Data Augmentation. To reduce mismatching between close-talking and far-field speech and strengthen model robustness, we augment the training recordings from the microphone and the cellphone. To be specific, we use Pyroomacoustics [11], based on the algorithm of image source model(ISM), to simulate the room impulse response(RIR) generator, and one utterance from the microphone and the cellphone respectively generates 5 replicas from diverse microphone arrays which are placed at -1.5m, 1m, 3m, 5m, left or right 3m far from sound source. Additionally, with the method in Kaldi[1] [8], reverberation, noise, music and babble are mixed into the training samples at random signal-to-noise ratio(SNR) between 0 to 20 dB, resulting in about 750,000 augmented recordings.

As in [1, 12], we use the background noise of the testing utterance to perform enrollment augmentation. Specifically, we adopt the above VAD method to detect the non-speech parts of the testing utterance for each trial; then in line with the SNR of the testing recording, these non-speech parts are mixed with the original enrollment utterance to get a simulated one.

3. Neural Speaker Embeddings

3.1. Encoder Networks

Since [2, 3], methods to represent speaker characteristics are dominated by deep neural networks. These approaches use

[1] github.com/kaldi-asr/kaldi/blob/master/egs/sre16/v2.

encoder networks to extract frame-level representations from acoustic features; that is followed by a pooling layer to aggregate into segment-level speech characteristics; and finally, one fully connected classification network projects the extracted segment-level representations to corresponding speaker ids. We term the segment-level speaker characteristics as neural speaker embeddings or x-vectors. It is important to have enough discriminative information in the embeddings to distinguish different speakers.

E-TDNN and F-TDNN. We use E-TDNN and F-TDNN as in [13], but with the following differences: Exponential Linear Unit(ELU) defined as $f(x) = max(0, x) + min(0, e^x - 1)$, rather than ReLU, is used as the nonlinear activation; for the outputs of each parameter layer, batch normalisation is applied before the nonlinearity; the nonlinear activations of size 512 at the penultimate layer are used as embedding features for verification tasks as in [7]. These distinctions are also applicable to ResNet and SE-ResNet below.

ResNet. We use standard 34-layer convolutional residual network (ResNet) architecture [14] in our experiments, as is described in Table 1. In Table 1, $[(3 \times 3, 64)_2] \times 3$ means 3 residual blocks, one of them consisting of 2 convolutional layers with kernel size of 3×3 and 64 filters, others in analogy; for the first block of Res2 $\sim$ 4 with different numbers of channels between the input and output, a short cut connection between them is needed, using one convolutional layer with kernel size of 1×1.

Layer	Configuration
Conv1	$(3 \times 3, 64)$, stride (1×2)
Res1	$[(3 \times 3, 64)_2] \times 3$
Res2	$[(3 \times 3, 128)_2] \times 4$
Res3	$[(3 \times 3, 256)_2] \times 6$
Res4	$[(3 \times 3, 512)_2] \times 3$
Conv2	$(3 \times 3, 512)$, stride (1×2)
Pooling	statistical pooling(mean + std dev)
Linear1	2048×512
Linear2	512×512(embedding features)
Classifier	$512 \times$ #Spks

Table 1: *The convolutional ResNet-34 architecture.*

SE-ResNet. For modeling channel relationship, we use "Squeeze-and-Excitation" residual network(SE-ResNet) [15] in our experiments. In SE-ResNet, the SE block adaptively recalibrates per channel feature responses by explicitly modelling interdependencies between channels, which corresponds to channel-wise attention mechanism. While the convolutional residual part of SE-ResNet is as in Table 1, the SE block is depicted in Table 2 where N_c is the number of channels.

Layer	Configuration
Linear1	$N_c \times max(N_c/16, 32)$
Nonlinear1	ELU
Linear2	$max(N_c/16, 32) \times N_c$
Nonlinear2	Sigmoid

Table 2: *The "Squeeze-and-Excitation" block of SE-ResNet.*

3.2. Objective Loss Function

In our experiments, the objective loss function is the addition of additive cosine margin softmax loss and equidistant triplet-based loss, as are described below.

Additive Cosine Margin Softmax Loss. Additive cosine margin softmax loss, aka $CosAMS$, was proposed in [16]. With embedding feature x_j as the jth observation of a mini-batch and the constraint of zero biases in the classifier layer, $cos(\theta_{<x_j, w_c>}) = \frac{w_c^T x_j}{||w_c|| \cdot ||x_j||}$, where w_c is the weight vector corresponding to class c, $\theta_{<x_j, w_c>}$ is the angle between x_j and w_c. The $CosAMS$ loss function is as follows,

$$
\begin{aligned}
L_{CosAMS} &= -\frac{1}{B} \sum_{j=1}^{B} \log \frac{e^{\eta(\cos(\theta_{<x_j, w_{y_j}>}) - m)}}{Z_{x_j}}, \\
Z_{x_j} &= e^{\eta(\cos(\theta_{<x_j, w_{y_j}>}) - m)} + \sum_{i \neq y_j} e^{\eta \cos(\theta_{<x_j, w_i>})},
\end{aligned}
\tag{1}
$$

where y_j is the label corresponding to x_j, η is the scale hyperparameter, and m is the margin that forces x_j to be more discriminiative.

To avoid local optimum or divergence when training models with the discriminative loss function such as L_{CosAMS}, we use an annealing strategy on m to make training process stable [7]. Empirically, we increase the margin m linearly from 0 to the target margin value as $m = min(m_{max}, m_{inc} \times \tilde{e})$, where $\tilde{e} \in [0, 1, 2, ...]$ is the training epoch index. In our experiments, we set $\eta = 30$, $m_{inc} = 0.07$, $m_{max} = 0.2$.

Equidistant Triplet-based Loss. Equidistant triplet-based loss, or $EDTri$ for short, was proposed in [17]. While traditional triplet loss aims to force the distance between the matched positive sample and the anchor less than that between the mismatched negative one and the anchor by at least a given margin α, $EDTri$ loss further introduces equidistant constraint terms that pull the matched samples closer by adaptively constraining two samples of the same class to be equally distant from another one of a different class in each triplet. By optimizing $EDTri$ loss, the algorithm progressively maximizes intra-class similarity and inter-class variances, contributing to more discriminative embeddings.

To be specific, for a certain anchor x_a, we choose the closest mismatched sample x_n and the farthest matched sample x_p in the embedding feature space to form a triplet $\{x_a, x_p, x_n\}$, where the labels satisfy $y_p = y_a$ and $y_n \neq y_a$. The $EDTri$ loss function is as

$$
\begin{aligned}
L_{EDTri} &= L_{Tri} + L_{EquiD}, \\
L_{Tri} &= \frac{1}{B} \sum_{j=1}^{B} [d(x_a, x_p) - d(x_a, x_n) + \alpha]^+, \\
L_{EquiD} &= \frac{1}{B} \sum_{j=1}^{B} ([d(x_a, x_p) - d(x_p, x_n) + \alpha]^+ + \\
&\qquad |d(x_p, x_n) - d(x_a, x_n)|),
\end{aligned}
\tag{2}
$$

where d is the $l2$-norm distance, $[\cdot]^+ = \max(\cdot, 0)$. In our experiments, let $\alpha = 0.3$.

3.3. Model Training

Inspired by the transfer learning strategy in [12], we use the corpora in OpenSLR [2](including SLR18, SLR33, SLR38, SLR47, SLR49, SLR62 and SLR68, in total of 9127 speakers) to pretrain the speaker embedding models as described in Sec. 3.1. Then the models are finetuned with domain-dependent dataset as in [1]: in the first task, use the HI-MIA dataset(SLR85, [18]) and the first 30 utterances of FFSVC 2020 training dataset; in the second task, use the remaining FFSVC 2020 training

[2]http://openslr.org.

Description	Task#1		Task#2	
	minDCF	EER(%)	minDCF	EER(%)
Development dataset				
baseline[1]	0.57	6.01	0.58	5.83
E-TDNN	0.456	4.32	0.685	6.76
F-TDNN	0.495	4.45	0.704	6.91
ResNet	0.427	3.58	**0.557**	**4.9**
SE-ResNet	**0.394**	**3.12**	0.569	5
SE-ResNet + PLDA	0.464	4.286	-	-
fusion	**0.329**	**2.61**	**0.499**	**4.23**
Evaluation dataset(30% trials)				
baseline[1]	0.62	6.37	0.66	6.55
submission	**0.4557**	**4.25**	**0.5482**	**4.72**

Table 3: *Performance results for speaker verification, with the cosine similarity as back-end scoring in general, unless otherwise specified.*

dataset; both plus the corresponding augmented audios as described in Sec.2.

Models are trained using the RADAM optimizer [19], the weight decay is 5×10^{-4}. The learning rate is scheduled with the cyclical strategy [20], which brings in two benefits: to allow more rapid traversal of saddle point plateaus and to hit the optimum learning rate. The hyper-parameters of cyclical learning rate(CLR) are listed in Table 4. At the beginning of each epoch, training samples are randomly shuffled.

hyper-parameter	pretraining	finetuning
max_lr	10^{-3}	2.5×10^{-5}
base_lr	2.5×10^{-4}	6.25×10^{-6}
up_step_size	2 epochs	half an epoch

Table 4: *The hyper-parameters of CLR.*

4. Evaluations and Results

4.1. Scoring

Utterances with the whole length are used for evaluation. The cosine similarity and probabilistic linear discriminant analysis(PLDA) [21] serve as back-end scoring, but for the latter, we could not achieve performance improvement. For each trial, two utterances, inclusive of the original and simulated ones (as described in Sec.2), are used for enrollment, their embeddings are first whitening, then normalized to the length and finally averaged into the speaker embedding; the same method is also appropriate for the testing audios from 4 channels of the far-field microphone arrays.

4.2. Fusion

As the fusion strategy, the scores from the models of E-TDNN, F-TDNN, ResNet, SE-ResNet are linearly weighted into one regression value, where the weighting coefficients, as listed in Table 5, are tuned on the development dataset.

Task	E-TDNN	F-TDNN	ResNet	SE-ResNet
#1	0.13	0.15	0.26	**0.46**
#2	0	0.15	**0.57**	0.28

Table 5: *The weighting coefficients for score fusion.*

4.3. Performance Results

The performance results for speaker verification on the development and evaluation dataset are reported in Table 3. As the baseline system [1], we adopt minimum detection cost function(minDCF, with P_{target} = 0.01) as primary metric, and equal error rate (EER) as auxiliary one; their smaller values correspond to better performances.We observe that, in both tasks, our methods perform much better than the baselines, even just with the single model(ResNet or SE-ResNet). From Table 3 and 5, we could see that the SE-ResNet model contributes much more to the performance improvement in the first text-dependent verification task, which may be attributed to the cross-channel correlation modeling capability that is effective for short utterances; but the same tendency could not be observed yet in the second text-independent task.

4.4. Model's Efficiency

In the Table 6 are the parameter size and the average inference run time on NVIDIA Tesla P100 GPU of the above encoder networks, with a single thread and 64-processor 2.5-GHz Intel(R) Xeon(R) CPUs. For these configs, the inference run time on CPU is usually $4 \sim 5$ times slower than that on GPU. Notice that the acoustic feature processing is on CPU, and the processing time for enrollment data accounts for a third.

Model	#Parameter(M)	run time per trial(ms)	
		Task#1	Task#2
E-TDNN	6	29.64	40.38
F-TDNN	13	33.6	42.34
ResNet	25	57.6	57.33
SE-ResNet	25	93.48	86.8

Table 6: *The parameter size and the average inference run time on GPU.*

5. Conclusions

This paper describes our AntVoice system in far-field speaker verification for FFSVC 2020. We use SE-ResNet for cross-channel relationship modeling that brings in consistent performance improvement in the text-dependent verification task. Moreover, a combination of additive cosine margin softmax loss and equidistant triplet-based loss is explored for enlarging

intra-class similarity and inter-class variances which contributs to more discriminative embeddings. Based on these innovative methods, our system performances fully surpass the baselines, even just with the single model. The far-field speaker recogntion task under complex acoustic conditions is challenging and deserves more further research.

6. References

[1] X. Qin, M. Li, H. Bu, W. Rao, R. K. Das, S. Narayanan, and H. Li, "The interspeech 2020 far-field speaker verification challenge," *arXiv preprint arXiv:2005.08046*, 2020.

[2] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. Interspeech*, 2017, pp. 999–1003.

[3] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.

[4] F. R. R. Chowdhury, Q. Wang, I. LopezMoreno, and L. Wan, "Attention-based models for text-dependent speaker verification," in *IEEE International Con-ference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5359–5363.

[5] G. Bhattacharya, J. Alam, and P. Kenny, "Deep speaker embeddings for short-duration speaker verification," in *Interspeech*, 2017, pp. 1517–1521.

[6] Z. Wang, K. Yao, X. Li, and S. Fang, "Multi-resolution multi-head attention in deep speaker embedding," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[7] Z. Wang, K. Yao, S. Fang, and X. Li, "Joint optimization of classification and clustering for deep speaker embedding," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 284–290.

[8] D. Povey, A. Ghoshal, G. Boulianne, and et al., "The kaldi speech recognition toolkit," in *IEEE Automatic Speech Recognition and Understanding Workshop(ASRU)*, 2011.

[9] T. Yoshioka and T. Nakatani, "Generalization of multi-channel linear prediction methods for blind mimo impulse response shortening," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 10, pp. 2707–2720, 2012.

[10] L. Drude, J. Heymann, C. Boeddeker, and R. Haeb-Umbach, "NARA-WPE: A python package for weighted prediction error dereverberation in Numpy and Tensorflow for online and offline processing," in *13. ITG 2018*, Oct. 2018.

[11] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A python package for audio room simulation and array processing algorithms," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 351–355.

[12] X. Qin, D. Cai, and M. Li, "Far-field end-to-end text-dependent speaker verification based on mixed training data with transfer learning and enrollment data augmentation," in *Proc. Interspeech*, 2019, pp. 4045–4049.

[13] J. Villalba, N. Chen, D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, J. Borgstrom, L. P. García-Perera, F. Richardson, R. Dehak *et al.*, "State-of-the-art speaker recognition with neural network embeddings in nist sre18 and speakers in the wild evaluations," *Computer Speech & Language*, vol. 60, p. 101026, 2020.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[15] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2018, pp. 7132–7141.

[16] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[17] F. Xu, W. Zhang, Y. Cheng, and W. Chu, "Metric learning with equidistant and equidistributed triplet-based loss for product image search," in *The Web Conference(WWW)*, 2020, pp. 57–65.

[18] X. Qin, H. Bu, and M. Li, "Hi-mia: A far-field text-dependent speaker verification database and the baselines," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7609–7613.

[19] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," *arXiv preprint arXiv:1908.03265*, 2019.

[20] L. N. Smith, "Cyclical learning rates for training neural networks," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017, pp. 464–472.

[21] S. Ioffe, "Probabilistic linear discriminant analysis," in *European Conference on Computer Vision(ECCV)*. Springer, 2006, pp. 531–542.