Extraction of Noise-Robust Speaker Embedding Based on Generative Adversarial Networks

Jianfeng Zhou[1], Tao Jiang[2], Qingyang Hong*[2] and Lin Li*[1]
[1]School of Electronic Science and Engineering, Xiamen University, China
[2]School of Information Science and Engineering, Xiamen University, China
E-mail: {jfchou,taojiang}@stu.xmu.edu.cn,{qyhong,lilin}@xmu.edu.cn

***Abstract*—In the field of speaker verification, the speaker systems based on x-vector framework are widely used in many scenarios. However, it suffers from the performance degradation caused by noise disturbance. In this paper, we firstly analyzed the noisy robustness of x-vector by training the networks using a mixture dataset which includes clean data and corrupted data. Then, we proposed a novel adversarial strategy against noise interference and extracted the noise-robust speaker embedding with x-vector. The proposed adversarial method named as triple-net GAN employs three connected networks: a generator network (G), a discriminator network (D) and a classifier network (C). The spectral coefficients of clean and noisy speech utterances are fed to the G, of which the structure is nearly the same as x-vector. The outputs of G are transferred in a parallel way to the D and C. And the labels of D are set binary for clean data and corrupted data, while the labels of C are set corresponding to speaker identities, which aims to learn the speaker embedding features invariant to the noise. Finally, we executed the experiments with different variants of triple-net GAN to verify the denoising capability of the proposed adversarial method. Experimental results on Librispeech corpus demonstrate that our proposed method could achieve a better performance under the noisy environments.**

Keywords: noise-robust, generative adversarial networks, speaker embedding, speaker verification.

I. Introduction

Recently deep learning [1] has shown remarkable success in speech processing tasks [2], [3], [4]. Several solid studies [5], [6], [7] focused on using different neural networks to verify the speaker indentities. The most typical one is the x-vector [8], which uses a feed-forward deep neural network to map the utterances into the fixed-dimension speaker embeddings and then PLDA [9] is used for scoring. Since proposed by Snyder et al, the x-vector has outperformed the i-vector [10] system in most scenarios of speaker verification, and has been widely recognized as one of the state-of-the-art frameworks.

In the field of speaker verification, there is a large quantity of literature concerning the noise robustness of the speaker verification systems, since the noises sharply degrade the performance. A common way to improve the noise robustness of the speaker verification systems is to train the PLDA model using a dataset consisting of clean data and corrupted data (data with noise) [11], [12]. In addition, most of the denoising methods, like [13], a PLDA mixture model for noisy robustness [14] and model adaptation on noise condition [15], are based on the traditional speaker verification systems, like GMM-UBM [16] or i-vector. However, rare methods [17] are concentrated on improving the noise robustness on the embedding level.

More recently, much attention has been poured into exploring the possibility of generative adversarial networks (GANs) [18]. Several approaches based on GANs have also been proposed to solve the issues in speaker verification. In the paper [19], Zhang et al. attempted to use conditional GANs to solve the impact of performance degradation caused by the variable-duration of utterances. Ding et al. [20] proposed a multi-task GANs framework to extract more distinctive speaker representation. And Yu et al. [21] trained an adversarial network to extract bottleneck feature for front-end denoising. Moreover, Bhattacharya et al. [22], [23] borrowed the adversarial training idea from GANs to tackle domain mismatch problems. The aforementioned works found that GANs could be applied to different scenarios of speaker verification and provide a significant performance improvement.

In this paper, we propose a new adversarial strategy against noise interference to extract the noise-robust speaker embedding. Motivated by [11], [12], we first explore the robust capability of the x-vector model by training the x-vector using a mixture dataset including clean data and corrupted data with different types of noise in different signal-to-noise ratios (SNRs). Then a triple-net GAN is proposed to be incorporated in x-vector to form a new framework for the noise-robust speaker embedding extraction, which could further eliminate the noisy disturbance and improve the performance under the noisy conditions. Besides, we have also experimented the proposed framework using two other variants of GANs to further verify the effectiveness of the proposed adversarial strategy in extracting noise-robust speaker embedding.

II. DNN SPEAKER EMBEDDING SYSTEM

In this work, we use the typical x-vector model [8] to extract the utterance-level representation. The whole architecture includes the time delay neural network (TDNN) [24] layers that are operated on the frame level, a statistics pooling layer that aggregates over the frame-level output, and finally the fully-connected (FC) layers on the segment level. And the implementation details are outlined in TABLE I. The network is trained to classify the speaker identities using the conventional cross entropy loss. When the training phase is over, the output layer of the network will be discarded and the x-vectors will be extracted at layer FC2. Rectified Linear

TABLE I
THE TOPOLOGY OF X-VECTOR ACHITECTURE.

Layer	Layer context	Total context	Input × output
TDNN1	[-2, +2]	5	23 × 256
TDNN2	[-2, +2]	9	256 × 512
TDNN3	[-3, +3]	15	512 × 512
TDNN4	{t}	15	512 × 1024
TDNN5	{t}	15	1024 × 1024
Stats pooling	[0,T]	T	1024T × 2048
FC1	{0}	T	2048 × 1024
FC2	{0}	T	1024 × 1024
Softmax	{0}	T	1024 × N

The x-vectors are extracted from layer FC2, and N corresponds to the number of speakers in the training set.

Unit (RELU) is used as non-linear activations for all layers of the network expcept FC2, while the layer FC2 uses the sigmoid function as the non-linear activation. Additionally, batch normalization is used on all layers except the statistics pooling layer. In the verification stage, length normalization and linear discriminant analysis (LDA) [25] transformation will be applied to the x-vectors and then PLDA will be used for scoring.

III. NOISE-ROBUST SPEAKER EMBEDDING

A. Generative Adversarial Networks

The generative adversarial networks (GANs) [18] consist of two parts: a generator (G) that is trained to generate samples indistinguishable from real samples (x) by taking random noise (z) as input, and a discriminator (D) that is trained to determine which distribution the samples obey, the generated data distribution $p_G(z)$ or the real data distribution $p(x)$. In order to get a well-trained generator, the GANs would be trained in an adversarial way by playing a minimax game. The minimax game can be executed with the GANs loss (which means the loss for GAN training) function $V(D,G)$, which could be formulated as follow:

$$\min_G \max_D V(D,G) = E_{x\sim p_{(x)}}[\log D(x)] + E_{z\sim p_G(z)}[\log(1-D(G(z)))] \quad (1)$$

where x is the real sample and z is the random noise input. $G(*)$ and $D(*)$ refer to the output of the generator and discriminator, respectively. The first item $E_{x\sim p(x)}\,[logD(x)]$ represents the mathematical expectation that the discriminator classifies the real samples correctly, and the second item $E_{z\sim p_G(z)}[log(1-D(G(z)))]$ represents the mathematical expectation that the discriminator classifies the generated samples incorrectly.

In the resent years, various variants of GANs have been proposed, each of which has its own particular characteristic. For example, Least Squares GANs (LSGAN) [26] and Wasserstein GANs (WGAN) [27], both of which have the same architecture of GANs but differ in loss function.

B. X-vector Incorporating GANs

We proposed a triple-net adversarial framework to extract the noise-robust speaker embedding as shown in Figure 1.

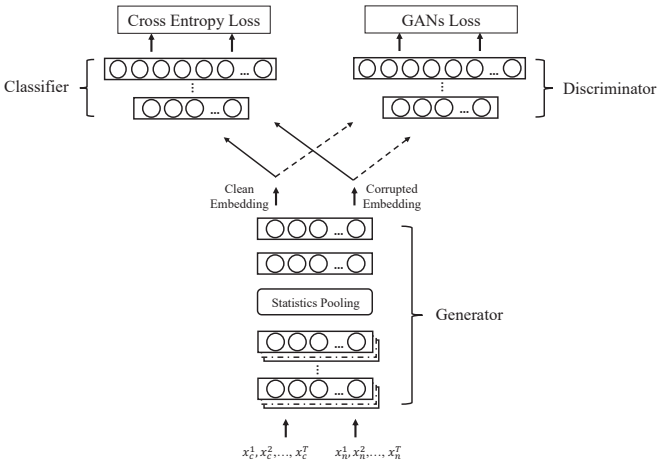

Fig. 1. The diagram of the proposed triple-net adversarial framework. Specifically, a standard x-vector architecture includes generator and classifier while a standard GAN includes generator and discriminator.

This new framework consists of three connected networks: a generator (G), serving as a transformer, that tries to transform clean samples (x_c) and corrupted samples (x_n) to clean embeddings and corrupted embeddings, respectively, a discriminator (D) that tries to determine whether the input embeddings are generated from corrupted samples or clean samples, and a classifier (C) that classifies the embeddings to its corresponding speaker labels. Specically, we train the classifier cooperated with the generator to extract the discriminative speaker representation. Besides, we play a minimax game by training the discriminator to classify the embeddings correctly, and simultaneously training the generator to generate the embeddings to fool the discriminator. Through the training strategy with the proposed adversarial method, the embeddings extracted from the generator maintain the speaker characteristic and simultaneously improve the noise robustness.

For the training, we train the classifier (parametrized by θ_c) using the cross entropy loss:

$$L_C(x) = -\frac{1}{M}\sum_{i=1}^{M} log[f_{softmax}(C(x_i^*))] \quad (2)$$

where M is the batch size, $C(x_i^*)$ is the output of classifier that corresponds to the ground truth of the ith sample in a batch. And $f_{softmax}$ is a softmax function, which can be formulated as follows:

$$f_{softmax}(C(x_i)) = \frac{e^{C(x_i)}}{\sum_j^N e^{C(x_i^j)}} \quad (3)$$

where $x_i^j (j = 1, 2, ..., N)$ means the jth output corrsponding to label j. As to the adversarial training, rather than directly using the minimax loss, we split the optimization into two independent objectives, one for the generator (parametrized by θ_g) and one for the discriminator (parametrized by θ_d). Therefore, we train the generator by $\min_{\theta_g} L_G$ and synchronous-

ly train the discriminator by $\min_{\theta_d} L_D$, which can be formulated as follow:

$$\min_{\theta_g} L_G(x_c, x_n) = \frac{\lambda}{M}\sum_{i=1}^{M} log(1 - D(G(x_n^i))) + L_C(x_c) + L_C(x_n) \quad (4)$$

$$\min_{\theta_d} L_D(x_c, x_n) = -\frac{1}{M}\sum_{i=1}^{M}[log(D(x_c^i)) + log(1 - D(G(x_n^i)))] \quad (5)$$

where λ is a scale parameter. x_c^i means the ith clean sample in a batch and x_n^i means the ith corrupted sample in a batch.

C. Training Algorithm

During training, the generator and the discriminator are competing against each other in an adversarial way. Additionally, the classifier is trained in a straight back-propagation. The complete training pseudo-code of the triple-net GAN shown in Algorithm 1. Pairs of samples (x_c, x_n) are randomly chosen to train the networks using Adam optimizer [28] with learning rates $\alpha_1, \alpha_2, \alpha_3$ for different networks backpropagation respectively. Crucially, we train the generator k times to balance the adversarial training.

Our implementation is realized on the Tensorflow toolkit [29]. In our experiments, the hyperparameters are set as follows: $\alpha_1 = 0.003, \alpha_2 = 0.003, \alpha_3 = 0.003, k = 3, \lambda = 1$.

Algorithm 1: The training procedure of the triple-net GAN

1. Initialize the parameters of generator, discriminator and classifier $\boldsymbol{\theta_g}, \boldsymbol{\theta_d}, \boldsymbol{\theta_c}$. Specifically, we initial the parameters of generator $\boldsymbol{\theta_g}$ from the pre-trained x-vector structure.
2. **repeat**
3. Sample the training data $(\boldsymbol{x_c}, \boldsymbol{x_n})$
4. Update the classifier using Adam optimizer
$$\boldsymbol{\theta_c} \leftarrow \boldsymbol{\theta_c} - \boldsymbol{\alpha_1}\boldsymbol{\nabla}_{\boldsymbol{\theta_c}}\left[\boldsymbol{L_C}\left(\boldsymbol{x_c}\right) + \boldsymbol{L_C}\left(\boldsymbol{x_n}\right)\right]$$
5. Update the discriminator using Adam optimizer:
$$\boldsymbol{\theta_d} \leftarrow \boldsymbol{\theta_d} - \boldsymbol{\alpha_2}\boldsymbol{\nabla}_{\boldsymbol{\theta_d}}\boldsymbol{L_D}\left(\boldsymbol{x_c}, \boldsymbol{x_n}\right)$$
6. **For k** steps **do**
Update the generator using Adam optimizer:
$$\boldsymbol{\theta_g} \leftarrow \boldsymbol{\theta_g} - \boldsymbol{\alpha_3}\boldsymbol{\nabla}_{\boldsymbol{\theta_g}}\boldsymbol{L_G}\left(\boldsymbol{x_c}, \boldsymbol{x_n}\right)$$
End For
7. **until** $\boldsymbol{\theta_g}, \boldsymbol{\theta_d}, \boldsymbol{\theta_c}$ converge.

IV. EXPERIMENTS

A. Dataset and Experimental Setting

To evaluate the effective performance of the proposed method in the noisy environments, text-independent speaker verication (SV) experiments are conducted based on Librispeech [30]. In the experiments, the train-clean-500 part of Librispeech is used as a training dataset which contains about 148,688 utterances from 1,166 speakers and the test-clean part of Librispeech is used as a test dataset, in which 15 utterances of each speaker are selected as enrollment utterances and the remains are used for verification (about 80,800 trials).

We have made a noise corrupted version of the training data and the test data mentioned above by artificially adding different types of noise at different SNR levels. Specifically, the corrupted utterances for training are made by adding one of the five noise types (White, Babble, Mensa, Cafeteria, Callcener)[1] randomly on the SNR levels of 10dB or 20dB. Additionally, we also have made a mixture dataset, in which five out of six samples in clean training data are added by the random noise in the same way mentioned above. However, for the speaker verification the corrupted utterances are obtained by adding one of the five noise types on the SNR levels of 0dB, 5dB, 10dB, 15dB and 20dB, respectively.

All audios are converted to 23-dimensional MFCCs with a frame-length of 25ms and a frame shift of 10ms. Then, a frame-level energy-based voice activity detector (VAD) is conducted to the spectral coefficients. During the training process, we randomly sample 2 seconds from each recording. Since the average duration of utterances in the training dataset is about 12 seconds, we repeat six times to form an epoch. Specifically, we constitute pairs by taking the clean utterance and its corrupted counterpart as a pair.

Since the noise-robust embedding architecture is based on the x-vector framework, the implementation details of the generator are outlined in TALBLE I except the softmax layer. The classifier is one fully-connected layer with N nodes, which corresponds to the number of speakers in the training data, while the discriminator is also one fully-connected layer but with only two nodes corresponding to the binary lables for the clean data and the corrupted data.

B. Results and Analysis

Firstly, we investigate the performance of baseline system (Baseline), which is the standard x-vector system trained with the clean data. As shown in TABLE II, under the clean condition the x-vector would achieve a competitive result, but the noisy disturbance could sharply degrade the performance of x-vector, especially the White noise. Then, we explore the robustness of x-vector by training the network using the mixture dataset (MIX). From TABLE II, it could be found that the MIX system dramatically obtains the better performance under noisy environments. Though the performance might be slightly degraded under the clean condition since the training process is not condition-specfic comparing with the Baseline, but that is still comparable with that of the baseline system.

Thereafter, we investigate the capability of triple-net GAN (TNGAN) for extracting the noise-robust speaker embedding. Specially, we initialize the parameters of generator using the

[1]White and Babble were collected by Guoning Hu, and could be downloaded at http://web.cse.ohio-state.edu/pnl. Besides, Cafeteria Noise, Callcener, and Mensa were provided by HUAWEI TECHNOLOGIES CO., LTD.

TABLE II
EER(%) OF THE SV SYSTEMS FOR DIFFERENT NOISE TYPES AND SNRS.

Noise	SNR(dB)	Baseline	MIX	TNGAN
-	Clean	**1.39**	3.02	2.82
White	00	38.12	14.41	**11.63**
	05	33.07	8.27	**6.73**
	10	27.18	6.58	**5.30**
	15	22.52	5.64	**4.65**
	20	17.72	4.85	**4.11**
	Mean	27.72	7.95	**6.48**
Babble	00	31.93	12.57	**10.40**
	05	25.84	6.04	**5.64**
	10	19.50	4.55	**4.21**
	15	14.60	4.11	**3.71**
	20	9.65	3.76	**3.61**
	Mean	20.30	6.21	**5.52**
Cafeteria	00	29.9	8.71	**7.67**
	05	24.31	5.64	**4.90**
	10	18.71	4.41	**4.06**
	15	13.86	4.01	**3.71**
	20	9.85	3.81	**3.42**
	Mean	19.33	5.32	**4.75**
Callcenter	00	28.32	8.86	**7.18**
	05	22.28	5.15	**4.65**
	10	16.93	4.06	**3.86**
	15	11.19	3.81	**3.52**
	20	7.08	3.56	**3.37**
	Mean	17.16	5.09	**4.51**
Mensa	00	31.98	10.50	**9.21**
	05	26.63	6.04	**5.20**
	10	20.69	4.85	**4.46**
	15	15.25	4.31	**3.91**
	20	10.45	4.11	**3.91**
	Mean	21.00	5.96	**5.34**

TABLE III
EER(%) OF THE DIFFERENT GAN-BASED SV SYSTEMS FOR DIFFERENT NOISE TYPES AND SNRS.

Noise	SNR(dB)	TNGAN	TNLSGAN	TNWGAN
-	Clean	**2.82**	2.87	2.87
White	00	11.63	**11.53**	11.58
	05	**6.73**	6.78	6.88
	10	**5.30**	5.40	**5.30**
	15	4.65	**4.60**	4.65
	20	**4.11**	**4.11**	**4.11**
	Mean	**6.48**	**6.48**	6.50
Babble	00	**10.4**	10.54	10.54
	05	5.64	5.69	**5.59**
	10	**4.21**	4.26	4.31
	15	**3.71**	3.81	3.81
	20	3.61	**3.52**	3.56
	Mean	**5.52**	5.56	5.56
Cafeteria	00	7.67	**7.62**	**7.62**
	05	**4.90**	5.00	**4.90**
	10	4.06	4.06	**3.96**
	15	**3.71**	3.81	**3.71**
	20	3.42	**3.32**	3.47
	Mean	4.75	4.76	**4.73**
Callcenter	00	7.18	7.33	**7.08**
	05	4.65	**4.60**	**4.60**
	10	**3.86**	3.91	**3.86**
	15	3.52	3.47	**3.42**
	20	3.37	3.27	**3.27**
	Mean	4.51	4.51	**4.45**
Mensa	00	9.21	**8.86**	8.96
	05	5.20	5.25	5.25
	10	**4.46**	**4.46**	4.51
	15	**3.91**	**3.91**	**3.91**
	20	3.91	**3.86**	**3.86**
	Mean	5.34	**5.27**	5.30

parameters of the pre-trained MIX system. Comparing the results on column 4 and 5 of TABLE II, we can observe a significant relative reduction in equal error rate (EER) across different SNR levels. The relative reduction of average EERs comparing with the MIX system are about 18.5% on White noise, 11.1% on Babble noise, 10.7% on Cafeteria noise, 11.4% on Callcenter noise and 10.4% on Mensa noise, which have shown the effectiveness of the proposed generative adversarial strategy on extracting a noise-robust embedding.

Besides, we further verified the proposed generative adversarial strategy by using two different variants of triple-net GAN, triple-net LSGAN (TNLSGAN) and triple-net WGAN (TNWGAN) respectively, which may learn different aspects of feature space. The results shown in TABLE III reflect that the different types of triple-net GAN could achieve the better performance in different SNR levels on different noises respectively, and all of which outperformed the Baseline and the MIX systems. It is definitely promising that the proposed triple-net adversarial framework could improve the performance under the noisy environments.

V. CONCLUSION

This paper proposed an adversarial framework by incorporating GANs with x-vector framework for noise-robust speaker embedding extraction. We demonstrated that the performance of speaker verification system degrade sharply under noisy environments and proposed the adversarial framework to deal with such issue. Results on Librispeech showed that our proposed method could achieve a significant performance improvement under the noisy environments. Additionally, we further verified the capability of the proposed triple-net adversarial framework for extracting noise-robust speaker embedding by using two another variants of GANs.

VI. ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China (Grant No.61876160)

REFERENCES

[1] Lecun Y, Bengio Y, Hinton G. Deep learning.[J]. Nature, 2015, 521(7553):436.
[2] Hinton G, Deng L, Yu D, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups[J]. IEEE Signal processing magazine, 2012, 29(6): 82-97.
[3] Lei Y, Scheffer N, Ferrer L, et al. A novel scheme for speaker recognition using a phonetically-aware deep neural network[C]// IEEE International Conference on Acoustics. 2014.
[4] Wang Y, Skerry-Ryan R J, Stanton D, et al. Tacotron: Towards end-to-end speech synthesis[J]. arXiv preprint arXiv:1703.10135, 2017.
[5] Variani E, Lei X, Mcdermott E, et al. Deep neural networks for small footprint text-dependent speaker verification[C]// IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2014:4052-4056.
[6] Heigold G, Moreno I, Bengio S, et al. End-to-end text-dependent speaker verification[C]//Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on. IEEE, 2016: 5115-5119.
[7] Chen K, Salman A. Learning speaker-specific characteristics with a deep neural architecture[J]. IEEE Transactions on Neural Networks, 2011, 22(11):1744-1756.

[8] Snyder D, Garcia-Romero D, Povey D, et al. Deep neural network embeddings for text-independent speaker verification[C]//Proc. Interspeech. 2017: 999-1003.
[9] Prince S J D, Elder J H. Probabilistic linear discriminant analysis for inferences about identity[C]//Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on. IEEE, 2007: 1-8.
[10] Dehak N, Kenny P J, Dehak R, et al. Front-end factor analysis for speaker verification[J]. IEEE Transactions on Audio, Speech, and Language Processing, 2011, 19(4): 788-798.
[11] Lei Y, Burget L, Scheffer N. A noise robust i-vector extractor using vector taylor series for speaker recognition[C]// IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2013:6788-6791.
[12] Lei Y , Burget L , Ferrer L , et al. Towards noise-robust speaker recognition using probabilistic linear discriminant analysis[C]// IEEE International Conference on Acoustics. IEEE, 2012.
[13] Tan Z , Mak M W , Mak B , et al. Denoised senone i-vectors for robust speaker verification[J]. IEEE/ACM Transactions on Audio Speech & Language Processing, 2018, 26(4):820-830.
[14] Li N , Mak M W , Chien J T . Deep neural network driven mixture of PLDA for robust i-vector speaker verification[C]// Spoken Language Technology Workshop. IEEE, 2017.
[15] Mekonnen B W , Dufera B D . Noise robust speaker verification using GMM-UBM multi-condition training[C]// Africon. IEEE, 2015.
[16] Reynolds D A , Quatieri T F , Dunn R B . Speaker verification using adapted gaussian mixture models[J]. Digital Signal Processing, 2000, 10(1-3):19-41.
[17] Snyder, David, et al. "X-vectors: Robust DNN embeddings for speaker recognition." 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018.
[18] Goodfellow I J, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[C]// International Conference on Neural Information Processing Systems. MIT Press, 2014:2672-2680.
[19] Zhang J, Inoue N, Shinoda K. I-vector transformation using conditional generative adversarial networks for short utterance speaker verification[J]. arXiv preprint arXiv:1804.00290, 2018.
[20] Ding W, He L. MTGAN: Speaker verification through multitasking triplet generative adversarial networks[J]. arXiv preprint arXiv:1803.09059, 2018.
[21] Yu H, Tan Z H, Ma Z, et al. Adversarial network bottleneck features for noise robust speaker verification[J]. 2017:1492-1496.
[22] Bhattacharya G, Alam J, Kenny P. Adapting end-to-end neural speaker verification to new languages and recording conditions with adversarial training[J]. arXiv preprint arXiv:1811.03055, 2018.
[23] Bhattacharya G, Monteiro J, Alam J, et al. Generative adversarial speaker embedding networks for domain robust end-to-end speaker verification[J]. arXiv preprint arXiv:1811.03063, 2018.
[24] Peddinti V, Povey D, Khudanpur S. A time delay neural network architecture for efficient modeling of long temporal contexts[C]//Sixteenth Annual Conference of the International Speech Communication Association. 2015.
[25] Xanthopoulos P, Pardalos P M, Trafalis T B. Linear discriminant analysis[J]. Chicago, 2015, 3(6):27-33.
[26] Mao X, Li Q, Xie H, et al. Least squares generative adversarial networks[C]//Computer Vision (ICCV), 2017 IEEE International Conference on. IEEE, 2017: 2813-2821.
[27] Arjovsky M, Chintala S, Bottou L. Wasserstein gan[J]. arXiv preprint arXiv:1701.07875, 2017.
[28] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.
[29] Abadi M, Agarwal A, Barham P, et al. TensorFlow: large-scale machine learning on heterogeneous distributed systems. arXiv preprint (2016)[J]. arXiv preprint arXiv:1603.04467.
[30] Panayotov V, Chen G, Povey D, et al. Librispeech: An ASR corpus based on public domain audio books[C]// IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2015:5206-5210.