GENERATIVE ADVERSARIAL NETWORKS BASED DATA AUGMENTATION FOR NOISE ROBUST SPEECH RECOGNITION

Hu Hu, Tian Tan, Yanmin Qian[†]

SpeechLab, Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
{mihawk@sjtu.edu.cn, tantian@sjtu.edu.cn, yanminqian@tencent.com}

ABSTRACT

Data augmentation is an effective method to increase the size of training data and reduce the mismatch between training and testing for noise robust speech recognition. Different from the traditional approaches by directly adding noise to the original waveform, in this work we utilize generative adversarial networks (GAN) for data generation to improve speech recognition under noise conditions. With this method, the generated speech samples are based on spectrum feature level and produced frame by frame without dependence among them, and the augmented data has no true labels. Then to effectively use these untranscribed augmented data, an unsupervised learning framework is designed for acoustic modeling. The proposed GAN-based data augmentation approach is evaluated on Aurora4. The experimental results show that a relative $\sim$7.0% WER reduction can be obtained by the proposed approach upon an advanced acoustic model.

Index Terms— robust speech recognition, very deep convolutional neural network, data augmentation, generative adversarial networks, unsupervised learning

1. INTRODUCTION

In recent years we have witnessed significant progress in automatic speech recognition (ASR) due to the introduction of deep learning [1, 2, 3]. Many previous studies applied deep neural network (DNN) based models for acoustic modeling. They showed promising performance and reduced the word error rate (WER) a lot when compared with the conventional GMM model. Nevertheless, these systems still do not work well when processing speech in noisy environments, such as scenarios with additive noise, channel distortion and reverberation [4, 5, 6, 7].

The main problem for noisy speech recognition is the mismatch between the training and testing, due to that the number of noise types in real scenarios are so large that it is impractical to collect enough data to cover all conditions for real applications. Thus in order to improve the robustness of ASR systems, data augmentation is commonly utilized to enlarge the noisy training data and reduce the mismatch in applications. For example several recent studies tried to increase the quantity of training data for far-filed speech recognition [8, 9]. The basic idea in these two works is generating extra data via directly adding the simulated noises to the original speech waveform. They can obtain significant improvements on the related test set. However, there are also several limitations in this approach: the quantity of the generated data size is also dependent on the simulated noise types; directly adding additive or convolutional noise to waveform artificially may cause another internal representations mismatch between the feature levels of generated and real data. Thus, the new data augmentation method is demanded.

Generative adversarial networks (GAN) have attracted a lot of interests in computer vision communities [10, 11, 12]. It can learn generative models via adversarial training, which produces samples from the real data distribution. Based on the basic GAN, the work in [13, 14] proposed Wasserstein GAN (WGAN) to further improve the loss function and training method, which can achieve a better performance. In speech processing, GAN has been preliminarily applied in some tasks, such as speech synthesis [15, 16], voice conversion [17, 18], speech enhancement [19], spoken language identification [20] and even acoustic scene classification [21]. However there is still limited work for speech recognition.

In this work, we propose a new data augmentation strategy by utilizing generative adversarial networks to improve the noise robust speech recognition. The basic acoustic model we use is an advanced very deep convolutional neural network (VDCNN) [22]. Based on the input feature map of VDCNN, we use GAN to generate extra feature maps to enlarge the noisy training data. Furthermore, an unsupervised learning framework is developed to use the unlabeled augmented data in an effective mode for acoustic modeling, which can finally improve the system performance. The experiments on Aurora4 show that the system performance can be improved significantly by the proposed GAN-based data augmentation strategy.

The rest of this paper is organized as follows. Section 2 briefly introduces the basic generative adversarial networks and Wasserstein generative adversarial networks. The new proposed GAN-based data augmentation approach is described in Section 3. An unsupervised learning with the unlabeled generated data is presented in Section 4. Section 5 shows the experimental results and analysis, and Section 6 gives the conclusions.

2. GENERATIVE ADVERSARIAL NETWORK

Generative adversarial network (GAN) was firstly introduced by Goodfellow et al. in [10] as a powerful generative model for a wide range of applications. The basic idea of GAN is to set up a game between two players, i.e. a generator G and a discriminator D. The discriminator performs classification between the real samples

[†] Yanmin Qian is the corresponding author and now he is with Tencent AI Lab, Tencent, Bellevue, WA, USA.

This work was partly supported by the China NSFC projects (No. U1736202 and No. 61603252), and the Shanghai Sailing Program No. 16YF1405300. Experiments have been carried out on the PI supercomputer at Shanghai Jiao Tong University.

978-1-5386-4658-8/18/$31.00 ©2018 IEEE

and fake samples. The generator produces samples from a data distribution, which is usually a low dimensional random noise. The produced samples are then passed into the discriminator to determine their similarity with the real data. The generator is optimized to fool the discriminator while the discriminator is trained to distinguish the fake data from the real data. More specifically, the game between the generator G and the discriminator D is formulated as a two-player minimax game with the following cross entropy:

$$\min_{G}\max_{D}\mathbf{E}_{x\sim P_r}[log(D(x))]+\mathbf{E}_{z\sim P_g}[log(1-D(G(z)))] \quad (1)$$

where Pr is the real data distribution, and P_g is the generated data distribution. $D(x)$ represents the probability that x comes from the real data. z is the noise variable as the input to G.

More recently, researchers find that the traditional loss function shown above is potentially not continuous and thus cannot provide a usable gradient for the generator [13, 14]. Thus, they proposed Wasserstein distance to measure the difference between these two distributions, and D and G are trained by the following expression:

$$\min_{G}\max_{D\in L}\mathbf{E}_{x\sim P_r}[D(x)]-\mathbf{E}_{z\sim P_g}[D(G(z))] \quad (2)$$

where L is the set of 1-Lipschitz functions introduced by WGAN to restrict the discriminator. The Wasserstein distance has the desirable property of being continuous and differentiable almost everywhere under mild assumptions. Thus, WGAN is a more stable framework to be applied in many scenarios.

3. GAN FOR DATA AUGMENTAION

As described above, most of the previous data augmentation work adds various types of noise to the waveform directly, which has some drawbacks. Some methods start to use generative models for data generation [23, 21], but they are based on the original waveform level or generate samples on the whole sequence with related labels.

This work utilizes the GAN model for data augmentation. The basic unit we choose to generate data is the feature map on speech spectrum. Thus it is performed on the speech feature level, such as FBANK, to generate samples frame by frame. Given a K-dimension FBANK feature, the context expansion is applied with N frames on each side, so we can get a $(2N+1)\times K$-dimension feature map in the time-frequency domain, which is finally used as the real data input for the discriminator. In our experiments, we set $K=64$ and $N=8$, so 17×64 feature map is formed. The output of GAN is also the feature map of the same size, which will be utilized for acoustic modeling in ASR. It is noted that due to the randomness of the noise input for the generator and our frame-level data generation strategy, the labels are unknown for the generated samples (feature maps), and all the generated samples are independent from each other.

According to previous work on GAN, the structure configuration and training setting of GAN are very important for the model optimization. The configuration of our GAN structure is illustrated in Figure 1. For the discriminator, there are three convolutional layers, followed with two fully connected layers to classify the real and fake data. For the generator, similar to the discriminator, there are two fully connected layers to transfer the input random noise, and then the generator uses three transposed convolutional layers to generate the feature maps. After each convolutional, transposed convolutional and fully connected layer, batch normalization is adopted. The Leaky ReLU is applied in both discriminator and generator. The noise input for the generator is randomly sampled from a Gauss distribution. As described in Section 2, we use the WGAN training framework in this work to get the more stable training process.

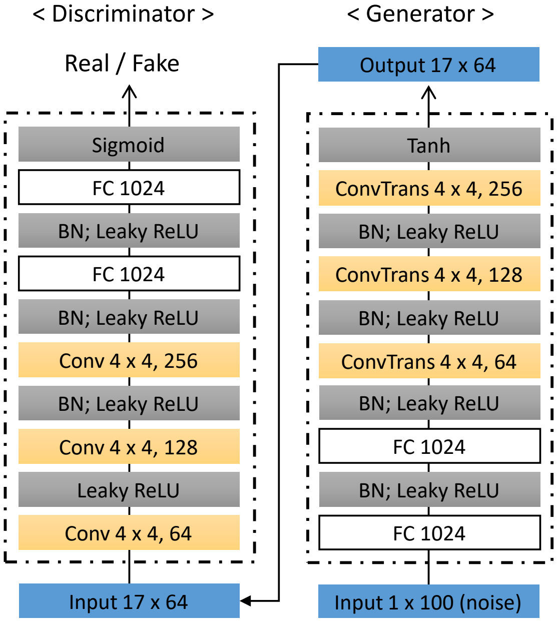

Fig. 1. The architecture of the proposed generator and discriminator in the GAN-based data augmentation for ASR. **Conv** means convolutional layer, **ConvTrans** means transposed convolutional layer, **FC** means fully connected layer, and **BN** means batch normalization. The model configuration, such as $[4\times4,64]$ indicates that the layer uses a 4×4 filter and the output contains 64 feature maps.

4. UNSUPERVISED LEARNING BY AUGMENTED DATA

Considering that each output feature map of GAN is generated from a random noise vector, it is hard for us to get the true label for the generated feature map. Thus an unsupervised learning strategy is developed to utilize these augmented data. Assuming that the distributions between the original data and generated data are similar from the well-trained GAN model, the augmented data from GAN can be firstly processed by the original acoustic model to collect the soft labels (the corresponding posterior probabilities). Once the soft label of each feature map is obtained, the original data can be pooled with the new generated data to train a new acoustic model. More specifically, the whole training procedure is shown in Algorithm 1.

The Kullback–Leibler (KL) divergence between the acoustic model output distribution and the related labels is used as the training criteria. In our experiments, minimizing the KL divergence is equivalent to maximizing the following expression:

$$\begin{aligned} J = &\sum_{\mathbf{o}_t\in Dgen}\sum_{s} p_A(s|\mathbf{o}_t)\log p_B(s|\mathbf{o}_t) \\ &+\sum_{\mathbf{o}_t\in Dorig}\sum_{s} p_{ref}\log p_B(s|\mathbf{o}_t) \quad (3)\end{aligned}$$

where $\mathbf{o}_t$ is the input feature and and s is the acoustic state. D_{orig} and D_{gen} are the original dataset and generated dataset respectively.

Algorithm 1 Unsupervised learning with augmented data

```
Use the original data $Dorig$ to train an original acoustic model $A$ for ASR and GAN model $N$ for data augmentation.
Use GAN model $N$ to generate extra dataset $Dgen$.
Use the original acoustic model $A$ to get the soft label for each augmented feature map frame by frame.
Pool the original dataset $Dorig$ with hard labels and augmented dataset $Dgen$ with soft labels to train a new acoustic model $B$.
```

p_{ref} is the reference alignment for the original transcribed data, which is the hard label. The posterior distributions of the acoustic model A and B are denoted as $p_A(y|\mathbf{o}_t)$ and $p_B(y|\mathbf{o}_t)$, where $p_A(y|\mathbf{o}_t)$ is the soft label, i.e. the posterior generated by the original acoustic model A. This approach allows us to utilize the large quantity of untranscribed augmented data more effectively. The architecture is illustrated in Figure 2.

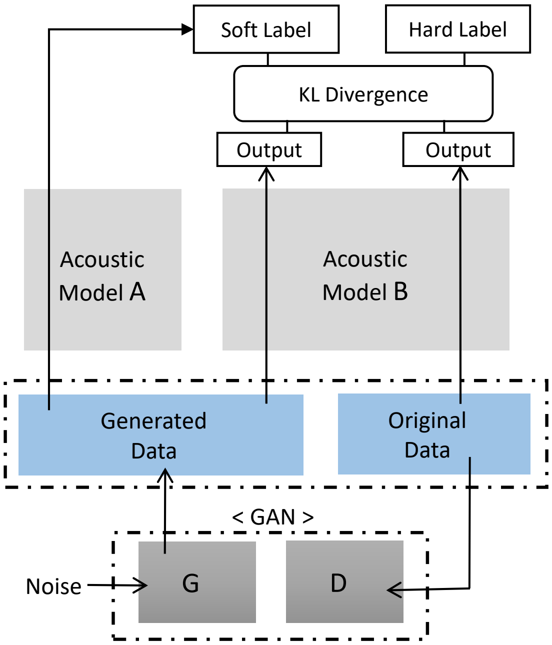

Fig. 2. The proposed unsupervised learning architecture with GAN-based data augmentation for acoustic modeling.

5. EXPERIMENTS

5.1. Experimental Setup

The proposed approach is evaluated and compared on the standard Aurora4 task, which has multiple additive noise conditions as well as channel mismatch. The Aurora4 task is a medium vocabulary speech recognition task based on the Wall Street Journal (WSJ0) corpus [24]. It contains 16 kHz speech data in the presence of additive noises and linear convolutional channel distortions, which are introduced synthetically to clean speech from WSJ0. The multi-condition training set with 7138 utterances from 83 speakers includes a combination of clean speech and speech corrupted by one of six different noises at 10-20 dB SNR. Half data is from the primary Sennheiser microphone and the other is from the secondary microphones. As for the training data, the test data is generated using the same types of noise and microphones. Test data can be grouped into 4 subsets: clean, noisy, clean with channel distortion, and noisy with channel distortion, which will be referred to as A, B, C, and D, respectively.

Gaussian mixture model based hidden Markov models (GMM-HMMs) are first built with Kaldi [25] using the standard recipe, consisting of 3K clustered states trained using maximum likelihood estimation with the standard Kaldi MFCC-LDA-MLLT-FMLLR features. After the GMM-HMM training, a forced-alignment is performed to get the state level labels. All the neural networks based acoustic models for speech recognition are built using CNTK [26] in this work. They were trained using cross entropy (CE) criterion with stochastic gradient descent (SGD) based back propagation (BP) algorithm. The task-standard WSJ0 bigram with 5K-word dictionary is used for decoding, and the standard testing pipelines in the Kaldi recipes are used for decoding and scoring. Our previous proposed very deep convolutional neural network (VDCNN) is used as the acoustic model for all setups in this work [22], which consists of ten convolutional layers and four fully connected layers. More details about the VDCNN model and experimental setup on acoustic modeling can be referred from [22].

All the GAN models for data augmentation used here are implemented with PyTorch [27]. The networks are trained using RMSProp, and the mini-batch size is set to 64. Batch normalization is used after the convolutional or transposed convolutional layers. We use Leaky ReLU in both discriminator and generator, and the negative slope is 0.2. The learning rate is set to 0.00005 in the model optimization for both discriminator and generator. During the training process, for each mini-batch data, the discriminator D is updated 5 times then followed one time update on the generator G, and the maximum training epoch is set to 20 for the model optimization.

5.2. Evaluation on Aurora4

In our experiments, VDCNN acoustic model based baseline is firstly built, which has shown great noise robustness in ASR [22], and the results are shown in the first line of Table 1[1].

For data augmentation using GAN, total 60-hour speech data is newly generated, and then the generated data is pooled with the original Aurora4 data to build the acoustic model. The results are shown as the second line of Table 1. It is observed that the GAN-based data augmentation can still obtain a significant improvement upon the strong acoustic model VDCNN. There is a relative ~6.0% WER reduction compared to the system only using the original noisy training data. Another interesting finding is that although no noise type is appointed at the generation stage, most of the gain is from the subset D with both additive noise and channel distortion. This observation further demonstrates the effectiveness of the proposed GAN-based data augmentation for noise robust speech recognition.

For the better comparison, we also perform the data generation using the normal mode: directly adding all types of noise on the original clean speech waveform manually. For Aurora4, we also use the six noise types from the corpus to generate 60-hour noisy speech data for acoustic modeling. In addition, we further pool all the augmented data from these two different approaches, i.e. using

[1] This performance is slightly worse than our previous number in [22] (9.02 vs. 8.81), since the different CNTK versions are used here.

120-hour generated data, for acoustic modeling. The related results are shown as the last two lines of Table 1. We see that the traditional data augmentation with a manual mode can indeed get a gain, but it is obviously smaller than that from the GAN-based approach. Moreover, the manual mode is more easier to obtain the biased performance on some conditions and gets degradations on the others. In addition, the generated data from different methods seems have their own properties. Combining the generated data from two strategies achieves another additional improvement.

Table 1. WER (%) comparison of different training data. `Manual` means directly adding noise to the original speech waveform manually, and `GAN` means the new proposed GAN-based data augmentation

Data	A	B	C	D	AVG
`original`	3.62	5.81	5.12	13.77	9.02
`+GAN`	3.34	5.70	4.93	12.79	8.51
`+Manual`	4.04	6.37	6.35	12.52	8.84
`+GAN & Manual`	3.10	5.52	4.95	12.65	8.37

Then different augmented data sizes from the same GAN model are compared, and the results are illustrated in Table 2. It is observed that increasing the augmented data size from GAN indeed can gradually improve the system, but the performance difference is not very large. The performance of system using 15hr augmented data even can approaches that using 90hr augmented data (less than absolute 0.1% difference on averaged WER on Aurora4). This demonstrates the efficiency and effectiveness of using GAN to do the data generation.

Table 2. WER (%) comparison of different training data sizes generated by GAN.

Data size	A	B	C	D	AVG
`original`	3.62	5.81	5.12	13.77	9.02
15h	3.31	5.53	4.93	13.04	8.55
30h	3.36	5.60	4.99	12.91	8.53
60h	3.34	5.70	4.93	12.79	8.51
90h	3.36	5.68	5.01	12.68	8.47

5.3. Visualization and analysis on generated feature maps

In order to better understand generated training samples from the GAN model, the produced feature map examples (17×64) are plotted and illustrated in Figure 3. It shows the comparison of different feature maps generated by the GAN model on different training stages. Four random noise vectors are used for data generation and each row corresponds the feature maps generated from the same random noise vector input but on different training epochs. For the better comparison between the generated data and real speech, several real feature maps from the original noisy corpus are also illustrated in the right part of Figure 3.

The illustration shows that all the generated feature maps indeed look like the real speech spectrum very much. With the training process proceeds, the speech patterns can be observed more obviously, which means that the quality of the generated data is gradually improved with more training epochs. Doing the comparison within the samples from different noise inputs for generator, the randomness and difference is obvious significant. This property of GAN-based data augmentation can enable us to produce noisy data with more random patterns for robust speech recognition.

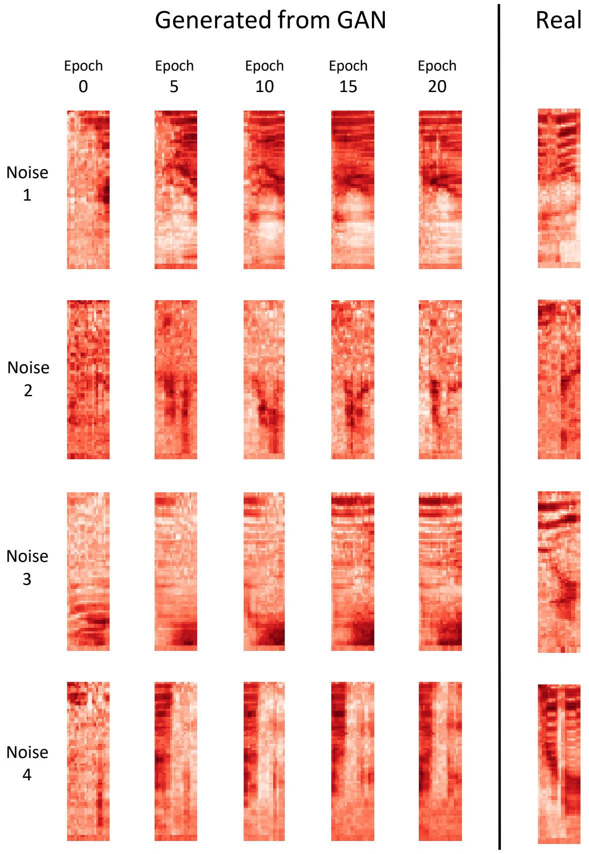

Fig. 3. Generated feature maps from GAN model on different training epochs. Four random noise inputs are used for GAN to generate four examples. Each row corresponds the feature maps generated from the same noise input but on different training epochs. The right part is the selected real feature maps from the original real noisy data.

6. CONCLUSION AND FUTURE WORK

In this paper we propose a new framework on data augmentation for noise robust speech recognition. Different from most conventional approaches by directly adding noise to the original waveform, the generative adversarial network is utilized. The augmented data from GAN is based on spectrum feature level and generated frame by frame (one frame corresponds one feature map). There is no dependence among the generated samples, even no true labels existing for them. Then an unsupervised learning strategy is designed to utilize these untranscribed augmented data in an effective mode. The proposed framework is evaluated on Aurora4 and shows more than a relatively ~7.0% WER reduction on this noisy ASR task.

This work is the first attempt to explore GAN on data augmentation for speech recognition. In our future work, we will extend and evaluate the proposed method on other noisy scenarios, such as the far-field scenario with reverberation.

7. REFERENCES

[1] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, 2012.

[2] Frank Seide, Gang Li, and Dong Yu, “Conversational speech transcription using context-dependent deep neural networks,” in *Proc. INTERSPEECH*, 2011.

[3] George E. Dahl, Dong Yu, Li Deng, and Alex Acero, “Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition,” *TASLP*, vol. 20, pp. 30–42, 2012.

[4] Yongqiang Wang and M. J. F. Gales, “Speaker and noise factorization for robust speech recognition,” *TASLP*, vol. 20, no. 7, pp. 2149–2158, 2012.

[5] D. Pearce, “Aurora working group: Dsr front-end lvcsr evaluation au/384/02,” *Ph.D. dissertation, Mississippi State Univ.*, 2002.

[6] T. Hain, L. Burget, J. Dines, P. N. Garner, F. Grezl, A. E. Hannani, M. Huijbregts, M. Karafiat, M. Lincoln, and V. Wan, “Transcribing meetings with the amida systems,” *TASLP*, vol. 20, no. 2, pp. 486–498, Feb 2012.

[7] Yanmin Qian, Tian Tan, Hu Hu, and Qi Liu, “Noise robust speech recognition on aurora4 by humans and machines,” in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2018.

[8] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L Seltzer, and Sanjeev Khudanpur, “A study on data augmentation of reverberant speech for robust speech recognition,” in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2017, pp. 5220–5224.

[9] Chanwoo Kim, Ananya Misra, Kean Chin, Thad Hughes, Arun Narayanan, Tara N. Sainath, and Michiel Bacchiani, “Generation of large-scale simulated utterances in virtual rooms to train deep-neural networks for far-field speech recognition in google home,” in *INTERSPEECH*, 2017, pp. 379–383.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems 27*, pp. 2672–2680. 2014.

[11] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros, “Image-to-image translation with conditional adversarial networks,” *arxiv*, 2016.

[12] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen, “Improved techniques for training gans,” *CoRR*, vol. abs/1606.03498, 2016.

[13] Martin Arjovsky and Léon Bottou, “Towards principled methods for training generative adversarial networks,” *arXiv preprint arXiv:1701.04862*, 2017.

[14] Martín Arjovsky, Soumith Chintala, and Léon Bottou, “Wasserstein GAN,” *CoRR*, vol. abs/1701.07875, 2017.

[15] T. Kaneko, H. Kameoka, N. Hojo, Y. Ijima, K. Hiramatsu, and K. Kashino, “Generative adversarial network-based postfilter for statistical parametric speech synthesis,” in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2017, pp. 4910–4914.

[16] Yuki Saito, Shinnosuke Takamichi, and Hiroshi Saruwatari, “Statistical parametric speech synthesis incorporating generative adversarial networks,” *TASLP*, 2017.

[17] Chin-Cheng Hsu, Hsin-Te Hwang, Yi-Chiao Wu, Yu Tsao, and Hsin-Min Wang, “Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks,” *CoRR*, vol. abs/1704.00849, 2017.

[18] Takuhiro Kaneko, Hirokazu Kameoka, Nobukatsu Hojo, Yusuke Ijima, Kaoru Hiramatsu, and Kunio Kashino, “Generative adversarial network-based postfilter for statistical parametric speech synthesis,” in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2017, pp. 4910–4914.

[19] Santiago Pascual, Antonio Bonafonte, and Joan Serrà, “SEGAN: speech enhancement generative adversarial network,” *CoRR*, 2017.

[20] Sheng Li Peng Shen, Xugang Lu and Hisashi Kawai, “Conditional generative adversarial nets classifier for spoken language identification,” in *INTERSPEECH*, 2017.

[21] David Han Seongkyu Mun, Sangwook Park and Hanseok Ko, “Generative adversarial network based acoustic scene training set augmentation and selection using svm hyper-plane,” in *Detection and Classification of Acoustic Scenes and Events*, 2017.

[22] Yanmin Qian, Mengxiao Bi, Tian Tan, and Kai Yu, “Very deep convolutional neural networks for noise robust speech recognition,” *TASLP*, vol. 24, no. 12, pp. 2263–2276, 2016.

[23] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alexander Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, “Wavenet: A generative model for raw audio,” in *Arxiv*, 2016.

[24] David Pearce and J Picone, “Aurora working group: Dsr front end lvcsr evaluation au/384/02,” *Inst. for Signal & Inform. Process., Mississippi State Univ., Tech. Rep*, 2002.

[25] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, “The kaldi speech recognition toolkit,” in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, 2011.

[26] Dong Yu, Adam Eversole, Mike Seltzer, Kaisheng Yao, Oleksii Kuchaiev, Yu Zhang, Frank Seide, Zhiheng Huang, Brian Guenter, Huaming Wang, Jasha Droppo, Geoffrey Zweig, Chris Rossbach, Jie Gao, Andreas Stolcke, Jon Currey, Malcolm Slaney, Guoguo Chen, Amit Agarwal, Chris Basoglu, Marko Padmilac, Alexey Kamenev, Vladimir Ivanov, Scott Cypher, Hari Parthasarathi, Bhaskar Mitra, Baolin Peng, and Xuedong Huang, “An introduction to computational networks and the computational network toolkit,” Tech. Rep., October 2014.

[27] Adam Paszke, Sam Gross, and Soumith Chintala, “Pytorch,” 2017.