Adversarial Training On General Voice Conversion

Hao-Chun Yang
Electrical Engineering Department
chadyoungy@gmail.com

Gao-Yi Chao
Electrical Engineering Department
ooxx15935720@gmail.com

Hui-Ting Hong
Electrical Engineering Department
w102060018w@gmail.com

Kun-Chieh Hsu
Material Science Department
s103031111@outlook.com

Abstract

In this project, we focus on dealing with the voice conversion problem. We apply a VAE-GAN with DNN structure which is inspired by [3]. It can solve the problem of voice conversion on non-parallel data using the idea of the generative model. In the following, we will first describe how people dealt with voice conversion problem previously and then introduce our approach. Later talk about our experiment result and finally give a brief conclusion.

Index Terms: voice conversion, speech synthesis, improved Wasserstein-generative adversarial network, variational autoencoder

1. Introduction

Voice conversion(VC) is a classical task which usually aims at transforming the voice of a source speaker into that of a target speaker while preserving the linguistic content[2]. In this project, we are going to deal with this classical task with unaligned corpora in conditional variational autoencoder with adversarial training.

Many techniques have been developed for VC, such as Gaussian mixture models, Hidden Markov models, Gaussian regression, RNN. One big issue in the training of the speech conversion system is data, which prefers parallel and pairwise data. Since the inter-speaker speech alignment could be a problem for machines to learn speech style among different individuals. Hence, in this work, we apply adversarial training architecture for speech style transformation.

2. Related Work

2.1. HMM

The problem for generating the speech sequence $O = (o_1, o_2, ..., o_T)$ from the HMM is to obtain a speech sequence which maximizes $P(O|, T)$ with respect to O,

$$O^* = \arg\max_O \sum_{all\ q} P(O, q|\lambda, T) \tag{1}$$

Much detailed formulations are shown in [11].

In Nose and Kobayashi's work[8], they extracted the fundamental frequency sequence (f0) and the phoneme sequence from the source speaker. After that, they trained a HMM to do phoneme alignment with source speaker's data. Finally, another HMM trained by target speaker's data generated the speech sequence from the f0 and aligned phonemes.

2.2. GMM

The source speech is represented by an n-frame time-series $X = (x_1, x_2, ..., x_n)$, where x_i is a d-dimensional vector, i.e. $X_i = [x_1, x_2, ..., x_d]^T$. The target speech is represented by an m-frame time-series $Y = (y_1, y_2, ..., y_m)$ where $y_i = [y_1, y_2, ..., y_d]^T$. Dynamic Time Warping algorithm is then adopted to align source features to targets to obtain feature pair series $Z = (z_1, z_2, ..., z_q)$ where $z_k = [x_i^T, y_j^T]^T$. The distribution of z is modeled by GMM as below:

$$p(z) = \sum_{l=1}^{L} c_l N(z, u_l, \sum_l) = p(x, y) \tag{2}$$

where c_l is the prior probabilites of Z, given the component l, and it satisfies $\sum_{l=1}^{L} c_l = 1$. $N(z, u, \sum)$ denotes the 2-dimensional normal distribution with the mean vector and the covariance matrix $\sum$. The parameters $(z, u, \sum)$ for the

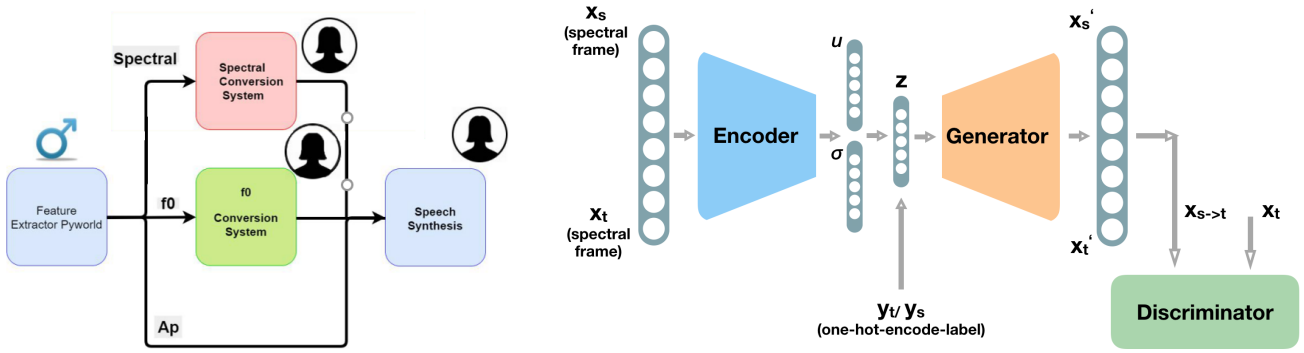

Figure 1. The figure on the left shows that the decomposition of the acoustic information in this project. First, the raw wave will be separated into fundamental frequency (f0), spectral envelop(Sp), aperiodicity(Ap). Then Sp is converted by our system then eventually the three components would be gathered again for target speech synthesis. The figure on the right shows the overview of our proposed architecture. There's an encoder, generator, and a discriminator in our deep voice conversion model. Note that The target representation y is manually given and aggregate with the latent representation z at the bottle network layer.

joint density $p(x, y)$ can be estimated using the Expectation Maximization algorithm.

The transformation function [2] that converts source feature x to target feature y is given below:

$$E(y|x) = \int y\, p(y|x)\, dy \tag{3}$$

$$p_l(y) = \frac{c_l N(x, u_l^x, \sum_l)}{\sum_{l=1}^{L} c_k N(z, u_k^x, \sum_k)} \tag{4}$$

2.3. RNN

Whether being spectrogram or waveform, inputs are likely to be time series. Therefore, it is intuitively to think of using RNN. In [9], Deep Bidirectional Long Short-Term (DBLSTM) architecture is applied to model not only the frame-wised relationship between source and target voice but also the long-range context-dependencies of acoustic trajectory. However, this architecture still requires parallel utterances which utilize the dynamic time warping (DTW) to align feature sequence between the source and target speakers.

3. Methodology

The structure we are going to use is the variational autoencoder combined with the discriminator for VC problem. First, we can decompose our autoencoder into two phases. During the first stage, the encoder encodes a speaker independent latent representation z, which can be viewed as the phonetic-representation of the speeches. Then, in the second stage, z will be concatenated with speaker-identification-representation y, which in our experiments is a one-hot-encoding vector for distinguished subjects. The task of evaluating the similarity between reconstruct feature and target feature goes to the discriminator. Here we use improved W-GAN, as the training objectives. Its discriminative power gives us a supervision on judging whether the generated speech probability distribution is approximate to the real target speakers speech distribution. We believe that with the combination of conditional VAE and adversarial training techniques, we are able to implement a voice conversion system with better acoustic style transfer result. Also, in our experiments, we applied both DNN and RNN version of the network. The main idea using RNN is that it can utilize their internal memory to process arbitrary sequences of inputs to exhibit the dynamic temporal behavior of a sequence, and we plan to use long short-term memory networks (LSTM) as an Encoder-Decoder structure to hopefully reconstruct indistinguishable feature compare with the target feature. Some results are given in the following section.

3.1. Preprocessing

We used WORLD[7] vocoder to extract voice components from the utterances. Voice components include fundamental frequency(f0) which is the pitch, spectral envelope(sp) which is the upper envelope curve of the spectrum and can be viewed as the tone, and aperiodicity(ap) which is upper envelope curve subtracted by lower envelope curve and contains all the other information. Sp would be further normalized to [-1,1]. The conversion of Sp would be our main task in this project, while f0 would be converted by a simple linear mapping function and Ap would remain unchanged from source to target speaker.

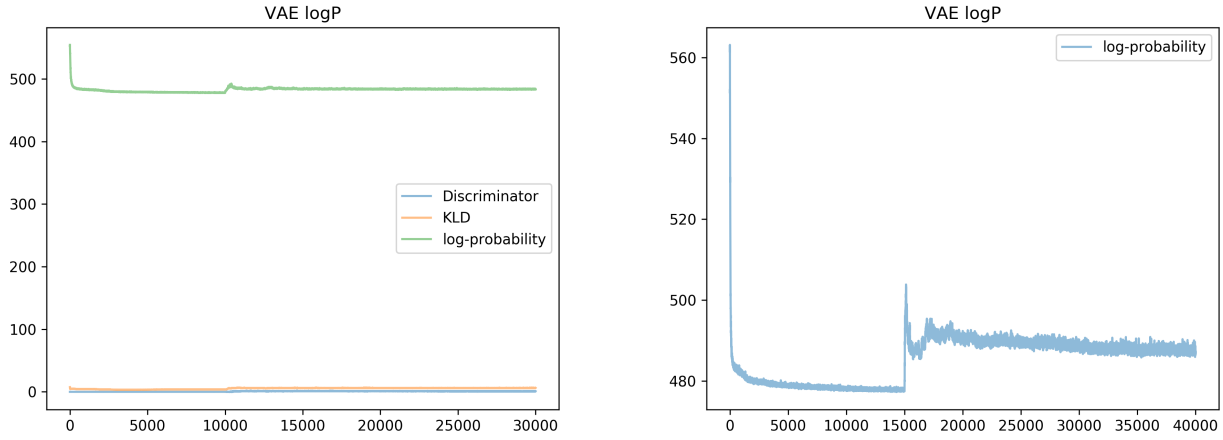

Figure 2. On the left-hand side, the figure shows the training loss curve of 3 different losses, including MSE, KLD, and w-distance, which are represented by green-line, orange-line and blue-line respectively. We can see pretty clear that at iteration 10000 there exist a peak on both three losses, while the MSE loss reacts most sensitive to it. On the right-hand side, we train on VAE in the first 15000 and start to optimize w-distance from 15000 to 40000 iterations. It is simply the enlarge result of MSE loss, which helps us to observe the result more clearly.

3.2. Speech Synthesis With C-VAE

Given spectral frame $X_s = \{x_s, n\}_{n=1}^{Ns}, X_T = \{x_t, n\}_{n=1}^{Nt}$, represent source and target speaker ,respectively. The goal of voice conversion is to find a function f so that the conditional distribution $p_{t|s}$ approximates the real data distribution p_t: $p_{t|s}\left(f\left(x_s, n\right)\right) \approx p_t\left(x_t, n\right)$ Recent works have proven the viability of speech modeling with VAE. We hope to achieve speaker-independent variational autoencoder by conditional VAE(C-VAE). The encoder ε_ϕ infers a latent content z_n , then the decoder g_θ mixes z_n with a speaker-specific variable y to reconstruct the input. $x_{t,n'} \approx f\left(x_s, n\right) = g_\theta\left(\varepsilon_\phi\left(x_s, n\right), y\right)$

Modeling speech with C-VAE In order to train the C-VAE ,we have to maximizes a variational lower bound of the log-likelihood:

$$logp_\theta\left(x|y\right) \leqslant -J_{vae}\left(x|y\right) = -(J_{obs}\left(x|y\right) + J_{lat}\left(x\right)) \tag{5}$$

$$J_{lat}\left(\phi, x\right) = D_{KL}\left(q_\phi\left(z|x\right)|\left|p_\theta(z)\right.\right) \tag{6}$$

$$J_{lat}\left(\phi, \theta, x, y\right) = -E_{q_\phi(z|x)}[logp_\theta\left(x|z, y\right)] \tag{7}$$

where $x \in X_s \cup X_t$, D_{KL} is the KL divergence, $p_\theta\left(z\right)$ is our prior distribution, $p_\theta\left(x|z, y\right)$ is our synthesis model, $q_\phi\left(z|x\right)$ is our inference model.

3.3. Adversarial Training With Improved Wasserstein Distance

Wasserstein GAN [1] has been proposed in 2017 for a more effective way training generative adversarial models. However, beause that wgan have the issue about exploding and vanishing gradients , so we try an altenative way to enforce the Lipschitz constraint. Here, we introduced the gradient penalty proposed in [4], to clip weights by penalizing the norm of gradient of the critic with respect to its input. Hence, the wgan loss J_{wgan} is defined as :

$$E_{\tilde{x}\sim P_g}[D(\tilde{x}))] - E_{\tilde{x}\sim P_r}[D(x))] + \lambda E_{\tilde{x}\sim P\tilde{x}}[(\|\bigtriangledown_{\tilde{x}} D(\tilde{x})) - 1\|)^2] \tag{8}$$

and eventually, parameters can be updated by following objective functions:

$$\psi \xleftarrow{update} -\bigtriangledown_\psi\left(-J_{wgan}\right) \tag{9}$$

$$\phi \xleftarrow{update} -\bigtriangledown_\phi\left(J_{obs} + J_{lat}\right) \tag{10}$$

$$\theta \xleftarrow{update} -\bigtriangledown_\theta\left(J_{obs} + \alpha * J_{wgan}\right) \tag{11}$$

where ψ, ϕ, θ are parameters of discriminator, encoder, generator respectively.

4. Experiment

We use the open dataset provided by Voice Conversion Challenge 2016 [10], which includes 10 speakers, 5 for

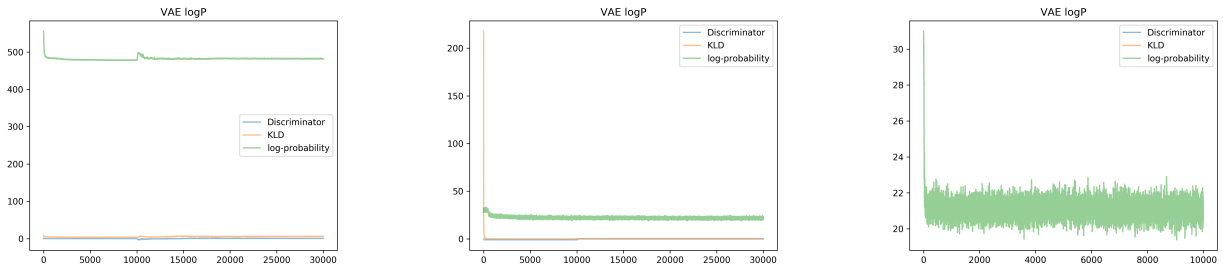

Figure 3. The left figure shows the loss curve of applying DNN architecture in autoencoder, and RNN architecture result in the middle figure. We can see the small peak of MSE-loss(green-line) on 10000 iterations is clear in DNN structure while not obvious in RNN. On the right figure, it simply zoom in the result of MSE-loss in RNN architecture. We could see that it just oscillates in a pretty limited range, and seems that it didn't find a way to go to to global minimum, which means that it may stuck in local minimum.

source speakers(3 women and 2 men) and 5 for target speakers(2 women and 3 men). 216 utterances in each speaker in total while 162 utterances among them are training data and 54 utterances are evaluation data.

For our best experiment result, we use 30000 iterations in total, batch-size set to 256, Using Adam for Stochastic Optimization[6], gradient penalty - λ - set to 1 and the weight of discriminator loss - α - set to 50. Please go to our GitHub repository [5] to hear the best reconstructed audio result.

In the following part, we will focus on discussing several tricks we have found during the training process and also the discussion on the effect of different structures of autoencoder.

4.1. Training Variational Autoencoder

During the training progress of the variational autoencoder, we found that the KullbackLeibler divergence (KLD) used to converge too fast to limit the reconstruction ability of our decoder. We believe it is because when KLD converges to zero also means it restricts the variability of the latent representation z, which leads to the local optimum of the whole network. Therefore, we tried different training scenarios and found that if we train on reconstructing criterion(MSE loss) in the first 5000 iterations, and then jointly training the KLD and reconstruct error in the following 5000 to 30000 iterations, it solves the problem.

One more thing needs to be mentioned is that in the first 10000 iterations we only update the KLD and MSE loss. From 10000 to 30000 iterations we then start to update both w-distance optimization and MSE+KLD. The reason why we update in this order is that we want to make sure our VAE is robust enough to start training discriminator. We can see the training loss curve (Fig2) shows a peak when iterations go to 10000, and that is the point when we start to optimize w-distance.

4.2. Structure of autoencoder (DNN V.S. RNN)

Compare generated results using DNN and RNN structure in autoencoder. We found that the results of the RNN architecture are quite unsatisfied. Simply from the loss curve(Fig3), we can see that the MSE-loss oscillates in a pretty small range. Also, compare with the DNN structure, it is not that sensitive to the join update of w-distance on iteration 10000^{th}. We believe the reason is that although recurrent neural network can preserve the memory information during training progress, however; the temporal information is not long enough (due to the hardware limitation, we only try on time-step smaller than 200). Usually, the results of using RNN will surpass those using DNN only when the time sequence is long enough, and that is the main reason why we got pretty poor results of using RNN.

5. Conclusion

Using a VAE-GAN structure can indeed solve the problem of the unaligned and inconsistent-content dataset and with several training tricks mentioned above, we create a more robust model on multi-person voice conversion system. As the demo is shown in the presentation we can hear the reconstructed voice is pretty nice, we believe with so many promising applications of voice conversion, including speech impaired, gamming, healthcare, etc. We can go way much longer in the future.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[2] Y. Chen, M. Chu, E. Chang, J. Liu, and R. Liu. Voice conversion with smoothed gmm and map adaptation. In *Eighth European Conference on Speech Communication and Technology*, 2003.

[3] Y.-C. W. Y. T. H.-M. W. Chin-Cheng Hsu, Hsin-Te Hwang. Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial

networks, 2017. Submitted to INTERSPEECH 2017 arXiv:1704.00849.

[4] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017.

[5] H. H. K. H. H.C. Yang, G.Y. Chao. General voice conversion. https://github.com/w102060018w/Adversarial-General-Voice-Conversion, 2017.

[6] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[7] M. Morise, F. Yokomori, and K. Ozawa. World: A vocoder-based high-quality speech synthesis system for real-time applications. *IEICE TRANSACTIONS on Information and Systems*, 99(7):1877–1884, 2016.

[8] T. Nose and T. Kobayashi. Hmm-based robust voice conversion using adaptive f0 quantization. In *Seventh ISCA Workshop on Speech Synthesis*, 2010.

[9] L. Sun, S. Kang, K. Li, and H. Meng. Voice conversion using deep bidirectional long short-term memory based recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 4869–4873. IEEE, 2015.

[10] T. Toda, L.-H. Chen, D. Saito, F. Villavicencio, M. Wester, Z. Wu, and J. Yamagishi. The voice conversion challenge 2016., 2016.

[11] J. Yamagishi. An introduction to hmm-based speech synthesis. *Technical Report*, 2006.