ADVERSARIALLY TRAINED AUTOENCODERS FOR PARALLEL-DATA-FREE VOICE CONVERSION

Orhan Ocal[†], *Oguz H. Elibol*[♯], *Gokce Keskin*[♯], *Cory Stephenson*[♯], *Anil Thomas*[♯], *Kannan Ramchandran*[†]

†University of California at Berkeley
Electrical Engineering and Computer Sciences
Berkeley, USA

♯Intel AI Lab
Santa Clara, USA

ABSTRACT

We present a method for converting the voices between a set of speakers. Our method is based on training multiple autoencoder paths, where there is a single speaker-independent encoder and multiple speaker-dependent decoders. The autoencoders are trained with an addition of an adversarial loss which is provided by an auxiliary classifier in order to guide the output of the encoder to be speaker independent. The training of the model is unsupervised in the sense that it does not require collecting the same utterances from the speakers nor does it require time aligning over phonemes. Due to the use of a single encoder, our method can generalize to converting the voice of out-of-training speakers to speakers in the training dataset. We present subjective tests corroborating the performance of our method.

Index Terms— Voice conversion, autoencoders

1. INTRODUCTION

Speech signals contain information besides the uttered message; among them are the speech characteristics that pertain to the speaker. The problem of modifying the speech so that it sounds as if it was uttered by another speaker is known as *voice conversion* [1]. Voice conversion is usually done by training a model that takes an input from a speaker and transforms it so that it sounds like it was spoken by another speaker. The training of the model might require *parallel data*, or can be done using *non-parallel data*. Parallel data refers to recording the same utterance from the speakers, and time-aligning them through dynamic time warping. On the other hand, non-parallel data refers to recordings from speakers where they speak different utterances, and there is no required time-aligning of signals.

In this paper, we present a method for voice conversion based on autoencoders trained on non-parallel data. An autoencoder is an artificial neural network used to learn a representation (encoding) of a given dataset in an unsupervised way [2]. This is done by training an encoder network and a decoder network jointly, where the encoder takes an input and gives out a representation (code), and the decoder outputs a reconstruction of the input based on this representation. The parameters of the encoder and the decoder networks are trained so that the reconstruction matches the input well with respect to a chosen criterion. The typical scenario where autoencoders are used is to learn an efficient representation of the data by constraining representation to be *smaller* compared to the size of the input [2].

Unlike the wide-spread use case of autoencoders, in this work, we use the autoencoder architecture not to find a compact representation of the input, but to learn a representation of the input speech that is *independent* across speakers while still yielding a good reconstruction of the input. For this, we use one encoder, multiple decoders (one for each speaker) and one classifier. The encoder output is guided to a speaker-independent representation in training time by an adversarial loss provided by the classifier. This classifier network takes as input the output of the encoder (the representation) and tries to identify the speaker. The encoder-decoder pairs are trained to minimize the reconstruction error while not enabling the classifier to get a good classification accuracy. In inference time, for performing voice conversion, we feed the speech input to the encoder, and use the decoder of the target speaker. Because we have a single encoder for all speakers, our algorithm can generalize to converting voices of speakers outside the training set to the speakers' in the training set.

2. RELATED WORK

Voice conversion algorithms can be divided into two with respect to datasets they require: algorithms which require parallel datasets and the ones that work on non-parallel datasets.

On the side of algorithms based on parallel datasets, the authors of [3] present a method that models the spectral envelope of speech signals by Gaussian mixture models and then fits a conversion function between the source and the target spectral envelops using time-aligned utterances from the two. Similarly, the authors of [4] propose a conversion method based on the maximum-likelihood estimation of the spectral parameter trajectory instead of working on snapshots.

On the side of parallel-data-free methods, recent work mostly make use of neural networks. The authors of [5, 6]

proposed algorithms to perform domain transfer on images using Generative Adversarial Networks (GANs) [7]. In order to transfer each sample across domains while preserving the contents of the sample, the authors introduce a new term in the loss function called the *cycle loss*. This loss tries to enforce that the sample matches the input after it is transferred to the new domain and then back to the old domain. Voice conversion can be seen as an instance of domain transfer, where the domains correspond to the speakers. The authors of [8] propose an algorithm to do voice conversion using non-parallel data through the aforementioned GAN architecture trained with the cycle loss.

The closest work to our approach is [9] where the authors propose using adversarially trained autoencoders for translating music across domains (instruments, genres, and styles). The authors propose using a universal encoder, and multiple decoders, one for each domain. The multiple autoencoder paths are trained with an adversarial classification loss. Our work differs from this related work in the problem domain, the choice of the features tailored for voice conversion on which the loss function is defined, and the network design which is simpler for computationally faster inference.

3. ADVERSARIALLY-TRAINED AUTOENCODERS FOR VOICE CONVERSION

Let n denote the number of speakers whose voices we want to convert to each other, and let $\mathcal{L}$ denote the set of ids of the speakers. We assume a training dataset consisting of m samples $\mathcal{S} = \{(x_i, l_i)\}_{i=1}^m$, where x_i is the ith utterance, and $l_i \in \mathcal{L}$ is the speaker id for the ith utterance.

We construct one encoder E and n decoders $\mathcal{D} = \{D_i\}_{i\in\mathcal{L}}$, one for each speaker, and a classifier C that is going to be used for defining the adversarial loss. Given a reconstruction loss function f_r and a classification loss function f_c, the training is done to optimize

$$\min_{E,\mathcal{D}} \max_{C} \sum_{i=1}^{m} \left[f_r(x_i, D_{l_i}(E(x_i))) - f_c(l_i, C(E(x_i))) \right]. \quad (1)$$

Example choices for the loss functions are an ℓ_p-norm for the reconstruction loss, that is, $f_r(x, y) = \|x - y\|_p$, and the cross-entropy loss for the classification, that is, $f_c(l, y) = \log\left(e^{y_l} / \sum_{i\in\mathcal{L}} e^{y_i}\right)$.

The intuition behind this particular optimization problem for training is that we want the encoder to learn an embedding that does not carry information with respect to the speaker while still enabling the relevant decoder to reconstruct the speech. This is facilitated by training the encoder-decoder to result in a large error in the classifier.

There is another way to interpret this cost function in terms of finding representations that have small mutual information with the speaker label. If the bottleneck of the autoencoder has no mutual information with the label of the speaker, it means that all the information relevant to the speaker has been stripped off the input. So it would make sense to explicitly minimize this mutual information term as

$$\min_{E,\mathcal{D}} \mathbb{E}\left[f_r(X, D_L(E(X)))\right] + I(L; E(X)). \quad (2)$$

However, it is hard to calculate and minimize the mutual information. Recently, a lower-bound based on deep neural networks to approximate the mutual information has been proposed by the authors of [10]. The following proposition shows that, similarly, the classification accuracy can be viewed as a bound on the mutual information.

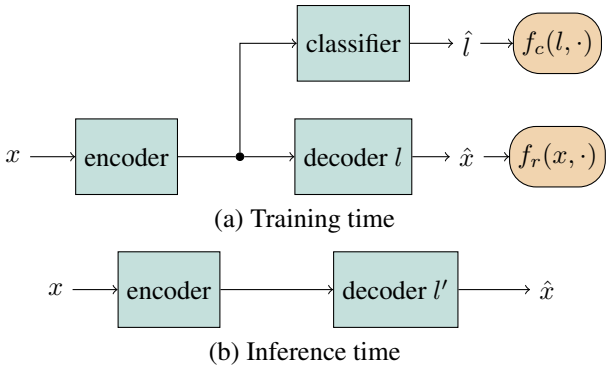


Fig. 1. Our architecture consists of multiple autoencoder paths, where there is a single encoder and multiple decoders, one for each speaker. The classifier takes as input the output of the encoder, and it provides the encoder an adversarial loss to guide the representation to be independent of the speaker.

Proposition 1 *Let L be the input id of the speaker, X be the speech sample, Z be the representation of X given by the encoder, and $\hat{L}$ be the estimation of speaker id based on Z. Let $p_e = P(L \neq \hat{L})$, and $p_e^* = \min_f P(L \neq f(Z))$; then*

$$\begin{aligned} &H(L) - h(p_e) - p_e \log(|L| - 1) \\ &\leq I(L; Z) \leq H(L) + \log_2(1 - p_e^*). \end{aligned} \quad (3)$$

The lower bound to the mutual comes from Fano's inequality, and the upper bound comes from manipulating definition of the best classifier; due to space constraints we omit the proof of the proposition. As can be seen from the proposition, the error probability of any classifier provides a lower bound, and the error probability of the best classifier provides an upper bound on the mutual information. Hence, in the cost function (1), the classifier is trained to maximize this lower-bound on the mutual information to get an approximation to it. Then, the encoder-decoder pair is trained to minimize this approximation to the mutual information along with the the reconstruction loss. If the neural network were able to represent the best classifier, and the optimization algorithm were able to find it, then we would be able to get an upper bound on the mutual information as well. Figure 2 shows the lower and upper bounds for the 4 speakers chosen uniformly at random.

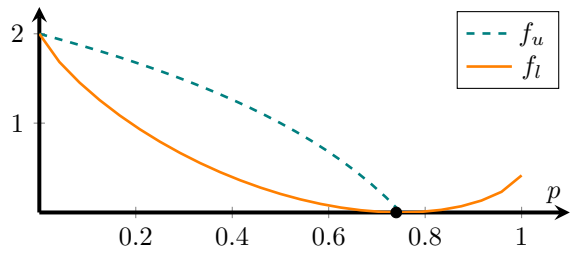


Fig. 2. An example for the bounds on the mutual information given in equation (3). Speaker id L chosen uniformly at random from a set of 4 speakers. The curve $f_l(p) = H(L) - h(p) - p\log(|L|-1)$ denotes the lower bound, and the curve $f_u(p) = H(L) + \log_2(1-p)$ denotes the upper bound on the mutual information $I(L,Y)$.

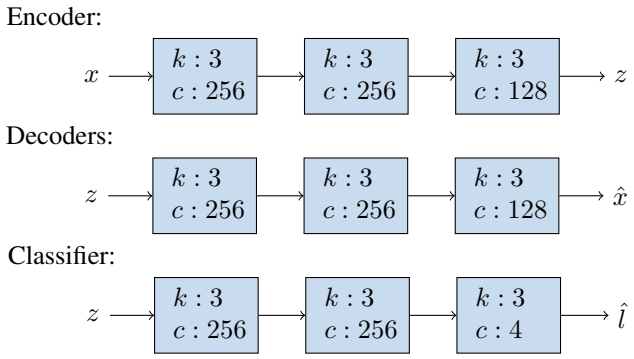


Fig. 3. Architectures of the encoder, decoders and classifier. Each block represents convolution followed by instance normalization an ReLU. The convolution kernel size is denoted by k, and the number of output channels is denoted by c.

4. EXPERIMENTS

As numerical experiments we select five speakers, two female (denoted with F1 and F2) and three male (denoted with M1, M2 and MX), from English multi-speaker corpus of CSTR voice cloning toolkit [11]. We use four of these speakers (F1, F2, M1, and M2) in training the neural networks, and set aside one of the speakers (MX) for evaluating the performance for conversion of voice from out-of-training speakers.

We train multiple autoencoder paths on the mel-frequency spectrogram magnitudes of speech. The parameters for the spectrograms are as follows: the number of FFT bins is 1024, the hop length between consecutive windows is 256, number of mel-spaced filters is 128, and minimum and maximum frequencies of input signal considered are 40 Hz and 8000 Hz respectively. The reconstruction loss, f_r, is chosen to be ℓ_1 loss, and the classification loss, f_c, is chosen to be the cross-entropy loss. The encoder, decoder and classifier are all three layer convolutional neural networks, where the convolutions are 1-dimensional have kernel size of 3. The number of hidden channels for each network is 256, and the representation size (code length) is chosen to be 128. We use instance normalization [12] before each non-linearity, and use ReLU as the activation functions. Fig. 3 shows an illustration of the architecture. To complete the loop on voice conversion, after the conversion of spectrograms using the neural network, we use Griffin-Lim's algorithm [13] to construct the audio signal similar to the way Tacotron [14] does.

As an example for converted voice spectrogram, we look at converting an input sample from one of the male speakers (M2) to the voice of one of the female speakers (F1). Fig. 4 shows the input spectrogram, spectrogram of the reconstructed speech and the spectrogram after voice conversion. As can be seen from the figure, the reconstruction resembles the input spectrogram, and the spectrogram of the converted voice has components at higher frequencies compared to the input's which aligns with the target speaker's voice characteristics.

In order to get quantitative performance metrics, we conducted subjective tests on Amazon Mechanical Turk [15]. We report the results of three types of tests: (1) AB testing, (2) ABX testing, and (3) mean opinion scores (MOS).

source speaker \ target speaker	F1	F2	M1	M2
F1	–	0.500 ± 0.224	0.825 ± 0.069	0.800 ± 0.080
F2	0.567 ± 0.128	–	0.756 ± 0.091	0.750 ± 0.087
M1	0.846 ± 0.063	0.818 ± 0.074	–	0.776 ± 0.064
M2	0.914 ± 0.067	0.856 ± 0.074	0.675 ± 0.105	–
MX	0.867 ± 0.088	0.900 ± 0.072	0.607 ± 0.083	0.767 ± 0.089

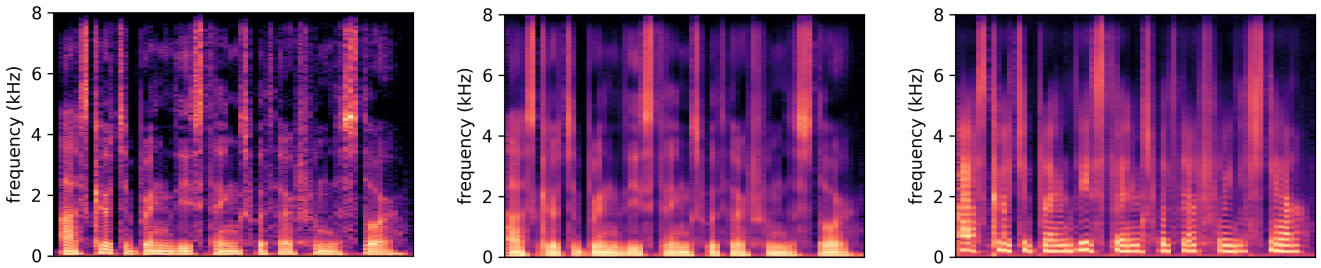
Table 1. AB test results.

source speaker \ target speaker	F1	F2	M1	M2
F1	–	0.651 ± 0.103	0.872 ± 0.069	0.868 ± 0.110
F2	0.650 ± 0.151	–	0.912 ± 0.069	0.860 ± 0.075
M1	0.855 ± 0.081	0.826 ± 0.082	–	0.776 ± 0.069
M2	0.750 ± 0.137	0.840 ± 0.077	0.634 ± 0.081	–
MX	0.776 ± 0.110	0.833 ± 0.063	0.653 ± 0.064	0.800 ± 0.179

Table 2. ABX test results.

source speaker \ target speaker	F1	F2	M1	M2
F1	–	2.371 ± 0.232	2.312 ± 0.195	2.138 ± 0.192
F2	1.850 ± 0.205	–	1.712 ± 0.180	1.750 ± 0.160
M1	1.588 ± 0.160	1.754 ± 0.126	–	2.327 ± 0.201
M2	2.140 ± 0.159	2.020 ± 0.164	2.689 ± 0.218	–
MX	2.670 ± 0.182	2.422 ± 0.190	2.650 ± 0.199	2.577 ± 0.19

Table 3. MOS.



(a) Input (b) Reconstruction (c) Conversion

Fig. 4. Spectrograms of input speech, reconstruction and voice conversion. Input is an utterance by a male speaker and the target is a female speaker who has a higher pitched voice compared to the input speaker. We observe that the converted voice has components at higher frequencies compared to the input which aligns with the characteristics of the target speaker.

The first of the subjective experiments is AB tests. In these tests, the listeners are given two utterances to compare. The first sample (A) is a recording by either the source or the target speaker chosen uniformly at random, and the second one (B) is an utterance converted to the voice of the target speaker from the source. In order to not bias the listener based on the contents of the utterances, we use different sentences for the given two samples. The subjects are asked to identify if the two samples are spoken by the same speaker or not. We call it a success if (1) the first sample (A) was a recording from the target and the subject said the two samples were from the same speaker, or (2) the first sample (A) was a recording from the source and the subject said the two samples were from different speakers.

Table 1 shows the frequency of success in the experiments along with 95% confidence intervals. As can be seen from the table, our algorithm achieves to change the source voice in the correct direction. In particular, the conversion of voice between speakers of different genders seem to be consistently perceived successfully by the subjects. An interesting observation is that converting the voice of an out-of-training speaker (given in the last row of the table) performs similarly with in-training speakers, which means that the encoder can generalize to out-of-training speakers.

The second subjective experiment is ABX tests. Here, the listeners are given three samples. The first two samples (A and B) are recordings from the source and target speakers (order is randomized), and the third sample (X) is an utterance converted from the source speaker to the voice of the target speaker. As in AB tests, we use different sentences for the given three samples in order not to bias the listener based on the contents of the utterances. The subjects are asked to choose which of the first two samples' voice (A's or B's) is the third sample's voice is closer to. We call it a success if the subject chooses the sample recorded from the target speaker.

Table 2 shows the frequency of success in the experiments along with 95% confidence intervals. Again, our algorithm changes the source voice in the correct direction, in particular, for converting between speakers with different genders. Similar to AB tests, the voice of an out-of-training speaker (last row of the table) performs similarly with in-training speakers.

The last of the subjective tests is MOS. Here, the subjects were given converted speech samples, and were asked to rate the quality of the sample. The quality scale used was Absolute Category Rating (ACR) [16], that is, integers with correspondences: 1-bad, 2-poor, 3-fair, 4-good, 5-excellent.

Table 3 shows the MOS along with 95% confidence intervals. Even though, our algorithm changes the source voice in the correct direction as was shown with the AB and ABX tests, the converted sample has artifacts that a listener can notice as is evident from the MOS. A large part of it is because small inconsistencies in the spectrogram magnitudes can result in significant artifacts in the output when Griffin-Lim's algorithm is used. This behavior was also identified by the authors of [17] who propose training a separate deep neural network that takes a spectrogram magnitude and outputs audio signals to reduce artifacts. Using such more elaborate approaches to reduce artifacts is part of ongoing research.

5. DISCUSSION

We presented a method for voice conversion using neural networks trained on non-parallel data. The method is based on an training multiple autoencoder paths where there is a single speaker-independent encoder and multiple speaker-dependent decoders. The autoencoder paths are trained to minimize the reconstruction error and an adversarial cost that tries to make the output of the encoder carry no information with respect to the speaker id. The training is unsupervised in the sense that we do not require parallel speech dataset from the speakers. We evaluated our method on a subset of speakers from the VCTK dataset. Qualitatively, we observe that the converted spectrograms carry characteristics of the spectrograms of the target speaker. The results of subjective tests corroborate our algorithm's voice conversion performance. Although our algorithm can convert the voice of the source speaker in the direction of the target, we observe that reconstructed audio has some artifacts. Reducing these artifacts is ongoing work.

6. REFERENCES

[1] E Moulines and Y Sagisaka, "Voice conversion: State-of-the-art and perspectives," *Speech Communication*, vol. 16, no. 2, pp. 125–126, Feb 1995.

[2] Geoffrey E Hinton and Richard S Zemel, "Autoencoders, minimum description length and helmholtz free energy," in *Advances in neural information processing systems*, 1994, pp. 3–10.

[3] Yannis Stylianou, Olivier Cappé, and Eric Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Transactions on speech and audio processing*, vol. 6, no. 2, pp. 131–142, 1998.

[4] Tomoki Toda, Alan W Black, and Keiichi Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2222–2235, 2007.

[5] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *arXiv preprint*, 2017.

[6] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman, "Toward multimodal image-to-image translation," in *Advances in Neural Information Processing Systems*, 2017, pp. 465–476.

[7] 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*, 2014, pp. 2672–2680.

[8] Takuhiro Kaneko and Hirokazu Kameoka, "Parallel-data-free voice conversion using cycle-consistent adversarial networks," *arXiv preprint arXiv:1711.11293*, 2017.

[9] Noam Mor, Lior Wolf, Adam Polyak, and Yaniv Taigman, "A universal music translation network," *arXiv preprint arXiv:1805.07848*, 2018.

[10] Ishmael Belghazi, Sai Rajeswar, Aristide Baratin, R Devon Hjelm, and Aaron Courville, "Mine: mutual information neural estimation," *arXiv preprint arXiv:1801.04062*, 2018.

[11] Junichi Yamagishi, "English multi-speaker corpus for CSTR voice cloning toolkit," `http://homepages.inf.ed.ac.uk/jyamagis/page3/page58/page58.html`, 2012.

[12] Xun Huang and Serge J Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization.," in *ICCV*, 2017, pp. 1510–1519.

[13] Daniel Griffin and Jae Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[14] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al., "Tacotron: Towards end-to-end speech synthesis," *arXiv preprint arXiv:1703.10135*, 2017.

[15] Michael Buhrmester, Tracy Kwang, and Samuel D Gosling, "Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data?," *Perspectives on psychological science*, vol. 6, no. 1, pp. 3–5, 2011.

[16] P ITU-T RECOMMENDATION, "Subjective video quality assessment methods for multimedia applications," 1999.

[17] Sercan Arik, Gregory Diamos, Andrew Gibiansky, John Miller, Kainan Peng, Wei Ping, Jonathan Raiman, and Yanqi Zhou, "Deep voice 2: Multi-speaker neural text-to-speech," *arXiv preprint arXiv:1705.08947*, 2017.