Sequence-to-Sequence Voice Conversion with Similarity Metric Learned Using Generative Adversarial Networks

Takuhirō Kaneko, Hirokazu Kameoka, Kaoru Hiramatsu, Kunio Kashino

NTT Communication Science Laboratories, NTT Corporation, Japan
{kaneko.takuhiro,kameoka.hirokazu,hiramatsu.kaoru,kashino.kunio}@lab.ntt.co.jp

Abstract

We propose a training framework for sequence-to-sequence voice conversion (SVC). A well-known problem regarding a conventional VC framework is that acoustic-feature sequences generated from a converter tend to be over-smoothed, resulting in buzzy-sounding speech. This is because a particular form of similarity metric or distribution for parameter training of the acoustic model is assumed so that the generated feature sequence that averagely fits the training target example is considered optimal. This over-smoothing occurs as long as a manually constructed similarity metric is used. To overcome this limitation, our proposed SVC framework uses a similarity metric implicitly derived from a generative adversarial network, enabling the measurement of the distance in the high-level abstract space. This would enable the model to mitigate the over-smoothing problem caused in the low-level data space. Furthermore, we use convolutional neural networks to model the long-range context-dependencies. This also enables the similarity metric to have a shift-invariant property; thus, making the model robust against misalignment errors involved in the parallel data. We tested our framework on a non-native-to-native VC task. The experimental results revealed that the use of the proposed framework had a certain effect in improving naturalness, clarity, and speaker individuality.

Index Terms: voice conversion, deep neural network, generative adversarial network, similarity metric learning

1. Introduction

In this paper, we address the problem of converting the voice characteristics of speech including speaker identity and pronunciation quality. We are particularly interested in developing a real-time (or low-latency) pronunciation-conversion system, which converts non-native speech into native-like intelligible speech in an online manner. The problem of converting non/para-linguistic information of speech while preserving linguistic information is called voice conversion (VC). Since pronunciation quality is part of a speaker’s identity in a wide sense, the pronunciation-conversion problem can be seen as a special class of the VC problem.

One successful VC framework involves statistical methods based on Gaussian mixture models (GMMs) [1, 2, 3]. This framework typically consists of collecting parallel data, namely a pair of time-aligned feature sequences of source and target speech and training a mapping function between the acoustic features of the source and target speech using the parallel data. At test time, the feature sequence of input speech is converted using the trained mapping function. This GMM-based framework uses a GMM to represent the conditional distribution of the target feature given the source feature. Recently, a neural network (NN)-based framework based on restricted Boltzmann machines [4] and feed-forward deep NNs [5, 6] and an exemplar-based framework based on non-negative matrix factorization (NMF) [7, 8] have also been proposed with notable success.

A well-known problem regarding the conventional VC frameworks is that acoustic-feature sequences generated from a converter tend to be over-smoothed, resulting in buzzy-sounding speech. This is caused by a side effect of assuming a particular form of similarity metric (e.g., mean squared error (MSE)) or distribution (e.g., Gaussian) for parameter training of the acoustic model so that the generated feature sequence that averagely fits the training target example is considered optimal. This over-smoothing occurs as long as a manually constructed similarity metric is used. Postfiltering methods based on the global variance [9, 10] or modulation spectrum [11] have been proposed with the aim of reconstructing the original spectro-temporal fine texture. However, these methods rely on heuristics since it is usually difficult to model the exact probability density of the features of real speech. Recently, a direct waveform-modification method using the spectrum differential feature has been proposed [12]. This method avoids generating over-smoothed spectra by transplanting the spectro-temporal fine texture of the source speech into the generated speech.

Most conventional VC frameworks are designed to convert acoustic features frame-by-frame. When it comes to a pronunciation-conversion task, we must consider the contextual or temporal dependencies of the conversion rules since the way speakers pronounce each phoneme may differ depending on the preceding, current, and succeeding contexts. To model long-term dependencies, some attempts have been made to model the mapping function using recurrent NNs (RNNs) [13] including the long short-term memory networks [14]. However, RNNs are not well suited to parallel implementations; thus, the conversion process becomes computationally demanding.

Except for some parallel-data-free frameworks [15, 16, 17], many VC frameworks require accurately aligned parallel data of source and target speech. When there is a large acoustic gap between source and target speech, collecting parallel speech data can be a painstaking process since automatic time alignment between the pair of acoustic-feature sequences becomes relatively difficult. Since many frameworks are weak against misalignment involved in parallel data, careful pre-screening and manual correction may be required to make these frameworks reliable. It is difficult to completely avoid misalignment errors particularly in a pronunciation-conversion task where the acoustic characteristics of each phoneme pronounced by source and target speakers can be very dissimilar.

In this paper, we propose a VC framework that (1) avoids over-smoothing of converted feature sequences, (2) provides a reasonable way of modeling time-dependent conversion rules, and (3) is robust against misalignment errors involved in the parallel data. We use a similarity metric implicitly obtained with a generative adversarial network (GAN) [18] that has been suc-
cessful in image generation [19, 20] and has begun to be applied in speech synthesis [21, 22, 23]. A GAN offers a framework for training a generator in such a way that it can deceive a discriminator that attempts to distinguish real data from the samples drawn from the generator. Thus, we expect that measuring the difference in the units in a particular layer in the GAN discriminator between the generated and target speech will enable the measurement of the similarity in high-level abstract space and mitigate the over-smoothing problem caused in the low-level data space. To model time-dependent conversion rules, we use a segmental feature (a sequence of acoustic features with a duration of hundreds of frames) as the input/output. For the generator, converter, and discriminator networks, we use a gated convolutional NN (CNN) [24], so that this network structure not only allows for modeling long-term dependencies by deepening the layers but also offers robustness against misalignment errors involved in the parallel data due to the nature of the CNN, known as the shift invariance property.

We previously proposed a learning-based postfilter using a GAN that adds a high-fidelity spectro-temporal texture to the feature sequence obtained with parametric speech synthesis so that the synthesized speech becomes as indistinguishable as possible from real speech [21]. The proposed framework combines the goals of VC and postfiltering, namely converting input speech as close as possible to target speech and making the output speech as indistinguishable as possible from real speech. We tested our framework on a non-native-to-native VC task. The experimental results showed that our proposed framework can be used to improve naturalness, clarity, and speaker individuality.

This paper is organized as follows. In Section 2, we explain the proposed sequence-to-sequence VC (SVC) framework with a GAN and gated CNN. We present the experimental results in Section 3 and summarize our findings in Section 4.

2. Sequence-to-sequence voice conversion

2.1. Framework overview

In a conventional VC framework, the goal is to convert voice characteristics, and a frame-by-frame approach, as shown in Figure 1(a), has been widely used for this conversion. In contrast, our goal is not only to convert voice characteristics but also convert pronunciation, e.g., to convert non-native speech to native-like speech. To achieve this goal, we need to model the temporal dependencies of a speech sequence. To achieve this, the conventional frame-by-frame framework is difficult to apply since it is limited to modeling the mapping function (namely, the converter (C)) in a short time (e.g., 5 ms) and is not able to model the temporal dependencies. Our proposed SVC framework was developed to overcome this limitation, as shown in Figure 1(b), which converts speech on a sequence-to-sequence basis using a C that models long-range context dependencies.

To obtain a suitable C, we need to find a suitable answers to the following two questions: “How is the C optimized?” and “How is the sequence structure modeled?” We answer the first question in Section 2.2 and the second question in Section 2.3.

2.2. Optimization with learned similarity metric and GAN

Sequence-to-sequence VC is considered a matching problem. Given a source acoustic-feature sequence \( x = (x_1, \ldots, x_L) \) and target acoustic-feature sequence \( y = (y_1, \ldots, y_T) \) as training data, the \( C \) that maps \( x \) to \( y \) can be optimized by minimizing the following objective function:

\[
L_{SVC} = S(y, C(x)),
\]

where \( S(x_1, x_2) \) is a similarity metric.

In conventional VC frameworks, the similarity between the source and target acoustic features is measured in the data space using a particular form of similarity metric, e.g., MSE:

\[
L_{SVC}^{Data} = \frac{1}{M_0} |y - C(x)|^2,
\]

where \( M_0 \) is the dimensions of \( y \). The weak point of these frameworks is that they are sensitive to the slight translation in the data space. For example, this metric imposes a large error when a sharp peak is shifted by only one frame. This causes over-smoothing, resulting in buzzy-sounding speech.

To solve this problem, we use a similarity metric learned using a GAN [18]. In the following sub-sections, we first give a brief overview of a GAN then explain our proposed framework with the similarity metric learned using a GAN.

2.2.1. Background: GAN

A GAN [18] is a framework for training a deep generative model using a minimax game. The goal is to learn a generator distribution \( P_G(y) \) that matches the true data distribution \( P_{Data}(y) \). It consists of two networks: a generator \( G \) that transforms noise variables \( z \sim P_{Noise}(z) \) to data space \( y = G(z) \) and a discriminator \( D \) that assigns probability \( D(y) \) when \( y \) is a sample from the \( P_{Data}(y) \) and assigns probability \( 1 - p \) when \( y \) is a sample from the \( P_G(y) \). In a GAN, the following binary cross entropy is maximized and minimized with respect to the \( D \) and \( G \), respectively.

\[
L_{GAN} = \log D(y) + \log(1 - D(G(z))).
\]

This enables the \( D \) to find the binary classifier that provides the best possible discrimination between true and generated data and simultaneously enables the \( G \) to fit \( P_{Data}(y) \). Both \( G \) and \( D \) can be trained using back-propagation.

2.2.2. Similarity metric learning with GAN

In a GAN, its \( D \)'s hidden feature space is optimized to determine the key difference between true and generated data to discriminate them. This means that a more abstract structure is determined in the \( D \)'s hidden feature space than that in the data space. We use this metric to measure similarity. This technique is also used in computer vision [25], in which this similarity metric enables a high-fidelity image to be generated. In the term...
$D_l(y)$, which denotes the $l$-th layer hidden feature in the $D$, a new similarity metric is written as

$$L_{SVC}^{D_l} = \frac{1}{N_l} ||D_l(y) - D_l(C(x))||^2,$$

where $M_l$ is the dimension of $D_l(y)$.

Following a previous study [25], we train our model with the joint objectives of the GAN objective with Equation 3 and SVC objective with Equation 4:

$$\mathcal{L} = \mathcal{L}_{SVC}^{D_l} + \mathcal{L}_{GAN}.$$  (5)

Figure 2 gives an overview of the training procedure. In practice, we obtain better results when using not only a sample from the $G$ but also a sample from the $C$ as the discriminator input. This is because similar positive and negative samples are more useful for the $D$ to determine the decision boundary. In this setting, the GAN objective function is rewritten as

$$\mathcal{L}_{GAN} = \log D_l(y) + \log(1 - D_l(G(z))) + \log(1 - D_l(C(x))).$$  (6)

We optimize the $G$ by minimizing $\mathcal{L}_{GAN}$, optimize the $C$ by minimizing $\gamma \mathcal{L}_{SVC}^{D_l} + \mathcal{L}_{GAN}$, and optimize the $D$ by minimizing $-\mathcal{L}_{GAN}$. Here, $\gamma$ is the trade-off parameter between $\mathcal{L}_{SVC}^{D_l}$ and $\mathcal{L}_{GAN}$, which we set to 1 in the experiments.

### 2.3. Sequence modeling with gated CNN

To achieve SVC, we need to model the long-range context-dependencies. To do so with a reasonably small number of parameters, we use a CNN. Figure 3 shows the network architectures of the $G$, $C$, and $D$. We consider converting the $F \times T$ time-sequential acoustic features (e.g., mel-cepstrum), where $F$ is the feature dimension and $T$ is the frame length. We carefully design the network architectures considering the objective of each network. To convert the source acoustic features to the target ones considering the temporal dependency in the overall features, we design the $C$ using a one-dimensional (1D) CNN [26, 27]. In this model, we regard the feature dimension as a channel (i.e., RGB in the case of an image) and convolve the feature using a $(1, k_2)$ kernel, where $k_2$ is the kernel width in the time direction. In the $G$ and $D$, we focus on the 2D texture (i.e., spectro-temporal structure) of acoustic features. Based on this intuition, we treat them as image-like data [19, 20]. We design the $G$ to generate the 2D texture from $K$-dimensional random noise using an up-sampling 2D CNN and design the $D$ to discriminate the 2D texture using a down-sampling 2D CNN.

Based on the observation that acoustic features have region dependency (e.g., they have different structures in voiced segments and unvoiced segments), we design all the CNNs using a gated CNN [24]. In a gated CNN, gated linear units (GLUs) are used as an activation function instead of regular rectified linear units (ReLU) [28] or Tanh activations. A GLU is a data-driven activation function, and the $l$-th layer output $H_l$ is calculated using the $(l - 1)$-th layer output $H_{l-1}$ and model parameters $W, V, b, c$.

$$H_l = (H_{l-1} * W + b) \odot \sigma(H_{l-1} * V + c),$$  (7)

where $C$ is the sigmoid function and $\odot$ is the element-wise product.
frameworks for comparison. One is a state-of-the-art frame-
by-frame VC framework (the FVC) [6] that uses a deep autoen-
coder to obtain a compact representation of MCEPs. It has 7
hidden layers, and the numbers of units in input, hidden, and
output layers are [25, 100, 40, 15, 150, 15, 40, 100, 25]. The
other framework has the same converter as the LSM but it is
optimized using MSE (the MSE).

3.2. Objective evaluation

Figure 5 shows the comparison of the MCEPs and STFT spec-
trograms of the SRC, TGT, FVC, MSE, and LSM. In the FVC
and MSE, the spectral textures were over-smoothed both in the
mel-cepstral and spectral domains. In the LSM, the spectral tex-
tures similar to the target ones were reproduced not only in the
mel-cepstral domain but also in the spectral domain, although
the spectral conversion is performed in the mel-cepstral domain.

3.3. Subjective evaluation

We conducted listening tests by referring to the Voice Conver-
sion Challenge 2016 (VCC 2016) [37] at which naturalness and
similarity were used as evaluation metrics. We also evaluated
for clarity to assess the effectiveness in pronunciation conver-
sion. We conducted an AB test on naturalness and clarity: par-
ticipants were asked to select the preferred one or to opt for
neutral if they did not have any preference. To measure the
similarity of the VC samples, the same/different paradigm was
used: participants were given two samples and asked whether
the samples were produced by the same speaker. For evalua-
tion, ten sample pairs were selected from the test data. In the
AB test, all pairs were presented in both orders (AB and BA) to
eliminate bias in the order of stimuli. There were seven partici-
pants who were well-educated English speakers.

Tables 2 and 3 list the average preference scores for nat-
uralness and clarity, respectively. These results indicate that
the LSM outperformed the other two. In particular, the differ-
ence was marked regarding clarity, indicating that the LSM is
effective against the over-smoothing problem. Figures 6 shows
the similarity to the TGT and SRC with the three frameworks.
All frameworks were competitive regarding dissimilarity to the
SRC, but the LSM outperformed the MSE and FVC and ex-
ceded the neutral rate regarding similarity to the TGT. Hence,
we can conclude that the proposed framework can improve the
quality of VC.

Table 1: Details of network architectures of G, C, and D

<table>
<thead>
<tr>
<th>G (Input: K Random noise, Output: F × T MCEP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1024 fully connected, BNorm, ReLU</td>
</tr>
<tr>
<td>[F/4] × [T/4], 128 fully connected, BNorm, ReLU</td>
</tr>
<tr>
<td>4 × 4 128 conv. ↑, BNorm, ReLU</td>
</tr>
<tr>
<td>4 × 4 2 convolution, ↑, GLU</td>
</tr>
<tr>
<td>1 fully connected, sigmoid</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C (Input: F × T MCEP, Output: F × T MCEP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1 × 11 1024 conv., GLU) × 5 stacks</td>
</tr>
<tr>
<td>1 × 12 F conv., GLU</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>D (Input: F × T MCEP, Output: 1 Probability)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 × 4 128 conv., ↓, GLU</td>
</tr>
<tr>
<td>4 × 4 256 conv. ↓, BNorm, GLU</td>
</tr>
<tr>
<td>1024 fully connected, BNorm, LeLU</td>
</tr>
<tr>
<td>1 fully connected, sigmoid</td>
</tr>
</tbody>
</table>

Figure 5: Comparison of MCEPs (upper row) and STFT spectrograms (lower row) of SRC, TGT, FVC, MSE, and LSM

Tables 2 and 3 list the average preference scores for nat-
uralness and clarity, respectively. These results indicate that
the LSM outperformed the other two. In particular, the differ-
ence was marked regarding clarity, indicating that the LSM is
effective against the over-smoothing problem. Figures 6 shows
the similarity to the TGT and SRC with the three frameworks.
All frameworks were competitive regarding dissimilarity to the
SRC, but the LSM outperformed the MSE and FVC and ex-
ceded the neutral rate regarding similarity to the TGT. Hence,
we can conclude that the proposed framework can improve the
quality of VC.

Table 2: Average preference scores (%) for naturalness with 95% confidence intervals. Bold font indicates top scores.

<table>
<thead>
<tr>
<th></th>
<th>Former</th>
<th>Latter</th>
<th>Neutral</th>
</tr>
</thead>
<tbody>
<tr>
<td>FVC vs. LSM</td>
<td>32.9 ± 7.9</td>
<td>54.3 ± 8.4</td>
<td>12.9 ± 5.6</td>
</tr>
<tr>
<td>MSE vs. LSM</td>
<td>27.1 ± 7.5</td>
<td>65.0 ± 8.0</td>
<td>7.9 ± 4.5</td>
</tr>
</tbody>
</table>

Table 3: Average preference scores (%) for clarity with 95% confidence intervals. Bold font indicates top scores.

<table>
<thead>
<tr>
<th></th>
<th>Former</th>
<th>Latter</th>
<th>Neutral</th>
</tr>
</thead>
<tbody>
<tr>
<td>FVC vs. LSM</td>
<td>17.1 ± 6.3</td>
<td>72.9 ± 7.5</td>
<td>10.0 ± 5.0</td>
</tr>
<tr>
<td>MSE vs. LSM</td>
<td>10.0 ± 5.0</td>
<td>84.3 ± 6.1</td>
<td>5.7 ± 3.9</td>
</tr>
</tbody>
</table>

Figure 6: Similarity to TGT and SRC with VC frameworks

4. Conclusions

We explored sequence-to-sequence VC. To measure the similar-
ity for complex sequential data, we proposed a VC framework
that uses a similarity metric learned using a GAN. To model the
sequence structure, we introduced a gated CNN into our model.
In the experiments, we applied our framework to non-native-to-
native VC tasks. The results showed that our proposed frame-
work outperformed the state-of-the-art framework, as well as
that optimized using a conventional similarity metric, in terms
of naturalness, clarity, and speaker individuality. Our proposed
framework using a learned similarity metric is not limited to a
particular task, so we plan to apply it to other tasks such as TTS.

Acknowledgements: We thank Mr. Keisuke Oyamada (Univ.
of Tsukuba), who kindly provided us with the source code of the
state-of-the-art VC framework [6]. This work was supported by
JSPS KAKENHI Grant Numbers JP26730100 and JP26280060.
5. References


