Cross-Dialectal Voice Conversion with Neural Networks

Weixun GAO††, Student Member, Qiyling CAO††, and Yao QIAN†††, Nonmembers

SUMMARY In this paper, we use neural networks (NNs) for cross-dialectal (Mandarin-Shanghainese) voice conversion using a bi-dialectal speakers’ recordings. This system employs a nonlinear mapping function, which is trained by parallel mandarin features of source and target speakers, to convert source speaker’s Shanghainese features to those of target speaker. This study investigates three training aspects: a) Frequency warping, which is supposed to be language independent; b) Pre-training, which drives weights to a better starting point than random initialization or be regarded as unsupervised feature learning; and c) Sequence training, which minimizes sequence-level errors and matches objectives used in training and converting. Experimental results show that the performance of cross-dialectal voice conversion is close to that of intra-dialectal. This benefit is likely from the strong learning capabilities of NNs, e.g., exploiting feature correlations between fundamental frequency (F0) and spectrum. The objective measures: log spectral distortion (LSD) and root mean squared error (RMSE) of F0, both show that pre-training and sequence training outperform the frame-level mean square error (MSE) training. The naturalness of the converted Shanghainese speech and the similarity between converted Shanghainese speech and target Mandarin speech are significantly improved.

key words: voice conversion, neural network, cross-dialectal, frequency warping, pre-training, sequence training

1. Introduction

Voice conversion is a technique that can convert the voice of speaker A (source speaker) to that of speaker B (target speaker) with limited recording speech of the target speaker. It modifies the voice characteristics of the source speaker’s speech so that it can be perceived as if it were uttered by the target speaker without losing any information or message delivered by the source speaker’s speech. It is used as one of the key technologies in many applications, such as customizing a voice needed for a smart home system, simulating dialogues between various characters in a language learning system, personalizing output in a speech-to-speech translation system, and converting electrolaryngeal speech to normal speech for laryngectomees in a speaking-aid system, among others.

Voice conversion transforms the spectrum and the fundamental frequency (F0) of the source speaker’s speech to match those of the target speaker. The statistical approach in voice conversion has significantly improved the quality of converted speech in the past few decades. Voice conversion first uses parallel utterance pairs of the source and target speakers to train a Gaussian mixture model (GMM), then converts the source speaker’s parameters into those of the target speaker in a minimum-mean-square error or maximum-likelihood (ML) sense, and finally generates speech from converted parameters by a source-filter speech product model [1], [2]. Voice conversion is also used to convert the speech synthesized by a unit selection–based TTS system [3].

Recently, neural networks (NNs) based voice conversion, which uses a nonlinear mapping function to model the relationship between input source and output target speakers’ features, has been shown to outperform GMM-based voice conversion [4]–[7]. A feed-forward neural networks is used in voice conversion for the mapping of spectral features of a source speaker to those of a target speaker [4]. Deep belief nets (DBN) with stacked restricted Boltzmann machines (RBMs) is used to build high-order eigen spaces of the source/target speaker, where it is supposed to capture more speaker individuality information than the phonetic information, and should be better to convert the source speech to the target speech than in the traditional cepstrum space [5]. Gaussian distribution in mixture component of GMM, which is used to model the augmented source and target feature vectors, is replaced by an RBM, which can well capture the inter-dimensional and inter-speaker correlations with the joint spectral features [6]. Conditional restricted Boltzmann machine (CRBM), which can perform linear and non-linear transformations simultaneously, is also employed to voice conversion for a robust transformation function [7].

Cross-lingual voice conversion is a challenge to the task where source and target languages are different. When the parallel utterance pairs are not available in the real application of intra-lingual voice conversion, it is commonly suggested to find the units (or frames) in the utterance (or feature vectors) of source speaker that are closest to those of target speaker to generate parallel speech (or feature) pairs [8], [9]. This method is also applicable to cross-lingual voice conversion. Vocal tract length normalization (VTLN), which aims to compensate for the vocal tracts of different sizes, is employed to voice conversion [4], [8]. It was supposed that the variability in inter-speaker voice characteristics caused by vocal tract length should be language

---

Copyright © 2014 The Institute of Electronics, Information and Communication Engineers
independent.

In this paper, we propose to use neural networks (NNs), which have strong learning capabilities, to cross-dialectal (Mandarin-Shanghainese) voice conversion. Shanghainese or Hu dialect is a dialect spoken in the city of Shanghai and the surrounding region. Although it is classified as part of the Sino-Tibetan family of languages, it is largely not mutually intelligible with other Chinese varieties such as Mandarin [10]. So our approach to cross-dialectal voice conversion is also applicable to cross-lingual voice conversion.

2. Cross-Dialectal Voice Conversion

A schematic diagram of our approach to cross-dialectal (Mandarin-Shanghainese) voice conversion is shown in Fig. 1.

There are two stages in our approach: training and converting.

In the training stage, a parallel database, which contains source and target speakers’ Mandarin recordings in the same sentences, is needed. The input speech signals, the parallel source and target speakers’ waveforms, are converted to feature vectors, e.g., spectral envelope feature, line spectrum pair (LSP), fundamental frequency (F0) and voicing strength (VS), by feature extraction module. Frequency warping, which is employed to VTLN, is used to warp the frequency axis of source speaker’s spectrum by expanding or compressing in different regions to minimize the distance to target spectrum. The utterances from source and target speakers are usually of different lengths, so dynamic time warping (DTW), which calculates an optimal match between two sequences given a similarity measure, is used to align the feature vectors of the source speaker and those of the target speaker. Each source feature vector has a corresponding target feature vector after DTW. To capture the temporal changes, dynamic (delta and delta-delta) features, i.e., first and second order time derivative, are appended to feature vectors.

Let’s define the feature vectors of source and target speakers as \( X = [X_1^T, X_2^T, \ldots, X_L^T]^T, X_i = [x_i^T, \Delta x_i^T, \Delta^2 x_i^T]^T \) and \( Y = [Y_1^T, Y_2^T, \ldots, Y_L^T]^T, Y_i = [y_i^T, \Delta y_i^T, \Delta^2 y_i^T]^T \), where \( x_i, y_i \) are static components, \( \Delta x_i, \Delta y_i \) are velocity components and \( \Delta^2 x_i, \Delta^2 y_i \) are acceleration components. The feature vectors \( X \) and \( Y \) can be calculated by \( X = Gx \) and \( Y = Gy \), where \( x = [x_1^T, x_2^T, \ldots, x_L^T]^T \) and \( y = [y_1^T, y_2^T, \ldots, y_L^T]^T \) are static feature sequences and \( G \) is a block matrix which composes of matrices: Identity matrix, delta coefficient matrix \( (G_A) \) and delta-delta coefficient matrix \( (G_{A^2}) \). \( T \) is the total number of frames in the sequence.

A multi-layer feed forward neural network is used to train the mapping functions between the source \( X \) and the target \( Y \) feature vectors. For a 3-layer (with 2 hidden layers) neural network, the mapping function is

\[
\hat{Y} = F(X) = f(h_2h_1h_0(X)) \tag{1}
\]

where \( (W, b) = (W^1, b^1, W^2, b^2, W^3, b^3) \), \( (W^l, b^l) \) represent the weight matrix and bias of the \( l \)-th layer to the \((l-1)\)-th layer of the neural network, respectively. Hyperbolic tangent (or tanh) function is used as activation function for two hidden layers,

\[
f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{2}
\]

while linear activation function is employed at the final output layer. In NNs training, all weights are trained by optimizing a cost function, i.e., the mean square error (MSE), between target feature vectors and the predicted output vectors with the back-propagation (BP) procedure [11]. The cost function is defined by

\[
D(\hat{Y}, Y) = \frac{1}{2T} \| \hat{Y} - Y \|^2 + \frac{\lambda}{2} \sum_{l,i,j} (w_{ij}^l)^2 \tag{3}
\]

where \( w_{ij}^l \) is the weight between \( j \)-th neural node of \((l-1)\)-th layer and \( i \)-th neural node of \( l \)-th layer. The second item of Eq. (3), a regularization term (also called a weight decay term [15]), is used to prevent over-fitting in the training procedure, and \( \lambda \) controls the relative importance of two items. The NNs is trained by using batch gradient descent. It is optimized by a “mini-batch” based stochastic gradient descent algorithm,

\[
(W^l, b^l) \leftarrow (W^l, b^l) + \varepsilon \frac{\partial D}{\partial (W^l, b^l)}, \quad 0 \leq l \leq L \tag{4}
\]

where \( \varepsilon \) is a preset learning rate.

In the converting stage, the feature vector \( X \) is firstly extracted from the source speaker’s utterance in Shanghainese by feature extraction module and frequency warping. Next, the warped feature vector \( X \) is mapped to the feature vector \( \hat{Y} \) with the well trained neural network model. In order to generate a smoother parameter trajectory, dynamic features are used as a constraint in the parameter generation algorithm [12] based on maximum likelihood (ML) criterion,

\[
\hat{y} = R^{-1} \Sigma^{-1} \hat{\bar{Y}} \tag{5}
\]

\[
R = G^T \Sigma^{-1} G \tag{6}
\]

where \( \hat{y} = [\hat{y}_1^T, \Delta \hat{y}_1^T, \Delta^2 \hat{y}_1^T]^T \) is the output feature vector from NNs, \( G \) and \( \Sigma \) are the dynamic feature coefficient matrix and
the covariance matrix, respectively. We set the converted features, \( \hat{Y} \), as mean vectors and pre-computed (global) variances of target features from all training data as \( \Sigma \), and feed them into a ML-based parameter generation module. Finally the converted target speech in Shanghainese is synthesized by a linear predictive coding (LPC)-based vocoder with \( \hat{y} \) (voiced and unvoiced decision by a pre-set threshold of VS).

3. Training Aspects

There are several training aspects in our approach of cross-dialectal voice conversion.

3.1 Frequency Warping

Frequency warping (FW) aims to compensate for the vocal tract of different sizes and is supposed to be language independent. A piecewise linear function, a bilinear function, or more sophisticated functions can be used to warp spectral frequencies of source speaker toward those of target speaker. Our frequency warping approach is based on a bilinear warping function,

\[
p^{\alpha}(\omega) = \omega + 2 \tan^{-1} \left( \frac{(1 - \alpha) \sin \omega}{1 - (1 - \alpha) \cos \omega} \right)
\]

where \( \omega \) is the spectral frequency for warping and \( \alpha \) is the warping factor.

A single warping factor for all the utterances of a speaker is effective in VTLN, which is generally used in speech recognition to remove individual speaker characteristics. However, all acoustic classes do not reveal the same spectral variation caused by physiological differences [9], [13]. We proposed a frequency warping approach based on a time-varying bilinear function [9] to reduce the weighted spectral distance, as

\[
D(\log p_s(\omega) - \log p_y(\omega)) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left( \log p_s(\omega) - \log p_y(\omega) \right) \left( p_s(\omega) - p_y(\omega) \right) d\omega
\]

between the spectrum of source speaker \( p_s(\omega) \) and the target speaker \( p_y(\omega) \). The parallel spectra from the source and the target speakers are used to train a GMM, where a bilinear warping factor is estimated for each Gaussian component by minimizing the weighted log spectral distance (LSD). The weight assigned to spectral distance is correlated with spectral formant, which is sensitive to the perception of speaker characteristics [14]. It also can be seen as a Kullback-Leibler distance (KLD), a measure of (dis)similarity between two probability distributions in probability and information theory, between the two spectra. The warping factor \( \hat{\alpha}_t \) for each frame of the source speaker is dynamically generated from the trained GMM, as

\[
\hat{\alpha}_t = \sum_m \left( \frac{c_m N(\omega_{tx}, \mu_{tx}, \Sigma_{tx})}{\sum_m c_m N(\omega_{tx}, \mu_{tx}, \Sigma_{tx})} \right) \alpha_m
\]

where \( c_m, \mu_{tx} \) and \( \Sigma_{tx} \) are the weight, mean vector and covariance matrix of the \( m \)-th Gaussian component of the source speaker, respectively. \( \alpha_m \), the bilinear warping factor for the \( m \)-th Gaussian component, is obtained by a full grid search by minimizing the weighted LSD between the source and the target speakers’ spectra associated with the mean vector of that component.

F0 is adjusted according to,

\[
\hat{F}_0 = \frac{(F_{0s} - u_s)}{\sigma_s} \cdot \sigma_y + u_y
\]

where \( u_s, u_y, \sigma_s \) and \( \sigma_y \) are the means and the standard deviations of the source and the target speakers, respectively. After F0 modification, the resultant \( \hat{F}_0 \) is supposed to has the same statistical distribution as that of target speaker.

3.2 Pre-training

Pre-training is crucial in training deep structured models for speech recognition tasks [16], [17] which show that pre-training can initialize the weights to a better starting point than random initialization prior to BP that allows the BP to facilitate a rapid global learning and thus Deep NNs (DNNs) have outperformed traditional shallow networks [18], [19]. DNNs can be pre-trained as a DBN [20] or “layer-wise BP” [21].

In DBN pre-training, the network is trained in a layer-by-layer manner with stacked RBMs, where each successive pair of layers is treated as an RBM. A joint configuration, \((v, h)\) of the visible and hidden units has an energy function. For visible units are real-valued and hidden units are binary, the Gaussian-Bernoulli RBM is generally employed. The energy function of \((v, h)\) is defined

\[
E(v, h) = \sum_i \frac{(v_i - a_i)^2}{2} - \sum_j b_j h_j - \sum_i \sum_j w_{ij} v_i h_j
\]

where \( v_i, h_j \) are the states of visible unit \( i \) and hidden unit \( j \), \( a_i, b_j \) are their biases and \( w_{ij} \) is the weight between them. The network assigns a probability to every possible pair of a visible and a hidden vector through the energy function, as

\[
P(v, h) = \frac{1}{z} \exp(-E(v, h))
\]

where the partition function is given by

\[
z = \int_v \sum_h \exp(-E(v, h)) dv
\]

The weights \((w, a, b)\) that connect each pair of layers are trained in an unsupervised fashion by maximum likelihood estimation using contrastive divergence (CD) algorithm [22].

Alternatively, the network can be initialized using “layer-wise BP” pre-training. This procedure starts by training a multi-layer neural network with one hidden layer using back-propagation. The weights of the first hidden layer
are fixed, a new randomly initialized hidden layer is added into the network and output layer is introduced to replace the output layer of the initial network. The deeper network is then trained again using back-propagation. This procedure is repeated until the desired number of hidden layers is reached.

Pre-training can be used to learn a compressed, distributed representation for a set of data. In DBN pre-training, it maximizes the likelihood \( L = \prod_i P(v) = \prod_i \sum_h P(v, h) \) over training samples \( v(t) \). It learns a set of features to reconstruct the input features in an unsupervised way. In “layer-wise BP” pre-training, it minimizes cost function \( D(\tilde{X}, X) \), i.e., the discrepancy between \( F(X) \) and \( X \). It is supervised trained by BP. Pre-training is supposed to discover interesting structure about the data. If there is structure in the data, for example, if some of the input features are correlated, then this algorithm will be able to discover some of those correlations.

In real applications, it is not so difficult to obtain individual speaker’s utterances compared to those parallel utterances of source and target speakers. Pre-training can be used to learn the feature structures of source and target speakers, separately, or it can be seen as a kind of feature transformation, to facilitate the mapping function learning between source and target speakers with parallel utterances. A similar work is employed in [5] by using DBN to build high-order eigen space for voice conversion.

### 3.3 Sequence Training

Sequence training is employed to consider the sequence constraints from HMMs, dictionary, and language model, which are the LVCSR decoding objective, in DNN training for speech recognition. Sequence training, also called sequence discriminative training, e.g., maximum mutual information (MMI) and minimum Bayes risk (MBR) sequence training [23], [24], can significantly improve the performance of speech recognition. The gain of sequence training is mainly benefited from matched objectives used in training and decoding.

The neural network mentioned in Sect. 2 is trained with the error back-propagation using the frame-based objective MSE. Each output frame feature vector \( \tilde{Y} \) from NNs contains static and dynamic features. However, the final features used to generate target speaker’s utterance is the features \( \hat{y} \), which is got from a parameter generation algorithm as Eq. (5). To seek a better matched training criterion for converting, the cost function should be

\[
D(\hat{y}, y) = \frac{1}{2T} || \hat{y} - y ||^2 + \frac{\lambda}{2} \sum_{i,j} (w_{i,j})^2
\]  

(14)

The similar work, named minimum generation error (MGE) training [25], [30], was applied in HMM-based speech synthesis for adjusting the model parameters, means and variances, to minimize the generation error between synthesized and original parameter trajectories of the training data. Eq. (8) is closer to the perception of speaker characteristics than Eq. (14). However, the computational complexity of Eq. (8) between two spectra is high, so we use a weighted LSP distance to approximate it, as follows,

\[
\phi_{i,j} = s_i^2 \left( \frac{1}{y_{i,j} - y_{i,j-1}} + \frac{1}{y_{i,j+1} - y_{i,j}} \right)
\]  

(15)

where \( y_{i,j} \) are the \( i \)-th order of the \( r \)-th frame feature of target speaker and \( s_i \) is manually set to enhance the contribution of low LSP dimensions [31]. The Eq. (14) is modified as

\[
D_\phi(\hat{y}, y) = \frac{1}{2T} \sum_{i,j} \phi_{i,j}(\hat{y}_{i,j} - y_{i,j})^2 + \frac{\lambda}{2} \sum_{i,j} (w_{i,j})^2
\]  

(16)

The derivatives of neural network output can be taken based on chain rule,

\[
\frac{\partial D_\phi(\hat{y}, y)}{\partial \hat{Y}} = 2\Sigma^{-1} G R^{-1} (\hat{y} - y)^T
\]  

(17)

where \( \Phi = \text{diag}(\Phi_1, \Phi_2, \ldots, \Phi_T) \) and \( N \) is total number of LSP dimensions. Similarly, the derivatives of covariance matrix is

\[
\frac{\partial D_\phi(\hat{y}, y)}{\partial \Sigma^{-1}} = 2(\hat{y} - G \hat{y})(\hat{y} - y)^T \Phi R^{-1} G^T
\]  

(18)

The derivatives of the error function with respect to the network outputs can be used in standard back-propagation algorithms to find a minimum of the error function.

Mini-batch algorithms are widely used in neural network training as a way to speed-up stochastic convex optimization problems. However, it is not suitable for sequence training, where the size of batch for one iteration is equal to the length of one sentence. As to randomization, which is usually used for avoiding local minimum during NNs training, is also performed on the sentence level instead of conventional frame level. Compared with frame-level based MSE training, sequence training has a relatively larger computational cost because of the calculation of the inverse of \( R \).

### 4. Experiments and Results

#### 4.1 Experimental Setup

Three speakers’ recordings, listed in Table 1, are used in our experiments. A bi-dialect (Mandarin and Shanghainese) male speaker’s (Spk A) recordings are used as the source speaker’s data. The recordings from a female speaker (Spk C), who is Mandarin speaker and learning Shanghainese, and a male speaker (Spk B), who is Mandarin speaker and cannot speak Shanghainese, are used as Target speakers’

<table>
<thead>
<tr>
<th>Speaker</th>
<th>Data set</th>
<th>Training</th>
<th>Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source A (male)</td>
<td>M</td>
<td>100</td>
<td>50</td>
</tr>
<tr>
<td>Target B (male)</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Target C (female)</td>
<td>100</td>
<td>50</td>
<td>50</td>
</tr>
</tbody>
</table>
The training data contains 100 parallel utterances in Mandarin (M) from those three speakers and 100 parallel utterances in Shanghainese (S) from Spk A and C. Testing data is composed of 50 utterances in Mandarin and 50 utterances in Shanghainese from both Spk A and Spk C. Speech signals in those three speakers’ recordings are sampled at 16 kHz, windowed by a 25-ms window with a 5-ms shift. Spectral envelopes are estimated by STRAIGHT [26]. The LPCs of 40th order are transformed into line spectral pairs (LSP). F0 and voicing strength (VS) are extracted by the ESPS robust pitch tracing algorithm [27]. An exponential decay function [28] is used to interpolate F0 in unvoiced speech regions. 40th order LSP, log F0, VS, their first order and second order time derivatives are employed as one frame feature vector. For NNs training, the contextual frames, i.e., 2 left and 2 right, are appended to current frame for input feature vector, total 645 dimensions, while the output feature vector of NNs only contains the current frame, total 129 dimensions.

We run the experiments with many configurations to evaluate A. The performance of cross-dialectal voice conversion by using a bi-dialect speakers’ recording; B. whether frequency warping is beneficial to cross-dialectal voice conversion with NNs; C. whether pre-training is effective in NN-based voice conversion; D. the performance of sequence training in cross-dialectal voice conversion.

4.2 Evaluations and Results

4.2.1 Objective Measures

Objective measures are first used to evaluate the performance of voice conversion systems on testing data. Conversion quality is measured objectively in terms of distortions between natural test utterances of the original speaker and the converted speech frame-synchronously. The objective measures are F0 distortion in the root mean squared error (RMSE) and spectrum distance in log spectral distance (LSD).

A. Cross-dialectal voice conversion

We examine the performance of our proposed inter-dialectal voice conversion by comparing with intra-dialectal voice conversion. We input Spk C’s testing utterances in both Mandarin and Shanghainese into NNs, which is trained by only parallel utterances in Mandarin of Spk A and C, for voice conversion. Figure 2 shows the average LSD and F0 RMSE of the Mandarin and Shanghainese testing utterances of the original target speaker (Spk C) and those generated by converted source speakers (Spk A) by NN-based voice conversion with various numbers (10, 20, 50 and 100) of parallel Mandarin utterances for training. In the training, NNs architectures, e.g., the number of layers and the number of nodes per layer, are adjusted to get the optimal results. Figure 2 indicates that the performance of intra-dialectal voice conversion is better than that of inter-dialectal voice conversion, but with more training utterances, say 100, the gap between intra-dialectal and inter-dialectal voice conversion is small. Therefore, we think the NNs trained by only parallel Mandarin utterance is capable of converting the source speaker’s Shanghainese speech to target speaker’s Shanghainese speech.

B. Frequency warping (FW)

We evaluate the performance of frequency warping in NN-based cross-dialectal voice conversion. We use 10, 20, 50, and 100 parallel Mandarin utterances of Spk A and C to train NNs. Before the training, the spectra of Spk A is warped toward that of Spk C. To make the comparison valid, we also use 10, 20, 50, and 100 aligned same utterance pairs to train a GMM for getting time-varying warping factors, as mentioned in Sect. 3.1. The resultant number of Gaussian components are 2, 5, 8, and 12, respectively. F0 is also modified according to Eq. (10). In the testing phase, the spectra of Spk A’s Shanghainese utterance is first warped by a bilinear warping function with estimated warping factor, then the corresponding features: LSP, log F0 and VS is converted by well-trained NNs, finally the converted features are synthesized to waveform. Figure 3 shows the average LSD and F0 RMSE of the testing utterances of the original target speaker (Spk C) and those generated by converted source speakers (Spk A) by NNs, which is trained by various numbers (10, 20, 50 and 100) of parallel utterances with/without frequency warping. The frequency warping is beneficial to cross-dialectal voice conversion with NNs when the number
The average LSD and F0 RMSE of cross-dialectal voice conversion, which is trained by various numbers (10, 20, 50 and 100) of parallel Mandarin utterances with/without frequency warping.

of parallel training utterance pairs is small, say 10. But the NNs seem to be able to learn the variability caused by vocal tract length if it is trained with enough utterances. In addition, normalizing F0s of source speaker to the same distribution as that of target speaker is also useful when a few of the parallel utterances are available for training. In the following experiments, we do not use spectral frequency warping and F0 normalizing when the number of parallel training utterance pairs is larger than 20.

C. Pre-training

We fix the architecture of NNs, i.e., 3 hidden layers with 1024 nodes per layer, which gets the best performance by using 100 parallel Mandarin utterances, to test the efficiency of pre-training. The source and target speakers are Spk A and Spk C, respectively. The configurations of pre-training are listed as

a) Pre-training by DBN with 100 Mandarin utterances of Spk A
b) Pre-training by layer-wise BP with 100 Mandarin utterances of Spk A
c) Pre-training by layer-wise BP with 100 Mandarin utterances and 100 Shanghainese utterances of Spk A
d) Concatenate two NNs, which are pre-trained with 100 Mandarin utterances of Spk A and 100 Mandarin utterances of Spk C, respectively. The details can be referred to [5]. But we use layer-wise BP instead of DBN used in [5].

The average LSD and F0 RMSE of the Shanghainese testing utterances of the original target speaker (Spk C) and those converted source speakers (Spk A) by NNs without pre-training or with pre-training by above 4 configurations are shown in Table 2. It shows that the performance (on LSD and F0 RMSE) of NNs with pre-training is better than that of without pre-training, and the layer-wise BP pre-training outperforms the DBN pre-training. We think it is due to that the criterion of layer-wise BP pre-training is consistent with that of NNs training. The cost function in Eq. (3) is to minimize MSE between target and converted (predicted) vectors. In DBN pre-training, although NNs weights are pre-trained in a generative manner, an approximate maximum likelihood criterion of contrastive divergence is still used to learn the model parameters. While in layer-wise BP, the hidden layers are added to the neural networks one by one to full convergence, it can remedy the modeling criterion mismatch in DBN pre-training.

The best performance of pre-training is got by above configuration c), which uses the utterances in both Mandarin and Shanghainese for pre-training. Inspiring by it, we build a NNs to convert both Mandarin and Shanghainese by using both parallel Mandarin and Shanghainese utterances. It is generally needed to build two NNs for Mandarin and Shanghainese, respectively. The architecture of NNs is illustrated in Fig. 4, where the input layer still contains 645 nodes, while the output layer is composed of 258 nodes: 129 for Mandarin and the rest for Shanghainese. 200 parallel utterances in both Mandarin and Shanghainese are used to train the NNs. The hidden layers are shared while the output layer is separated by Mandarin and Shanghainese conversion tasks. The average LSD and F0 RMSE of intra-dialectal voice conversion by different NNs architectures are listed in Table 3, which shows that the combined NNs, as Fig. 4, can improve the performance of NNs trained for Mandarin and Shanghainese voice conversion tasks separately.
Table 3 The average LSD and F0 RMSE of intra-dialectal voice conversion by different NNs architectures.

<table>
<thead>
<tr>
<th></th>
<th>Two NNs</th>
<th>Combined NNs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>M</td>
<td>S</td>
</tr>
<tr>
<td>LSD (dB)</td>
<td>5.73</td>
<td>5.72</td>
</tr>
<tr>
<td>F0 RMSE</td>
<td>29.3</td>
<td>24.5</td>
</tr>
</tbody>
</table>

Table 4 The average LSD and F0 RMSE of cross-dialectal voice conversion without sequence training (Eq. (3)) or with sequence training by the cost functions in Eq. (14) or Eq. (16).

<table>
<thead>
<tr>
<th>Cost functions</th>
<th>Eq. 3</th>
<th>Eq. 14</th>
<th>Eq. 16</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSD (dB)</td>
<td>6.07</td>
<td>5.91</td>
<td>5.92</td>
</tr>
<tr>
<td>F0 RMSE</td>
<td>28.7</td>
<td>27.9</td>
<td>27.9</td>
</tr>
</tbody>
</table>

D. Sequence training

To test sequence training for dialectal voice conversion, we use the architecture of NNs (3 hidden layers with 1024 nodes per layer) and 100 parallel Mandarin training utterances to evaluate Shanghainese testing utterances. The source and target speakers are Spk A and Spk C, respectively. The NNs are first initialized by the pre-training (Configuration c), then trained by frame-level MSE, Eq. (3), and finally refined by sequence training with cost functions in Eq. (14) or Eq. (16). Table 4 shows the average LSD and F0 RMSE of cross-dialectal voice conversion without sequence training (Eq. (3)) or with sequence training by the cost functions in Eq. (14) or Eq. (16). It shows that the sequence training is very effective to decrease the average LSD and F0 RMSE of the Shanghainese testing utterances of the original target speaker (Spk C) and those generated by converted source speakers (Spk A) by NNs. However, the performance of cost function in Eq. (16) is on par with that in Eq. (14).

4.2.2 Subjective Measures

We synthesize the Shanghainese testing utterances to further evaluate our approach by subjective listening tests. To avoid the over-smoothing problem of parameter trajectories converted by NNs and the resultant buzzy speech, formant sharpening based on LSP frequencies [29] are used for LSP generation.

The performance of frame-level MSE training, pre-training and sequence training are subjectively evaluated by perceptual tests. The architecture of 3 hidden layer with 1024 nodes per layer and 100 parallel Mandarin training utterances are used to train NNs. 50 Shanghainese utterances, which are converted by cross-dialectal voice conversion with/without pre-training (Configuration b and c), with/without sequence training (Eqs. (14) and (16)) and randomly selected from Spk A to Spk B and Spk A to Spk C conversions, are evaluated in four AB preference tests through crowdsourcing. Each test is participated by 20 subjects, who claim native Shanghainese speaker. Each subject evaluates 50 pairs by using headsets. There are three choices: 1) the former is better; 2) the latter is better; 3) no preference or neutral (The difference between the paired sentences cannot be perceived or can be perceived but difficult to choose which one is better). The preference scores are shown in Table 5. It supports most of our objective measure results. The AB preference test results show the sequence training by using Eq. (16), which minimizes the weighted LSP sequence errors, performs better than using Eq. (14) by comparing with frame-level MSE training, while these two sequence trainings are on a par on objective measures. We conjecture that putting more weights on spectral frequencies associated with “formants”, which is close to the perception of voice characteristics, than other frequencies may not be able to directly affect the objective measure of LSD.

The subjective measures of mean opinion scores (MOS) for speaker similarity and naturalness are also used for evaluating same testing utterances as in AB preference tests by crowdsourcing. The speaker similarity MOS measures how close the two utterances are: one is from the original target speaker and the other is synthesized speech by converted feature parameters. The naturalness MOS indicates the voice quality of the converted speech. The MOS is expressed as a number ranging from 1 to 5, where 1 is the lowest and 5 is the highest. The MOS results of similarity and naturalness for the Shanghainese utterances converted by NNs trained by 100 parallel Mandarin utterance (M_{100}) with pre-training (M_{100,pre}) and sequence training (M_{100,pre,Seq}) are shown in Fig. 5 and Fig. 6, respectively. They show that the pre-training and sequence training can significantly improve (at 95% confidence intervals) the similarity and the naturalness of converted Shanghainese speech in cross-dialectal voice conversion.
conversion system, i.e., average MOS can be improved from 2.85 to 3.20 and from 2.75 to 3.05 on similarity and naturalness evaluations.

5. Conclusions

This paper has extended neural networks based voice conversion to a cross-dialectal task. Frequency warping, pre-training and sequence training are studied in this task. NNs can also perform well in cross-dialectal voice conversion. Both objective and subjective measures indicate that pre-training and sequence training can significantly improve the performance of cross-dialectal voice conversion. In the future, we will test our approach to the voice conversion of cross language, say Mandarin and English, which are rather different due to belonging to different language families.

References


Weixun Gao received the B.S. and M.S. degrees in the department of Computer Science from Shanghai Normal University and the College of Software from Fudan University, China in 1995 and 2007, respectively. He now is with School of Information Science and Technology, Donghua University, Shanghai, China.

Qiying Cao now is a professor in the College of Computer Science & Technology, Donghua University, Shanghai, China. His research interests include ubiquitous computing, intelligent information processing, network and information security, embedded technology and network home appliances and computer-aided technologies.

Yao Qian received her BS and MS degrees in Computer Science and Linguistics from Shanghai Normal University, Shanghai, China, in 1995 and 2002, and Ph.D. degree in Electrical Engineering from the Chinese University of Hong Kong, Hong Kong, in 2005, respectively. In 2005, she joined in speech group of Microsoft Research Asia (MSRA), Beijing, China. Her recent research projects include speech synthesis, voice transformation, prosody modeling and computer-assisted language learning.