INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Spectro-Temporal Modelling with Time-Frequency LSTM and Structured Output Layer for Voice Conversion

Runnan Li[1], *Zhiyong Wu*[1,2], *Yishuang Ning*[1], *Lifa Sun*[2], *Helen Meng*[1,2], *Lianhong Cai*[1]

[1]MJRC, Graduate School at Shenzhen, Tsinghua University, China
[2]Dept. of Systems Engineering & Engineering Management, CUHK, Hong Kong SAR, China

{lirn15, ningys13}@mails.tsinghua.edu.cn,
{zywu, lfsun, hmmeng}@se.cuhk.edu.hk, clh-dcs@tsinghua.edu.cn

Abstract

From speech, speaker identity can be mostly characterized by the spectro-temporal structures of spectrum. Although recent researches have demonstrated the effectiveness of employing long short-term memory (LSTM) recurrent neural network (RNN) in voice conversion, traditional LSTM-RNN based approaches usually focus on temporal evolutions of speech features only. In this paper, we improve the conventional LSTM-RNN method for voice conversion by employing the two-dimensional time-frequency LSTM (TFLSTM) to model spectro-temporal warping along both time and frequency axes. A multi-task learned structured output layer (SOL) is afterward adopted to capture the dependencies between spectral and pitch parameters for further improvement, where spectral parameter targets are conditioned upon pitch parameters prediction. Experimental results show the proposed approach outperforms conventional systems in speech quality and speaker similarity.

Index Terms: Voice conversion, time-frequency long short term memory (TFLSTM), structured output layer (SOL)

1. Introduction

Voice conversion (VC) aims at learning the complex non-linear relationship of the acoustic features between source and target speakers. Lots of methods have been proposed for the task. Gaussian mixture model (GMM) based probabilistic approach [1] assumes acoustic features have a random component that can be reasonably described, which is further improved by exploiting global variance (GV) [2] to alleviate the over-smoothing problem. Non-negative matrix factorization (NMF) based approach [3] uses speech exemplars to synthesize target speech directly rather than convert the acoustic features.

More recently, inspired by the successful applications in automatic speech recognition (ASR) [4] and text-to-speech (TTS) synthesis [5], neural network (NN) based approaches have been increasingly popular in VC. [6] employed artificial neural network (ANN) to replace GMM to map the source and target features in high order space. [7] further improved ANN with deep neural network (DNN). To learn temporal context dependency across acoustic features, [8] proposed to employ deep bidirectional recurrent neural networks with long short-term memory (DBLSTM) that outperforms DNN based methods with the ability in capturing long-term dependencies.

In typical NN based models, the inputs are usually log-filter-bank features that are regarded as independent of each other [9]. Switching the positions of the features from any two filter-banks will not affect the overall performance of the network. However, in human spectrogram reading, phoneme prediction is relied on both patterns evolving in time and frequency axes. Just like switching any two frames destroys the time-wise pattern, switching the positions of any two banks will destroy the frequency-wise pattern. Moreover, formant structure determined by articulators has a particular distribution pattern over frequency axis that contributes to speaker timbre perceiving. These observations inspire us to employ a novel structure to learn time and frequency dynamics simultaneously.

In statistical parametric systems, pitch parameters are used to represent the state of the vocal folds and the spectral features are those associated with the articulators. While vocal folds and articulators are highly cooperated in human speech production [10], the dependency between these two features is valued to model. Traditionally, these features are produced by two independent subsystems or predicted as concatenated vectors by the output layer of models. Two problems arise from this. First, both approaches are difficult in modelling the dependency of spectral features on pitch contour parameters. Second, due to the unbalance in dimensionality between spectral and pitch features, the contribution of pitch parameter prediction in gradient statistics accumulated at the intermediate hidden layer is unduly suppressed. Therefore, it is preferable to employ a novel output layer in voice conversion system to exploit the correlation between spectral and pitch targets while reasonably balancing the error costs from prediction tasks in training.

This paper proposes the use of time-frequency LSTM (TFLSTM) [11] and structured output layer (SOL) [12] to address the above mentioned issues. Inheriting the properties of multidimensional LSTM, TFLSTM can model spectro-temporal dependencies across sequences by scanning time and frequency axes simultaneously. Using SOL as output layer allows the joint optimization and prediction of spectral and pitch targets, with an explicit dependency of spectral targets on pitch targets. To appropriately balance the error cost functions associated with spectral and pitch features, multi-task learning [13][14] is employed to train the proposed models.

2. Voice Conversion with Time-Frequency LSTM and Structured Output Layer

2.1. Time-frequency LSTM (TFLSTM) RNNs

Multidimensional LSTM is first proposed in [15] and [16] for handwriting recognition. [11] further optimized the structure by simplifying the multiple forget gates and memory units to a single forget gate and a single memory unit to significantly reduce the complexity.

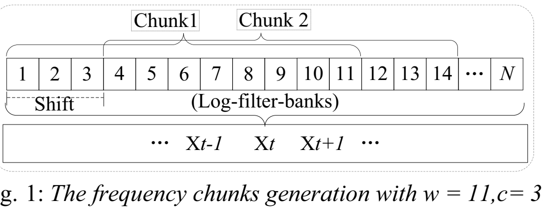

Fig. 1: *The frequency chunks generation with w = 11,c= 3.*

http://dx.doi.org/10.21437/Interspeech.2017-1122

The optimized structure, named time-frequency LSTM (TFLSTM), uses frequency chunks as inputs. As shown in Fig.1, for input $\mathbf{X}$ at time step t containing an N-dimensional vector of log-filter-bank values, the frequency chunks are generated by dividing the log-filter-banks into overlapped chunks with window-length w and window-shift c, leading $M = (N - w + c)/c$ chunks in each frame at time t.

The structure of TFLSTM cell is shown in Fig. 2, where ϕ denotes the tanh activation function. Same as the conventional LSTM, a TFLSTM cell contains one input gate, one forget gate, one memory unit and one output gate. However, each gate or unit now has two indices instead of one: frequency chunk k and time t. The formulation of the TFLSTM is as follows:

$$\mathbf{i}_{k,t} = \sigma(\mathbf{W}_{\mathrm{xi}}\mathbf{x}_{k,t} + \mathbf{W}^1_{\mathrm{hi}}\mathbf{h}_{k,t-1} + \mathbf{W}^2_{\mathrm{hi}}\mathbf{h}_{k-1,t} + \mathbf{W}_{\mathrm{ci}}\mathbf{c}_{k,t-1} + \mathbf{b}_{\mathrm{i}}) \quad (1)$$

$$\mathbf{f}_{k,t} = \sigma(\mathbf{W}_{\mathrm{xf}}\mathbf{x}_{k,t} + \mathbf{W}^1_{\mathrm{hf}}\mathbf{h}_{k,t-1} + \mathbf{W}^2_{\mathrm{hf}}\mathbf{h}_{k-1,t} + \mathbf{W}_{\mathrm{cf}}\mathbf{c}_{k,t-1} + \mathbf{b}_{\mathrm{f}}) \quad (2)$$

$$\mathbf{c}_{k,t} = \mathbf{f}_{k,t} \cdot \mathbf{c}_{k,t-1} + \mathbf{i}_{k,t} \cdot \tanh(\mathbf{W}_{\mathrm{xc}}\mathbf{x}_{k,t} + \mathbf{W}^1_{\mathrm{hc}}\mathbf{h}_{k,t-1} + \mathbf{W}^2_{\mathrm{hc}}\mathbf{h}_{k-1,t} + \mathbf{b}_{\mathrm{c}}) \quad (3)$$

$$\mathbf{o}_{k,t} = \sigma(\mathbf{W}_{\mathrm{xo}}\mathbf{x}_{k,t} + \mathbf{W}^1_{\mathrm{ho}}\mathbf{h}_{k,t-1} + \mathbf{W}^2_{\mathrm{ho}}\mathbf{h}_{k-1,t} + \mathbf{W}_{\mathrm{co}}\mathbf{c}_{k,t} + \mathbf{b}_{\mathrm{o}}) \quad (4)$$

$$\mathbf{h}_{k,t} = \mathbf{o}_{k,t} \cdot tanh(\mathbf{c}_{k,t}) \quad (5)$$

where $\mathbf{i}_{k,t}$, $\mathbf{f}_{k,t}$, $\mathbf{c}_{k,t}$, $\mathbf{o}_{k,t}$, $\mathbf{h}_{k,t}$ denote the activation vectors of input gate, forget gate, memory unit, output gate and hidden output at frequency chunk k time t. $\mathbf{W}$ terms are the weight matrices connecting different vectors: $\mathbf{W}_{\mathrm{xi}}$ denotes the weight matrix connecting input vectors $\mathbf{x}_{k,t}$ and input gate; $\mathbf{W}^1_{\mathrm{h\cdot}}$ and $\mathbf{W}^2_{\mathrm{h\cdot}}$ denote the weight matrices connecting to $\mathbf{h}_{k,t-1}$ and $\mathbf{h}_{k-1,t}$, the hidden output from the same frequency chunk at previous time step and previous frequency chunk at the same time step, respectively. $\mathbf{b}$ terms are the bias vectors.

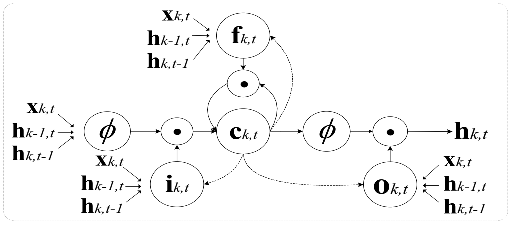

Fig. 2: *Structure of a TFLSTM cell at frequency chunk k time t. TFLSTM accepts $h_{k,t-1}$, $h_{k-1,t}$ and $x_{k,t}$ as inputs to compute current hidden $h_{k,t}$ following Eqs. (1) to (5).*

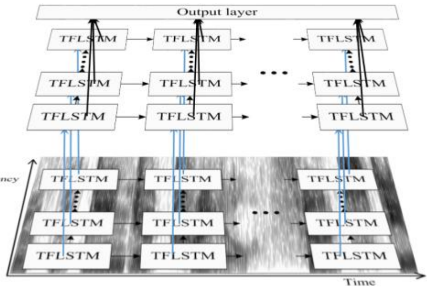

Fig. 3: *An example of stacked 2-layers TFLSTM.*

At each time step t, the hidden outputs $\mathbf{h}_{k,t}$ with $k = 1 \dots M$ can be merged into one super-vector $\mathbf{h}_t$ as a trajectory of time-frequency patterns. A deep TFLSTM can be built by stacking multiple TFLSTM hidden layers. The layer l will use the hidden output from the lower layer $\mathbf{h}^{l-1}_{k,t}$ instead of $\mathbf{x}_{k,t}$ in Eqs. (1) to (5) to generate the output $\mathbf{h}^l_{k,t}$. An example of 2-layers TFLSTM is shown in Fig. 3.

TFLSTM can also be extended to bidirectional for modelling temporal dependencies in both preceding and succeeding directions [17]. This can be done by processing input in both forward and backward directions using separate hidden layers and then feeding forward to the upper layer. Combining the deep TFLSTM and bidirectional TFLSTM, the deep bidirectional TFLSTM (DBTFLSTM) is proposed.

2.2. Structured output layer (SOL) TFLSTM RNNs

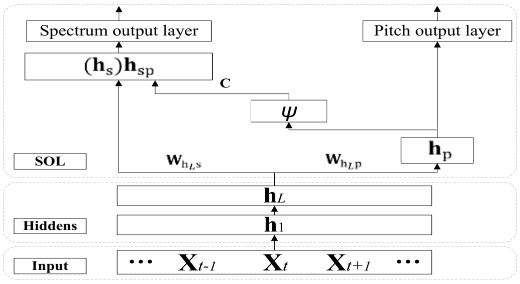
Spectrum output layer
Pitch output layer
$(\mathbf{h}_s)\mathbf{h}_{sp}$
C
ψ
SOL
$\mathbf{W}_{\mathrm{h}_L\mathrm{s}}$
$\mathbf{W}_{\mathrm{h}_L\mathrm{p}}$
$\mathbf{h}_\mathrm{p}$
$\mathbf{h}_L$
Hiddens
$\mathbf{h}_1$
Input
$\cdots\ \mathbf{X}_{t-1}\ \mathbf{X}_t\ \mathbf{X}_{t+1}\ \cdots$

Fig. 4: *The overall structure of the proposed model with SOL. Tasks share the same hidden representations and the spectrum prediction can benefit by using the hidden layer output of pitch prediction.* $\mathbf{X}$ *are the source acoustic feature inputs.*

In conventional VC systems, target spectral features and pitch parameters are either produced by two independent subsystems or predicted as concatenated acoustic features by a single output layer. However, these two approaches both have difficulty in modelling the dependencies of spectral features on pitch parameters. To address the issue, this paper proposes the use of structured output layer (SOL), in which pitch parameter prediction is used not only as a regularization during training but also as an auxiliary predictor in spectral feature prediction. The overview of the proposed structure is shown in Fig. 4.

In SOL, spectral feature prediction is set as the main task and conditioned upon the auxiliary task of pitch parameter prediction. This is realized by feeding the pitch parameter generation task's hidden layer output $\mathbf{h}_p$ through an activation function $\psi(\cdot)$ (e.g. Sigmoid or ReLU) modelling the correlation between the two tasks before being augmented to the hidden layer output $\mathbf{h}_s$ after the weight matrix $\mathbf{C}$ used to connect the two tasks is applied.

In the conventional multi-task formulation where no between task dependency is modelled, the two tasks share the same hidden layers $\{\mathbf{h}_1, \dots, \mathbf{h}_L\}$ and the prediction of spectral and pitch parameters are computed as follows:

$$\mathbf{h}_\mathrm{s} = (\mathbf{W}_{\mathrm{h}_L\mathrm{s}}\mathbf{h}_L + \mathbf{b}_\mathrm{s}) \quad (6)$$

$$\mathbf{O}_\mathrm{s} = \sigma_\mathrm{s}(\mathbf{h}_\mathrm{s}) \quad (7)$$

$$\mathbf{h}_\mathrm{p} = (\mathbf{W}_{\mathrm{h}_L\mathrm{p}}\mathbf{h}_L + \mathbf{b}_\mathrm{p}) \quad (8)$$

$$\mathbf{O}_\mathrm{p} = \sigma_\mathrm{p}(\mathbf{h}_\mathrm{p}) \quad (9)$$

where $\mathbf{O}_\mathrm{s}$ and $\mathbf{O}_\mathrm{p}$ are predicted spectral and pitch parameter outputs respectively. $\{\mathbf{W}_{\mathrm{h}_L\mathrm{s}}, \mathbf{b}_\mathrm{s}\}$ and $\{\mathbf{W}_{\mathrm{h}_L\mathrm{p}}, \mathbf{b}_\mathrm{p}\}$ are the weight matrices and bias vectors connecting the shared hidden layer $\mathbf{h}_L$ with the outputs associated with the two tasks. $\sigma_\mathrm{s}(\cdot)$ and $\sigma_\mathrm{p}(\cdot)$ are the linear output activation functions employed to produce the final predicted spectral feature and pitch parameter outputs.

In contrast, the proposed SOL based approach shown in Fig. 4 introduces an additional dependency of the primary spectrum prediction task on the auxiliary pitch parameter prediction task. The main spectral feature outputs are thus modified as:

$$\mathbf{h}_\mathrm{sp} = (\mathbf{W}_{\mathrm{h}_L\mathrm{s}}\mathbf{h}_L + \psi(\mathbf{h}_\mathrm{p})\mathbf{C} + \mathbf{b}_\mathrm{sp}) \quad (10)$$

$$\mathbf{O}_\mathrm{s} = \sigma_\mathrm{s}(\mathbf{h}_\mathrm{sp}) \quad (11)$$

2.3. Multi-task learning of SOL TFLSTM RNNs

In common with the conventional multi-task learning (MTL) framework, networks with structured output layer (SOL) can be trained by minimizing a global cost function expressed as a weighted sum of the two task-specific error costs as:

$$F_g = \alpha F_s + (1-\alpha)\,F_p \tag{12}$$

where F_s and F_p are the costs generated by the main task (spectral feature prediction) and the auxiliary task (pitch parameter prediction) computed as mean squared errors, and the global error cost in (12) can be re-expressed as:

$$F_g = \frac{1}{NT}\sum_1^N\sum_1^T[\alpha(O_s - s)^2 + (1-\alpha)\big(O_p - p\big)^2] \tag{13}$$

The gradients used to update the parameters θ in the SOL employed network are then computed as the weighted average gradient statistics computed over both tasks:

$$\frac{\partial F}{\partial \theta^k} = \frac{1}{NT}\sum_1^N\sum_1^T[\alpha\frac{\partial}{\partial \theta_s^k}(O_s - s)^2 + (1-\alpha)\frac{\partial}{\partial \theta_p^k}\big(O_p - p\big)^2] \tag{14}$$

With SOL, the proposed model can earn stronger performance and robustness in voice conversion facilitated by shared hidden layers and jointly training over multiple tasks. The use of a structured output layer can further exploit the regularization properties of the comparatively simpler auxiliary task of pitch prediction and its direct effect on the primary spectral feature generation task.

2.4. Framework of the voice conversion system

The overall architecture of the proposed voice conversion system is shown in Fig.5. In training stage, spectral features and pitch parameters, i.e. log F0 contour and voiced/unvoiced flag (V/UV), are extracted and sent for time alignment with dynamic time warping (DTW) procedure. DTW is set as unconstrained and the distance measure employed is the square root of the minimum sum of squared differences divided by the number of comparisons computing on spectral features. After being normalized to zero mean and unit variance, the time-aligned features are then fed into the network-based model for training. In conversion stage, acoustic features extracted from source speeches are fed directly to the trained model to predict target acoustic features. A vocoder is then used to synthesize the converted speech using the predicted features.

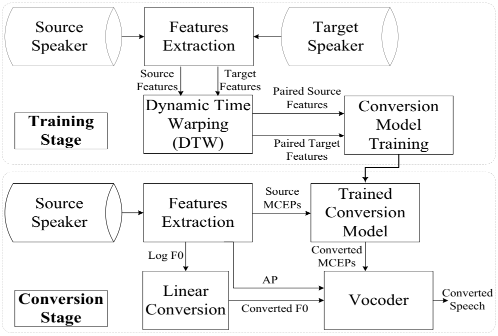

Fig. 5: *Overview of the voice conversion framework.*

3. Experiments

3.1. Experimental setup

The evaluation is conducted on the CMU ARCTIC parallel speech corpus. CLB (a female speaker from U.S) is used as the source speaker, and RMS (a male speaker from U.S) as the target speaker. The 1,132 parallel utterances are divided into three sets: the first 1,000 parallel utterances as the training set, the following 100 utterances as the validation set and the rest 32 utterances as the test set. The acoustic signals are sampled at 16kHz with mono channel. Acoustic features including 35-dimensional Mel-cepstral coefficients (MCEPs), 2 dimensional pitch parameters (log F0 and V/UV), 25-dimensional aperiodic component (AP) are extracted using STRAIGHT [18] with 25-ms frame length window and 5-ms frame shift. For training and validation sets, source and target feature sequences are aligned using DTW. Six different models with similar parameter numbers are implemented for comparison:

- **LSTM**: Conventional LSTM based approach containing one LSTM hidden layer with 1024 nodes.
- **TFLSTM**: TFLSTM based approach containing one TFLSTM hidden layer. The window-length and shift of frequency chunk is set as 11 and 3 respectively, thus $(35 - 11 + 3)/\,3 = 9$ chunks are fed into TFLSTM. Therefore, 9 memory cells are implemented for TFLSTM, each with 230 nodes.
- **DBLSTM**: DBLSTM based approach containing two bidirectional LSTM hidden layers with 672 nodes per layer (336 forward nodes and 336 backward nodes).
- **DBTFLSTM**: DBTFLSTM based approach containing two bidirectional TFLSTM hidden layers using the same settings as the above TFLSTM approach and each cell has 100 forward or backward nodes.
- **DBLSTM-SOL**: DBLSTM based approach with a structured output layer. The weight α used in (12) is 0.925, and tanh function is used as activation function $\psi(\cdot)$. Selection of $\psi(\cdot)$ and α is elaborated in the next section.
- **DBTFLSTM-SOL**: DBTFLSTM based approach with a structured output layer. The settings for $\psi(\cdot)$ and α is the same as the above DBLSTM-SOL approach.

Models without SOL will generate 35-dimensional output containing MCEPs only, while SOL derived models will generate 37-dimensional output containing MCEPs and pitch parameters. It should be noted this paper focuses on improving the performance of spectral feature conversion. Hence, to be fair for different models, although the source pitch parameters are needed in SOL derived models, the predicted pitch parameters from SOL are just ignored, and instead, the traditional linear prediction (LP) conversion method is used for converting pitch parameters. STRAIGHT vocoder is then employed to synthesize the converted speech using the converted MCEPs from aforementioned models, the generated pitch parameters from LP conversion and the original AP.

Models are trained using back-propagation through time (BPTT) [19] by unfolding RNNs into standard feed-forward networks through time steps. Keras [20] with Theano [21] as the backend is used to implement the above systems using mini-batch-based Adam training algorithm [22].

3.2. Hyper-parameters in structured output layer

Hyper-parameters including ψ and α can significantly affect the performance of SOL models, a series of experiments thus are conducted to figure out the optimal selection of activation function ψ and the value of weight α. Mel-cepstral distortion (Mel-CD), the Euclidean distance between the MCEPs of converted speech and that of target speech, is measured for objective evaluation. Table 1 presents the objective evaluation of spectral features prediction using different activation functions with prefixed α value. Tanh activation function outperforms others in the evaluation thus being selected as default ψ. Fig. 6 illustrates the objective evaluation results on spectral features prediction using different value of α. Optimal performance has been achieved using α=0.925 which is used as the default weight value.

Table 1: *Mel-CD (dB) of spectrum prediction with different activation functions with α = 0.925 in DBLSTM-SOL based system (S1) and DBFTLSM-SOL based system (S2).*

			ψ –activation		
System	Linear	Softmax	Sigmoid	ReLU	Tanh
S1	5.489	5.482	5.463	5.467	**5.425**
S2	5.296	5.302	5.265	5.284	**5.259**

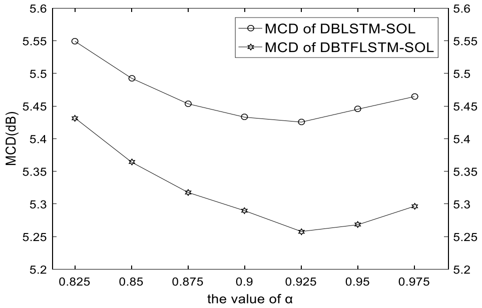

Fig. 6: *Mel-CD of spectral features predicted from DBLSTM-SOL and DBTFLSTM-SOL systems with different α values.*

3.3. Objective evaluation

In the objective evaluation, Mel-CD is measured to assess the conversion performance of systems employing different models. As illustrated in Table 2, TFLSTM based system outperforms LSTM baseline with a 3.5% relative improvement and achieves similar performance with conventional DBLSTM based system. When extending to DBTFLSTM, a 4.3% relative improvement is further gained. By employing SOL, the system gains further 0.9% relative improvement over DBTFLSTM based systems, which is the best among all systems.

Table 2: *Objective evaluation results of spectral features generated by aforementioned systems.*

Systems	Network architecture	Numbers of parameters	Mel-CD (dB)
LSTM	1*1024	4.3M	5.6858
TFLSTM	1*(9*230)	3.8M	5.5402
DBLSTM	2*(2*336)	3.7M	5.4852
DBTFLSTM	2*(2*9*100)	3.7M	5.3046
DBLSTM-SOL	2*(2*336)	3.7M	5.4254
DBTFLSTM-SOL	2*(2*9*100)	3.7M	**5.2588**

3.4. Subjective evaluation

Mean opinion score (MOS) is used to evaluate the perceived naturalness and quality of converted speeches. 15 utterances from the test set are randomly selected as the testing material. For each of them, 7 parallel utterances, containing 6 converted utterances from aforementioned approaches and the original target natural speech, are randomly shuffled and then evaluated by 15 listeners with no reported listening difficulties. These speeches are scored following a 5-point scale in naturalness and speech quality, in which the grades are standardized as 5 = Excellent (same as the natural speech), 4 = Good, 3 = Fair, 2 = Poor, 1 = Bad. As illustrated in Fig. 7, TFLSTM based system outperforms LSTM baseline, gaining relative improvements for naturalness and quality at 2.4% and 3.3% respectively. After extending to DBTFLSTM, the scores increase to 3.45 and 3.48, 2.7% and 2.6% better comparing to DBLSTM based system. Employing SOL in DBTFLSTM based system can further improve the naturalness and quality at 3.2% and 2.9% respectively, achieving 3.8% and 3.7% relative improvements over DBLSTM-SOL based system for naturalness and quality.

ABX preference test is used to measure the speaker similarity of converted speeches. Participants are asked to choose one of two converted utterances A or B of higher similarity to the target utterance X. If it is hard to tell, no preference (N/P) is allowed. All pairs are randomly shuffled to avoid preference bias. As illustrated in Fig. 8, TFLSTM based systems are frequently preferred over paired LSTM based systems. However, no significant improvement has been observed by extending TFLSTM to DBTFLSTM. DBTFLSTM-SOL based system has achieved better preferences over DBTFLSTM based system. The results illustrate the effectiveness of the proposed methods.

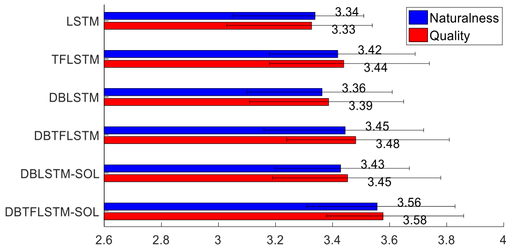

Fig. 7: *MOS test results for speech naturalness and quality with 95% confidence intervals, the natural speech has set as 'Excellent'.*

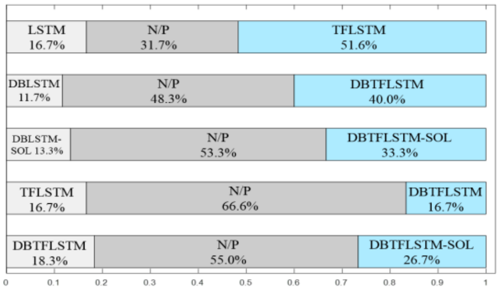

Fig. 8: *ABX preference test results for speaker similarity, where N/P stands for no preference.*

4. Conclusions

This paper proposes the use of TFLSTM and SOL in voice conversion. With TFLSTM, the proposed system can model both time-wise and frequency-wise patterns simultaneously through the input sequences. With SOL, the simple but related pitch parameter prediction task can be used as an auxiliary task to support the complex spectral feature prediction thus to explicitly exploit the correlation between pitch and spectrum. Experimental results suggest the improvement on speech quality and speaker similarity by using the proposed techniques. In the future, we will explore the possibility of combining proposed framework and text-to-speech (TTS) synthesis to generate personalized speeches with different expressive characteristics such as emphasis, interactive styles, etc.

5. Acknowledgement

This work is supported by National High Technology Research and Development Program of China (2015AA016305), National Natural Science Foundation of China (NSFC) / Research Grants Council of Hong Kong (RGC) joint research scheme (61531166002, N CUHK404/15) as well as NSFC (61375027, 61433018) and National Social Science Foundation of China (13&ZD189).

6. References

[1] Y. Stylianou, O. Cappe, and E. Moulines, “Continuous probabilistic transform for voice conversion,” IEEE Transactions on Speech and Audio Processing, vol. 6, no. 2, pp. 131–142, 1998.

[2] T. Toda, A. W. Black, and K. 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.

[3] Z. Wu, T. Virtanen, T. Kinnunen, E. S. Chng, and H. Li, “Exemplar-based voice conversion using non-negative spectrogram deconvolution,” [in] Proc. 8th ISCA Speech Synthesis Workshop, 2013.

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

[5] H. Zen, A. Senior and M. Senior, “Statistical Parametric Speech Synthesis Using Deep Neural Networks,” [in] Proc. ICASSP, pp. 8012-8016, 2013.

[6] S. Desai, E. V. Raghavendra, B. Yegnanarayana, A. W. Black, and K. Prahallad, “Voice conversion using artificial neural networks,” [in] Proc. ICASSP, 2009.

[7] T. Nakashika, R. Takashima, T. Takiguchi, and Y. Ariki, “Voice conversion in high-order eigen space using deep belief nets,” [in] Proc. INTERSPEECH, 2013.

[8] L. Sun, S. Kang, K. Li, and H. Meng, “Voice conversion using deep bidirectional long short-term memory based recurrent neural networks,” [in] Proc. ICASSP, 2015.

[9] A. Mohamed, G. Hinton, and G. Penn, “Understanding how deep belief networks perform acoustic modelling,” [in] Proc. ICASSP, pp. 4273–4276, 2012.

[10] K.N. Stevens, Acoustic Phonetics, MIT Press, 2000, ISBN 0-262-69250-3, 978-0-262-69250-2.

[11] J. Li, A. Mohamed, G. Zweig, and Y. Gong, “Exploring multidimensional LSTMs for large vocabulary ASR,” [in] Proc. ICASSP, 2015.

[12] P. Swietojanski, P. Bell, and S. Renals. "Structured output layer with auxiliary targets for context-dependent acoustic modelling." [in] Proc. INTERSPEECH, pp. 1964-1967, 2015.

[13] R. Caruana, Multitask learning, Springer, 1998

[14] M. L. Seltzer, and J. Droppo, “Multi-task learning in deep neural networks for improved phoneme recognition,” [in] Proc. ICASSP, 2013.

[15] A. Graves, S. Fernández, J. Schmidhuber, “Multi-dimensional recurrent neural networks,” [in] Proc. ICANN, pp. 549-558, 2007.

[16] A. Graves and J. Schmidhuber, “Offline handwriting recognition with multidimensional recurrent neural networks,” Advances in Neural Information Processing Systems, pp. 545-552, 2009.

[17] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” IEEE Transactions on Signal Processing, vol. 45, pp.2673–2681, 1997.

[18] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigne, "Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency based F0 extraction: Possible role of a repetitive structure in sounds,” Speech communication, vol. 27, no. 3, pp. 187–207, 1999.

[19] P. J. Werbos, “Backpropagation through time: what it does and how to do it,” [in] Proc. IEEE, vol. 78, no. 10, pp. 1550–1560, 1990.

[20] F. Chollet, Keras [OL]. [2016-11-19]. GitHub repository. https://github.com/fchollet/keras.

[21] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, “Theano: A CPU and GPU math compiler in Python,” [in] Proc. SciPy, pp. 1-7, 2010.

[22] D. P. Kingma, and J. L. Ba, “Adam: A method for stochastic optimization,” [in] Proc. ICLR, 2015.