Voice Conversion Using Deep Bidirectional Long Short-Term Memory based Recurrent Neural Networks

Conference Paper · April 2015

DOI: 10.1109/ICASSP.2015.7178896

CITATIONS
13

READS
270

4 authors:

Lifa Sun
The Chinese University of Hong Kong
4 PUBLICATIONS **18** CITATIONS
SEE PROFILE

Shiyin Kang
The Chinese University of Hong Kong
5 PUBLICATIONS **65** CITATIONS
SEE PROFILE

Kun Li
The Chinese University of Hong Kong
11 PUBLICATIONS **53** CITATIONS
SEE PROFILE

Helen M. Meng
The Chinese University of Hong Kong
233 PUBLICATIONS **2,229** CITATIONS
SEE PROFILE

VOICE CONVERSION USING DEEP BIDIRECTIONAL LONG SHORT-TERM MEMORY BASED RECURRENT NEURAL NETWORKS

Lifa Sun, Shiyin Kang, Kun Li and Helen Meng

Human-Computer Communications Laboratory
Department of System Engineering and Engineering Management
The Chinese University of Hong Kong, Hong Kong SAR, China
{lfsun, sykang, kli, hmmeng}@se.cuhk.edu.hk

ABSTRACT

This paper investigates the use of Deep Bidirectional Long Short-Term Memory based Recurrent Neural Networks (DBLSTM-RNNs) for voice conversion. Temporal correlations across speech frames are not directly modeled in frame-based methods using conventional Deep Neural Networks (DNNs), which results in a limited quality of the converted speech. To improve the naturalness and continuity of the speech output in voice conversion, we propose a sequence-based conversion method using DBLSTM-RNNs to model not only the frame-wised relationship between the source and the target voice, but also the long-range context-dependencies in the acoustic trajectory. Experiments show that DBLSTM-RNNs outperform DNNs where Mean Opinion Scores are 3.2 and 2.3 respectively. Also, DBLSTM-RNNs without dynamic features have better performance than DNNs with dynamic features.

Index Terms— voice conversion, bidirectional long short-term memory, recurrent neural networks, dynamic features

1. INTRODUCTION

Voice Conversion (VC) is a technique that aims to modify the speech of a source speaker to make it sound like that of a target speaker. The most obvious application for VC is Text-to-Speech (TTS) synthesis, for creating new and personalized voices. Other potential applications include security-related usage (e.g. hiding the identity of the speaker), vocal restoration in case of pathology, speech-to-speech translation, movie dubbing, as well as games and other entertainment applications [1].

Many techniques have been developed for VC. We can divide these techniques into two categories: rule-based approaches and statistical approaches. The rule-based voice conversion is achieved by modifying the acoustic information of the speech signal according to specific rules [2, 3]. Although this method can keep most of detailed information, it is not stable since different speakers need different rules. On the other hand, statistical approaches to VC estimate a mapping function between the spectral features of the source and target speech. Popular techniques include Gaussian Mixture Models (GMMs) and Deep Neural Networks (DNNs) [4, 5]. Stylianou et al. [6] proposed a GMM-based mapping method to convert the source speaker spectral envelops by a continuous parametric function. Toda et al. [7] improved GMM-based method through using dynamic features and global variance. Desai et al. [8] proposed a voice conversion method using Artificial Neural Networks. Chen et al. [9, 10] proposed a spectral modelling and conversion method using RBM. Nakashika et al. [11] used DNNs to achieve VC in a high order eigenspace. Nakashika et al. [12] also proposed a sequence modelling method using Recurrent Temporal Restricted Boltzmann Machines, which is a kind of Recurrent Neural Networks (RNNs).

The existing approaches have two main problems. First, GMMs and DNNs frame-based methods treat speech frames as independent input features and do not capture the temporal dependencies of speech sequences. Second, standard RNNs are able to capture the temporal information among speech frames, but they have limited capabilities in modelling context. Furthermore, standard RNNs can only make use of the previous context and not the future context. They also have limited storage to deal with long sequence because of the problem of vanishing and exploding gradients [13], hence they have difficulty in learning long-range context-dependencies.

To overcome these two problems, an alternative RNNs architecture, Bidirectional Long Short-Term Memory (BLSTM) [14, 15], is proposed for voice conversion in this paper. In previous work, BLSTM outperforms standard RNNs on numerous tasks involving sequence modelling, such as context-free and context-sensitive languages learning [16], large-vocabulary speech recognition [17, 18], feature enhancement [19], TTS synthesis [20], etc. Bidirectional recurrent connections can make full use of the context information in both forward and backward directions elegantly. The LSTM network architecture including memory blocks and peephole connections makes it possible to store information in linear memory cells over a longer period of time and to learn the optimal amount of contextual information for the task.

The organization of this paper is as follows: the RNNs and BLSTM architectures are described in Section 2. The baseline DNN-based system and the voice conversion system using BLSTM are respectively provided in Section 3 and Section 4. To evaluate the performance of our approach, objective and subjective experiments were conducted, and the results and analysis are presented in Section 5. Finally, Section 6 gives a summary and conclusion of this work.

2. NETWORK ARCHITECTURES

For a standard RNNs, given an input sequence $\boldsymbol{x} = (x_1, \cdots, x_T)$, the hidden vector $\boldsymbol{h} = (h_1, \cdots, h_T)$ and the output vector $\boldsymbol{y} = (y_1, \cdots, y_T)$ can be computed from $t = 1$ to T according to the following iterative equations:

$$h_t = \mathcal{H}(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad (1)$$

$$y_t = W_{hy}h_t + b_y \quad (2)$$

where $\mathcal{H}$ is the activation function of hidden layer, W is the weight matrix (e.g., W_{xh} is the input-hidden weight matrix), and b is the

bias vectors (e.g., b_h is the hidden bias vectors).

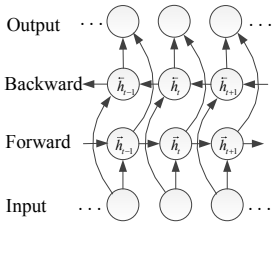

Fig. 1. Bidirectional RNNs

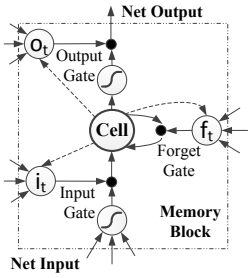

Fig. 2. A Memory Block

To make full use of the context of speech sequences in both preceding and succeeding directions, Bidirectional RNNs (BRNNs) were proposed [21]. As shown in Fig. 1, BRNNs compute the forward sequence $\vec{h}$ and the backward sequence $\overleftarrow{h}$ by iterating the forward layer from $t = T$ to 1 and the backward layer from $t = 1$ to T. The iterating functions are as follows:

$$\vec{h}_t = \mathcal{H}(W_{x\vec{h}}x_t + W_{\vec{h}\vec{h}}\vec{h}_{t-1} + b_{\vec{h}}) \quad (3)$$

$$\overleftarrow{h}_t = \mathcal{H}(W_{x\overleftarrow{h}}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t+1} + b_{\overleftarrow{h}}) \quad (4)$$

$$y_t = W_{\vec{h}y}\vec{h}_t + W_{\overleftarrow{h}y}\overleftarrow{h}_t + b_y \quad (5)$$

Furthermore, in the standard RNNs, $\mathcal{H}$ is usually a sigmoid or hyperbolic tangent function, which leads to the limitation of the inability to learn long-range context-dependencies. However, it is reported that the an LSTM network that contains memory blocks can solve this problem. An LSTM network consists of recurrently connected blocks, known as memory blocks. The structure of a single LSTM memory block is illustrated in Fig. 2. Every memory block contains self-connected memory cells and three adaptive and multiplicative gate units i.e. input, output, and forget gates which can respectively provide write, read, reset operations for the cells. Among them, forget gates are shown to be essential for problems involving continual or very long input strings [22]. $\mathcal{H}$ is implemented according to the following equations:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (6)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{ci}c_{t-1} + b_f) \quad (7)$$

$$c_t = f_t c_{t-1} + i_t tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (8)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (9)$$

$$h_t = o_t tanh(c_t) \quad (10)$$

where i, f, o, c refer to the input gate, forget gate, output gate and the element of cells C respectively. σ is the logistic sigmoid function.

Combining the advantages of BRNNs and LSTM, Bidirectional LSTM based RNNs were designed [14], which can make the best of long-range context in both forward and backward directions. Further, motivated the success of deep network architectures, Deep BLSTM-RNNs are considered to build up high level representation of input features. Similar to the structure of DNNs and Deep RNNs [23], Deep BLSTM can be created by stacking multiple BLSTM hidden layers. In this paper, we implement DBLSTM-RNNs system to achieve voice conversion.

3. BASELINE: CONVENTIONAL DNN BASED APPROACH

A system using conventional DNNs is developed as the baseline approach [11]. The DNN based architecture consists of two Deep Belief Networks and a concatenating NN, as shown in Fig. 3. In this approach, the speech parameters are extracted by the STRAIGHT analysis [24], including Mel-cepstral coefficients (MCEPs), fundamental frequency (F_0) and an aperiodic component. MCEPs are derived from spectral envelop. The aperiodic component is defined as the ratio between the lower and upper smoothed spectral envelopes in the frequency domain. The DNN model is trained by the back-propagation algorithm using parallel MCEPs features of the source and target speech. In the conversion stage, MCEPs features of the source are converted by the trained model frame by frame. Specifically, the DNN based approach is a frame-based method and does not consider the context-dependencies of acoustic sequence.

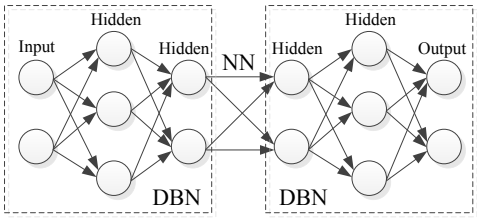

Fig. 3. The conventional DNN based approach.

A DNN based approach with dynamic features is also developed for comparison. Static features and dynamic features are used in the observation vector in this approach. Dynamic features consists of first-order and second-order time derivatives of speech parameters, hence this approach has the ability of modelling several successive frames and can smooth the converted spectral trajectory.

4. PROPOSED: DBLSTM-RNN BASED APPROACH

4.1. Basic Framework

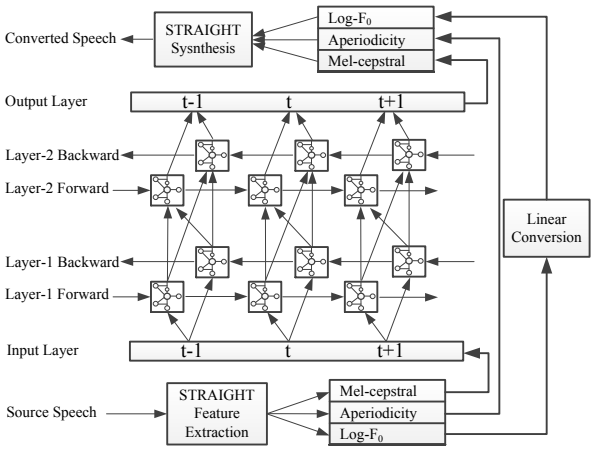

Fig. 4. The DBLSTM-RNN based voice conversion architecture.

We propose a new voice conversion approach using DBLSTM-RNNs. Fig. 4 shows the overall framework of the proposed system. In this system, the three feature streams including the MCEPs, log F_0 and the aperiodic component, are converted separately. MCEPs (except for the energy feature) are converted by the DBLSTM-RNN model. Log F_0 is converted by equalizing the mean and the standard deviation of the source and target speech, which is a widely used method in VC area. The aperiodic component is directly copied to synthesize the converted speech, since previous research shows that converting aperiodic component does not make statistically significant difference on the synthesized speech [25]. The system regards the whole utterance as input, which makes it possible to access the long-range context in both forward and backward directions.

4.2. Training Stage

Back-propagation (BP) is now the most widely used training tool in the field of artificial neural networks. But conventional back-propagation [26] is only suitable for the feed-forward networks. For RNNs, Rumelhart and Werbos et al. [27, 28] extended the conventional BP algorithm to back-propagation through time (BPTT), which can be used for sequential models.

In short, BPTT begins by unfolding an RNN into a standard feed-forward network through time steps. As shown in Fig. 5, an RNN containing one recurrent layer $\mathbf{f}$ and one feed-forward layer $\mathbf{g}$ can be unfolded as k instances of $\mathbf{f}$ and one instance of $\mathbf{g}$. In the above example, the network has been unfolded to a feed-forward network with the depth of k = 3. In real training, the k can be set to a fixed number (typically 20) or the length of the entire sequence. After unfolding, the training proceeds in a manner similar to training a feed-forward neural network with back-propagation algorithm, except that each epoch must run through the observations in sequential order. That is, for RNNs training, the weight gradients are computed for one sentence at a time.

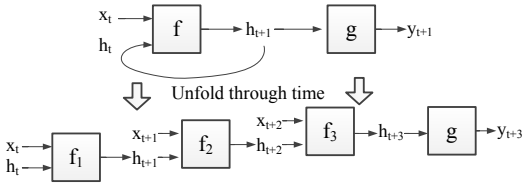

Fig. 5. Unfolding RNNs through time.

4.3. Conversion Stage

In the conversion stage, the input of the proposed system is one whole sentence of the source speech. Parameters, including F_0, aperiodicity and MCEPs, are extracted by STRAIGHT method. MCEPs are normalized before conversion, and then all the three feature streams are converted by previously described methods in section 4.1 respectively. Next, the STRAIGHT vocoder is used to synthesize the speech waveform from the converted F_0, aperiodicity, and the de-normalized MCEPs.

For the approaches using dynamic features, one step is added before waveform synthesis. Maximum Likelihood Parameters Generation (MLPG) [29] is conducted to generate smooth speech parameter sequence. The de-normalized converted MCEPs is used as the mean of the MLPG input probability density function (pdf), while the global variance of the whole training data is used as the variance of the MLPG input pdf.

5. EXPERIMENTS

5.1. Experimental Setup

In our voice conversion experiments, the data we use is the CMU ARCTIC corpus [30]. We select a male speaker (AWB) as the source, and a female speaker (SLT) as the target. The acoustic signals are sampled at 16kHz with mono channel, windowed by 25-ms. The frame shift is 5ms. To get the parallel utterances, the dynamic time warping (DTW) algorithm is used to align the features sequences of the source and target speaker. 49-dimensional Mel-cepstral features are used in DNN-based and DBLSTM-based approaches, while a total number of 147 dimensions covering static features, delta and delta-delta features are used in experiments with dynamic features. In the experiments, the number of training data is 349,852 frames (593 sentences, about 42 mins), while the number of validation data is 69,173 frames (119 sentences, about 9 mins). Four systems are implemented for comparison:

- **DNN**: DNN based system [11], the baseline approach.
- **DNN-DYN**: DNN based system with dynamic features.
- **DBLSTM**: Proposed system using DBLSTM-RNNs.
- **DBLSTM-DYN**: Voice conversion system using DBLSTM-RNNs approach with dynamic features.

In the DBLSTM approach, the number of units in each layer is [49 128 256 256 128 49] respectively, where each bidirectional LSTM hidden layer contains one forward LSTM layer and one backward LSTM layer. The training samples are normalized to zero mean and unit variance for each dimension before training. We train the networks using the BPTT with a learning rate of 1.0×10^{-5} and a momentum of 0.9. We use a C++ CUDA-enabled machine learning library named RECURRENT [31] to train the DBLSTM model. The training procedure is carried on one Tesla K20 GPU and it takes about 48 hours. The BLSTM-DYN approach differ from DBLSTM approach is that the number of units in input layer and output layer is 147, and MLPG is conducted before synthesis.

For the DNN approach, the number of units in each layer is the same as that of DNN-based approach. The networks are pre-trained using stochastic gradient descent with a mini-batch size of 128 training samples. In the training stage, 800 epoches are executed using the BP algorithm with a learning rate of 0.1. Under the same hardware configuration with the DBLSTM approach, it takes about 4 hours.

5.2. Objective Evaluation

One common objective evaluation method in VC [7, 8, 12] and speech synthesis [32, 33, 34] area is to compute the spectral distortion between the generated speech and the target speech. In VC area, Mel-cepstral Distortion (MCD) is the Euclidean distance between the MCEPs of converted speech and that of target speech. To evaluate the performance of the proposed system objectively, We use the MCD to measure how close the converted speech is to the target speech. MCD is defined as follows:

$$MCD[dB] = \frac{10}{ln10}\sqrt{2\sum_{d=1}^{N}(c_d - c_d^{converted})^2} \qquad (11)$$

where c_d and $c_d^{converted}$ denote the d-th coefficient of the target and converted Mel-cepstrum respectively. N is the dimension of Mel-cepstrum (except the energy feature).

The MCD evaluation is concluded on systems trained on training sets of different sizes. As shown in Fig. 6, our proposed approach outperforms the DNN method (baseline method) both with and without dynamic features. We can also see that dynamic features reduce the mel-cepstrum distortion of the DNN approach, but they do not have obvious effect on the DBLSTM method. A possible reason may be that the DBLSTM model has made full use of the long-range context information of acoustic sequences, and does not need the dynamic features.

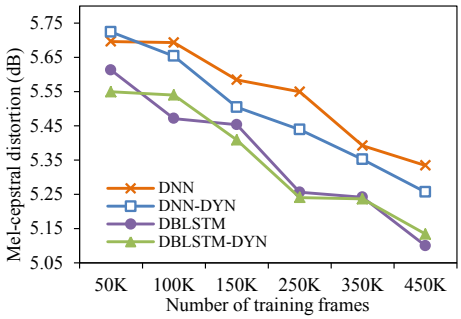

Fig. 6. Mel-cepstral distortion for each method. The smaller MCD value is, the closer the converted spectra is to the target spectra.

5.3. Subjective Evaluation

We conduct the Mean Opinion Score (MOS) test on the naturalness of the converted speech and the ABX preference test on the similarity. We use 16 utterances as the test set. The 16 utterances are converted by the four systems (DNN, DNN-DYN, DBLSTM and DBLSTM-DYN) respectively. 25 listeners are asked to rate 16 utterances from each system, hence generating 400 ratings per system.

In the MOS test, listeners are asked to compare the four utterances of each set[1] with the target speech, and select how natural the converted speech sounded using a 5-point scale (5: excellent, 4: good, 3: fair, 2: poor, 1:bad).

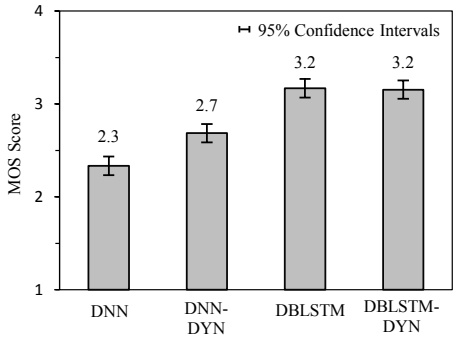

Fig. 7. MOS test results with the 95% Confidence Intervals.

[1]The converted speech samples can be found at `http://www.se.cuhk.edu.hk/~lfsun/icassp2015`

The MOS results in Fig. 7 show that listeners consider that the naturalness of outputs from DBLSTM and DBLSTM-DYN systems to be better than the DNN and DNN-DYN systems. The speech converted by the DBLSTM and DBLSTM-DYN systems sounds more clear with less background noise. The results also suggest that dynamic features can improve the continuity of converted speech in DNN approach, but do not have obvious enhancements for DBLSTM approach. The speech converted by the DNN system has glitches between the phonemes, while for the other three systems the converted speech sounds coherent and smooth.

In the ABX preference test, listeners are asked to choose which sample (A or B converted by two different systems) sounds more similar to X, which is the original target speaker's utterance. The samples A and B are shuffled to avoid preferential bias. We conduct four sets of comparative experiments: DNN with DBLSTM, DNN with DNN-DYN, DNN-DYN with DBLSTM, and DBLSTM with DBLSTM-DYN. For all the experiments, listens have three choices: A, B or no preference (N/P) when they cannot distinguish between the two. We use p-values to determine the significance of the results. The smaller the p-value, the larger the significance.

From the first two bars in Fig. 8, we see that the DBLSTM achieves significantly better preferences over the DNN and DNN-DYN approach. The third bar in Fig. 8 shows that the DNN-DYN approach is preferred over the DNN approach. The fourth bar suggests that the DBLSTM and DBLSTM-DYN methods have the similar levels of performance.

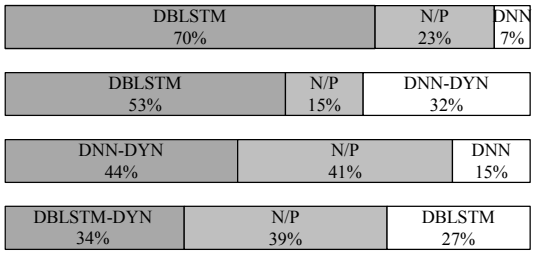

Fig. 8. ABX preference test results. The p-values of the four pairs are 3.9×10^{-24}, 3.0×10^{-3}, 2.4×10^{-4} and 0.24 respectively.

6. CONCLUSIONS

We have proposed a new voice conversion approach using DBLSTM-RNNs, which can model both the frame-wised relationship between the source and the target speech and the long-range context-dependencies of acoustic sequences. From both objective and subjective evaluation metrics, experimental results show that, our proposed method of DBLSTM-RNNs can improve the naturalness and continuity of the converted speech significantly, increasing the MOSs from 2.3 to 3.2. Our future work includes exploring the advantages of BLSTM in modelling spectral and F_0 features simultaneously.

7. ACKNOWLEDGMENT

This project is partially sponsored by a grant from the Hong Kong SAR Government General Research Fund (Project Number: 14205814)

8. REFERENCES

[1] J. Nurminen, H. Silen, and V. Popa, "Voice Conversion," *Speech Enhancement, Modeling and Recognition-Algorithms and Applications*, pp. 69–94, 2012.

[2] Z. W. Shuang, R. Bakis, S. Shechtman, and Y. Qin, "Frequency warping based on mapping formant parameters," in *Interspeech*, 2006.

[3] D. Erro and A. Moreno, "Weighted frequency warping for voice conversion," in *Interspeech*, 2007.

[4] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for Deep Belief Nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[5] A. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using Deep Belief Networks," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 14–22, 2012.

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

[7] 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.

[8] S. Desai, E. V. Raghavendra, B. Yegnanarayana, A. W. Black, and K. Prahallad, "Voice conversion using Artificial Neural Networks," in *ICASSP*, 2009.

[9] L. H. Chen, Z. H. Ling, Y. Song, and L. R. Dai, "Joint spectral distribution modeling using Restricted Boltzmann Machines for voice conversion," in *Interspeech*, 2013.

[10] L. H. Chen, Z. H. Ling, L. J. Liu, and L. R. Dai, "Voice conversion using Deep Neural Networks with layer-wise generative training," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 22, no. 12, pp. 1859–1872, 2014.

[11] T. Nakashika, R. Takashima, T. Takiguchi, and Y. Ariki, "Voice conversion in high-order eigen space using Deep Belief Nets," in *Interspeech*, 2013.

[12] T. Nakashika, T. Takiguchi, and Y. Ariki, "High-order sequence modeling using speaker-dependent recurrent temporal Restricted Boltzmann Machines for voice conversion," in *Interspeech*, 2014.

[13] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[14] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, vol. 18, no. 5, pp. 602–610, 2005.

[15] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] F. A. Gers and J. Schmidhuber, "LSTM recurrent networks learn simple context-free and context-sensitive languages," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1333–1340, 2001.

[17] A. Graves, N. Jaitly, and A. R. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *ASRU*, 2013.

[18] H. Sak, A. Senior, and F. Beaufays, "Long Short-Term Memory based Recurrent Neural Network architectures for large vocabulary speech recognition," *arXiv preprint arXiv:1402.1128*, 2014.

[19] M. Wollmer, Z. X. Zhang, F. Weninger, B. Schuller, and G. Rigoll, "Feature enhancement by bidirectional LSTM networks for conversational speech recognition in highly non-stationary noise," in *ICASSP*, 2013.

[20] Y. C. Fan, Y. Qian, F. L. Xie, and F. K. Soong, "TTS synthesis with bidirectional LSTM based Recurrent Neural Networks," in *Interspeech*, 2014.

[21] M. Schuster and K. K. Paliwal, "Bidirectional Recurrent Neural Networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[22] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[23] A. Graves, A. R. Mohamed, and G. E. Hinton, "Speech recognition with deep Recurrent Neural Networks," in *ICASSP*, 2013, pp. 6645–6649.

[24] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné, "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.

[25] Y. Ohtani, T. Toda, H. Saruwatari, and K. Shikano, "Maximum likelihood voice conversion based on GMM with STRAIGHT mixed excitation," in *Proc. ICSLP*, 2006.

[26] G. E. Hinton, "Learning distributed representations of concepts," in *Proceedings of the eighth annual conference of the cognitive science society*, 1986.

[27] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Cognitive modeling*, 1988.

[28] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[29] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for hmm-based speech synthesis," in *ICASSP*, 2000.

[30] J. Kominek and A. W. Black, "The CMU Arctic speech databases," in *Fifth ISCA Workshop on Speech Synthesis*, 2004.

[31] F. Weninger and J. Bergmann, "Currennt: CUDA-enabled machine learning library for Recurrent Neural Networks," `http://sourceforge.net/projects/currennt/`.

[32] S.Y. Kang, X.J. Qian, and H. Meng, "Multi-distribution Deep Belief Network for speech synthesis," in *ICASSP*, 2013.

[33] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using Deep Neural Networks," in *ICASSP*, 2013.

[34] S.Y. Kang and H. Meng, "Statistical parametric speech synthesis using weighted multi-distribution Deep Belief Network," in *Interspeech*, 2014.