UNIDIRECTIONAL LONG SHORT-TERM MEMORY RECURRENT NEURAL NETWORK WITH RECURRENT OUTPUT LAYER FOR LOW-LATENCY SPEECH SYNTHESIS

Heiga Zen, Haşim Sak

Google

{heigazen,hasim}@google.com

ABSTRACT

Long short-term memory recurrent neural networks (LSTM-RNNs) have been applied to various speech applications including acoustic modeling for statistical parametric speech synthesis. One of the concerns for applying them to text-to-speech applications is its effect on latency. To address this concern, this paper proposes a low-latency, streaming speech synthesis architecture using unidirectional LSTM-RNNs with a recurrent output layer. The use of unidirectional RNN architecture allows frame-synchronous streaming inference of output acoustic features given input linguistic features. The recurrent output layer further encourages smooth transition between acoustic features at consecutive frames. Experimental results in subjective listening tests show that the proposed architecture can synthesize natural sounding speech without requiring utterance-level batch processing.

Index Terms— Statistical parametric speech synthesis; recurrent neural networks; long short-term memory; low-latency;

1. INTRODUCTION

Statistical parametric speech synthesis (SPSS) [1] offers various advantages over concatenative speech synthesis [2]. However, the naturalness of the synthesized speech from SPSS is still not as good as that of the best samples from concatenative speech synthesizers. One of the major factors that degrades the naturalness is the accuracy of acoustic modeling [1].

Introduction of the deep neural network (DNN) [3], which is a feed-forward artificial neural network with many hidden layers, has opened a new research direction for acoustic modeling in SPSS [4–7]. A number of linguistic features that affect speech, including phonetic, syllabic, and grammatical ones, have to be taken into account in acoustic modeling for SPSS to produce natural sounding synthesized speech. A typical system uses around 50 different types of linguistic features [8]. Effective modeling of these complex context dependencies is one of the most critical problems [9]. In the DNN-based SPSS, a DNN is trained to learn a mapping function from linguistic features (inputs) to acoustic features (outputs) [4]. DNN-based acoustic models offer an efficient and distributed representation of complex dependencies between linguistic and acoustic features [10] and have shown the potential to produce natural sounding synthesized speech [4, 7]. The DNN-based SPSS was further extended to predict full conditional distribution of acoustic features rather than predicting only conditional mean values using mixture density output layer [11].

One limitation of the feed-forward DNN-based acoustic modeling is that the sequential nature of speech is ignored. Although certainly there are correlations between consecutive frames in speech data, the DNN-based approach assumes that each frame is sampled independently. Although this problem can be relaxed by smoothing predicted acoustic features using the speech parameter generation algorithm [12, 13], which utilizes dynamic features as constraints to generate smooth trajectories, it is desirable to incorporate the sequential nature of speech data to the acoustic model itself. Recurrent neural networks (RNNs) [14] provides an elegant way to model speech-like sequential data that embodies correlations between neighbouring frames. It can use all the available input features to predict output features at each frame [15]. Tuerk and Ronbinson [16] and Karaani *et al.* [17] applied standard RNNs to speech synthesis, whereas long short-term memory RNNs (LSTM-RNNs) [18], which can capture long-term dependencies, were recently applied to acoustic modeling for SPSS [19–21]; Fan *et al.* and Fernandez *et al.* applied deep bidirectional LSTM-RNNs, which can access input features at both past and future frames, to acoustic modeling for SPSS and reported improved naturalness [19, 22]. Fan *et al.* also claimed that deep bidirectional LSTM-RNNs can generate smooth speech parameter trajectories thus no smoothing step was required, whereas Zen *et al.* reported that having the smoothing step was still helpful with unidirectional LSTM-RNNs [20, 21].

Many text-to-speech (TTS) applications require low-latency speech synthesis. Since it is not straightforward to perform low-latency speech synthesis in SPSS due to the utterance-level batch processing nature of the speech parameter generation algorithm [12], three approaches have been proposed;

1. Use the time-recursive version of the speech parameter generation algorithm [12, 23].
2. Use an acoustic model allowing streaming inference such as autoregressive hidden Markov models (AR-HMMs) [24].
3. Split a sentence into sub-sentence-level blocks then perform synthesis at each block [25].

To achieve low-latency speech synthesis with LSTM-RNNs, this paper proposes a streaming synthesis architecture using unidirectional LSTM-RNNs with a recurrent output layer. The proposed approach can be put in the second category; the unidirectional architecture allows streaming inference, and the recurrent output layer further encourages smooth transition between consecutive acoustic frames. Experimental results in subjective listening tests show that the proposed architecture can synthesize natural sounding speech without requiring utterance-level batch processing.

The rest of this paper is organized as follows. Section 2 describes the proposed streaming synthesis architecture. Experimental results in subjective evaluations are presented in Section 3. Concluding remarks are shown in the final section.

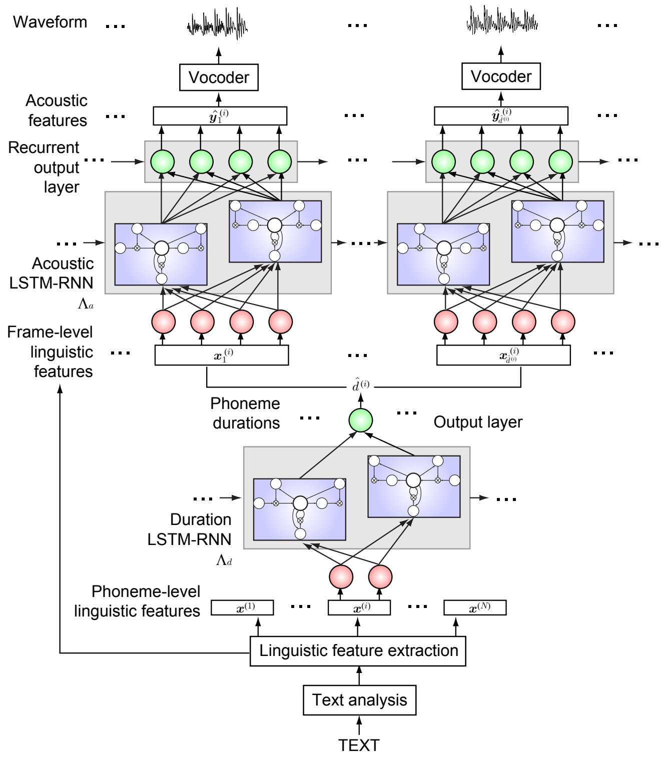
Waveform
Vocoder
Vocoder
Acoustic features
Recurrent output layer
Acoustic LSTM-RNN Λ_a
Frame-level linguistic features
Phoneme durations
Output layer
Duration LSTM-RNN Λ_d
Phoneme-level linguistic features
Linguistic feature extraction
Text analysis
TEXT

Fig. 1. Overview of the proposed streaming synthesis architecture using unidirectional LSTM-RNNs with a recurrent output layer.

2. STREAMING SYNTHESIS USING UNIDIRECTIONAL LSTM-RNNS WITH RECURRENT OUTPUT LAYER

Figure 1 illustrates the proposed speech synthesis architecture using unidirectional LSTM-RNNs with a recurrent output layer. Here, feature extraction, duration and acoustic feature prediction, and vocoding are executed in a streaming manner. The synthesis process can be outlined as follows:

```
1:  Perform text analysis over input text
2:  for $i = 1, \dots, N$ do                          ▷ Loop over phonemes
3:      Extract $\boldsymbol{x}^{(i)}$
4:      Predict $\hat{d}^{(i)}$ given $\boldsymbol{x}^{(i)}$ by $\Lambda_d$
5:      for $\tau = 1, \dots, \hat{d}_i$ do                     ▷ Loop over frames
6:          Compose $\boldsymbol{x}_\tau^{(i)}$ from $\boldsymbol{x}^{(i)}$, $\tau$, and $\hat{d}^{(i)}$
7:          Predict $\hat{\boldsymbol{y}}_\tau^{(i)}$ given $\boldsymbol{x}_\tau^{(i)}$ by $\Lambda_a$
8:          Synthesize waveform given $\hat{\boldsymbol{y}}_\tau^{(i)}$ then stream result
9:      end for
10: end for
```

where N is the total number of phonemes in the input sentence, and Λ_d and Λ_a are duration and acoustic LSTM-RNNs, respectively. $\boldsymbol{x}^{(i)}$ and $\hat{d}^{(i)}$ correspond to the phoneme-level linguistic feature vector and the predicted phoneme duration at the i-th phoneme. $\boldsymbol{x}_\tau^{(i)}$ and $\hat{\boldsymbol{y}}_\tau^{(i)}$ are frame-level linguistic feature vector and the predicted acoustic feature vector at the τ-th frame in the i-th phoneme, respectively. Note that the text analysis step is batch processing, whereas the remaining steps are streamed, as the first step is usually significantly faster than the remaining ones. The details of the LSTM-RNN and recurrent output layer are described in the next section.

2.1. LSTM-RNN

The LSTM-RNN architecture is designed to model temporal sequences and their long-term dependencies [18]. It has special units called *memory blocks*. The memory blocks contain memory cells with self-connections storing the temporal state of the network in addition to special multiplicative units called gates to control the flow of information. It has been successfully applied to various applications, such as speech recognition [26, 27], handwriting recognition [28], and speech synthesis [19–22].

Typically, feedback loops at hidden layers of an RNN are unidirectional; the input is processed from left to right, i.e. the flow of the information is only forward direction. To use both past and future inputs for prediction, Schuster proposed the bidirectional RNN architecture [15]. It has forward and backward feedback loops that flow the information in both directions. This architecture enables the network to predict outputs using inputs of entire sequence. The bidirectional version of LSTM-RNNs have been proposed [28] and applied to acoustic modeling for TTS [19, 22].

However, as inference using bidirectional LSTM-RNNs involves the propagation of inputs over time in both forward and backward directions, bidirectional LSTM-RNNs inherently have large latency; to predict the first frame of a sequence, inputs for the last frame need to be propagated through the network over time. This prohibits using bidirectional LSTM-RNNs in commercial TTS services; if a user enter a very long text as input for TTS, its latency can be prohibitively large.

Unidirectional LSTM-RNNs do not have this issue as the forward propagation can be done in a frame-synchronous, streaming manner. They can still access future inputs by windowing, looking-ahead, or delaying outputs with reasonable increase in the number of parameters. This paper investigates unidirectional LSTM-RNNs as the acoustic model for TTS.

2.2. Recurrent Output Layer

A single hidden-layer, forward-directional RNN[1] computes hidden activations $\{\boldsymbol{h}_t\}_{t=1}^T$ and output features $\{\boldsymbol{y}_t\}_{t=1}^T$ given input features $\{\boldsymbol{x}_t\}_{t=1}^T$ by iterating the following recursion.

$$\boldsymbol{h}_t = f\left(\boldsymbol{W}_{hx}\boldsymbol{x}_t + \boldsymbol{W}_{hh}\boldsymbol{h}_{t-1} + \boldsymbol{b}_h\right), \quad (1)$$

$$\boldsymbol{y}_t = \phi\left(\boldsymbol{W}_{yh}\boldsymbol{h}_t + \boldsymbol{b}_y\right), \quad (2)$$

where $\boldsymbol{h}_0 = \boldsymbol{0}$, $\boldsymbol{W}_{hx}$, $\boldsymbol{W}_{yh}$, and $\boldsymbol{W}_{hh}$ correspond to the weight matrices for input/hidden connection, hidden/output connection, and feedback loop at the hidden layer. $\boldsymbol{b}_h$ and $\boldsymbol{b}_y$ are the bias vectors for the hidden and output layers, respectively, and $f(\cdot)$ and $\phi(\cdot)$ are the activation functions for the hidden and output layers, respectively. The feed-back mechanism in Eq. (1) – *i.e.* activations at the previous time step being fed back into the network along with the inputs, allows the network to propagate information across frames (time) and learn sequences.

The recurrent output layer is a simple extension of the conventional RNN; use recurrent connection at the output layer as well. Equation (2) is extended to have recurrent term as

$$\boldsymbol{y}_t = \phi\left(\boldsymbol{W}_{yh}\boldsymbol{h}_t + \boldsymbol{W}_{yy}\boldsymbol{y}_{t-1} + \boldsymbol{b}_y\right), \quad (3)$$

where $\boldsymbol{W}_{yy}$ is the weight matrix for the recurrent connection at the output layer. The recurrent connection at the output layer can be

[1]For notation simplicity the activation function definitions for simple RNN are given here to describe recurrent output layer. In the actual implementation, h_t is computed with an LSTM layer.

viewed as a trainable time-invariant smoother for output features. It encourages smooth transitions between consecutive frames.

3. EXPERIMENTS

3.1. Experimental Conditions

Speech data in US English from a female professional speaker was used for the experiments. The training and development data sets consisted of 34 632 and 100 utterances, respectively. A set of speaker-dependent duration and acoustic feed-forward DNNs and unidirectional LSTM-RNNs were trained from the data.

From the speech data and its associated transcriptions, phonetic alignments were automatically generated using an HMM-based aligner, which was trained in a bootstrap manner. Phoneme-level linguistic features for the DNNs and the LSTM-RNNs included 445 and 291 linguistic contexts[2] (*e.g.* phoneme identities, stress marks, the number of syllables in a word, position of the current syllable in a phrase), respectively. Then phoneme-level linguistic features, 3 numerical features for coarse-coded position of the current frame in the current phoneme, and 1 numerical feature for duration of the current segment were used to form frame-level linguistic features.

The speech analysis conditions were similar to those used for the Nitech-HTS 2005 [29] system. The speech data was downsampled from 48 kHz to 16 kHz, then 40 mel-cepstral coefficients [30], logarithmic fundamental frequency ($\log F_0$) values, and 5-band aperiodicities (0–1, 1–2, 2–4, 4–6, 6–8 kHz) [29] were extracted every 5 ms. The output features of the duration DNNs and LSTM-RNNs were phoneme-level durations. The output features of the acoustic DNNs and LSTM-RNNs were acoustic features consisting of 40 mel-cepstral coefficients, $\log F_0$ value, and band 5 aperiodicity. To model $\log F_0$ sequences, the continuous F_0 with explicit voicing modeling approach [31] was used; voiced/unvoiced binary value was added to the output features and $\log F_0$ values in unvoiced frames were interpolated. To evaluate the effect of the speech parameter generation algorithm-based smoothing, DNNs and LSTM-RNNs were trained with and without dynamic features in their acoustic features.

Both the input and output features were normalized in advance; the input features were normalized to have zero-mean unit-variance, whereas the output features were normalized to be within 0.01–0.99 based on their minimum and maximum values in the training data. The architecture of the DNNs was 4 hidden-layer, 1024 units per layer, with the rectified linear activation function (ReLU) [32] at their hidden layers. The architecture of the LSTM-RNNs was 1 forward-directed hidden LSTM layer with 256 memory blocks. A linear activation function was used in the output layer for the DNNs and LSTM-RNNs, *i.e.*, $\phi(\boldsymbol{x}) = \boldsymbol{x}$. Both feed-forward (Eq. (2)) and recurrent (Eq. (3)) architectures were investigated for the output layer of the acoustic LSTM-RNNs. The feed-forward architecture was used for the output layers of the duration LSTM-RNNs as output feature-level continuity is not required for durations.

To reduce the training time and impact of having many silence frames, 80% of silence frames were removed from the training data.[3] Durations of the beginning and ending silences were excluded from the training data for the duration DNNs and LSTM-RNNs. The weights of the LSTM-RNNs were initialized randomly (no pretraining was performed), whereas those of the DNNs were initialized using the layer-wise back-propagation (BP) pre-training [33]. Then they were updated to minimize the mean squared error between the target and predicted output features. A GPU implementation of a mini-batch stochastic gradient descent (SGD)-based BP algorithm with AdaDec-based learning rate scheduling [34] and momentum term [35] was used. For training the LSTM-RNNs, a distributed CPU implementation of mini-batch ASGD-based truncated back propagation through time (BPTT) [36] algorithm was used [27]. Training was continued until the mean squared error over the development set converged. Training the DNNs and LSTM-RNNs took approximately a half day and a day, respectively.

At the synthesis stage, durations and acoustic features were predicted from linguistic features using the trained networks. If the acoustic features included dynamic features, entire utterance-level, batch-processing version of the speech parameter generation algorithm (case 1 in [12]) was used to generate smooth acoustic feature trajectories.[4] Here, the per-dimension variances computed from all training data were used with the speech parameter generation algorithm. Otherwise, the predicted acoustic features were used directly in the later vocoding step. Spectral enhancement based on post-filtering in the cepstral domain [39] was applied to improve the naturalness of the synthesized speech. From the acoustic features, speech waveforms were synthesized using the Vocaine vocoder [40].

To subjectively evaluate the performance of the systems, preference and mean opinion score (MOS) tests were also conducted. 100 utterances not included in the training data were used for evaluation. One subject could evaluate a maximum of 30 pairs in the preference test and 30 stimuli in the MOS test. Each pair was evaluated by five subjects in the preference test, whereas each stimulus was evaluated by seven subjects in the MOS test. The subjects used headphones. In the preference tests, after listening to each pair of samples, the subjects were asked to choose their preferred one, whereas they could choose "neutral" if they did not have any preference. In the MOS tests, after listening to a stimulus, the subjects were asked to rate the naturalness of the stimulus in a 5-scale score (1: Bad, 2: Poor, 3: Fair, 4: Good, 5: Excellent).

3.2. Experimental Results

Table 1 shows the preference test results. The following trends and analysis can be seen and derived from the results:

1. Smoothing over frames was essential for DNNs (row 1).
   $\rightarrow$ Discontinuity due to frame-wise mapping.
2. LSTM-RNNs produced significantly better speech than DNNs (rows 2–3).
   $\rightarrow$ Recurrent architecture helps.
3. Smoothing over frames was still helpful for LSTM-RNNs though it was less significant compared with the DNN (rows 1, 4 & 7).
   $\rightarrow$ Recurrent architecture at hidden layer is not enough.
4. Two smoothing approaches (dynamic features or recurrent output layer) gave similar naturalness (row 6).
   $\rightarrow$ Smoothing by dynamic features, which requires an utterance-level batch processing, can be replaced by the recurrent output layer, which allows streaming processing.

[2]The input features for the DNNs included past and future 2 contexts at phoneme, syllable, word, and phrase levels. Those for the LSTM-RNNs included only future 2 contexts at these levels, as the unidirectional LSTM-RNNs can access the past contexts through their recurrent connections.

[3]A preliminary experiment showed that removing silence frames had no negative impact in training LSTM-RNNs.

[4]The generation algorithm considering global variance [37] was not investigated in this experiment as it has larger latency than the standard speech parameter generation algorithm. Although there is a low-latency version of this algorithm [38], it is not directly applicable to the proposed framework.

Table 1. Subjective preference scores (%) between speech samples from the feed-forward DNNs and the unidirectional LSTM-RNNs with and without dynamic features and recurrent output layer. "Feed-forward" and "Recurrent" correspond to the use of feed-forward and recurrent output layers. "w/" and "w/o" indicate the usage of dynamic features. The systems which achieved significantly better preference at $p < 0.01$ level are in the bold font.

	DNN		LSTM-RNN						
	Feed-forward		Feed-forward		Recurrent				
row	w/	w/o	w/	w/o	w/	w/o	Neutral	p-value	Z-score
1	**67.8**	12.0	–	–	–	–	20.0	$< 10^{-9}$	19.5
2	18.4	–	**34.9**	–	–	–	47.6	$< 10^{-6}$	-5.0
3	26.4	–	–	**35.2**	–	–	38.4	0.01	-2.5
4	–	–	**21.0**	12.2	–	–	66.8	$< 10^{-2}$	3.5
5	–	–	24.2	–	18.2	–	57.6	0.04	2.1
6	–	–	21.8	–	–	21.0	57.2	0.78	0.3
7	–	–	–	11.4	–	**32.0**	56.6	$< 10^{-9}$	-7.9
8	–	–	–	–	16.6	**29.2**	54.2	$< 10^{-4}$	-4.3

5. Cascading the two smoothing approaches degraded the naturalness (rows 5 & 8).
   $\rightarrow$ Possibly due to oversmoothing.

Although Fan *et al.* claimed that dynamic features and smoothing step were not required with deep bidirectional LSTM-RNNs [19], they did not perform comparison between them with and without dynamic features. The experimental results here indicate that smoothing was still necessary for unidirectional LSTM-RNNs, though it could be done in a streaming manner using the recurrent output layer rather than dynamic feature-based approach.

Table 2. Subjective MOSs of speech samples from the feed-forward DNNs with dynamic feature-based smoothing and the unidirectional LSTM-RNNs with a recurrent output layer.

Model	# of params	5-scale MOS
DNN	3 749 79	3.370 ± 0.114
LSTM-RNN	476 435	3.723 ± 0.105

Table 2 shows the MOS test results. The proposed unidirectional LSTM-RNNs with a recurrent output layer achieved 3.723 in 5-scale MOS. It can be also be seen from the table that LSTM-RNNs offer more efficient acoustic modeling than feed-forward DNNs.

4. CONCLUSIONS

This paper has proposed a streaming speech synthesis architecture using unidirectional LSTM-RNNs with a recurrent output layer. The use of unidirectional rather than bidirectional ones allows a fully streaming architecture and low-latency speech synthesis. The recurrent output layer encourages smooth transitions between consecutive frames. Experimental results in subjective preference and MOS listening tests confirmed that the proposed architecture could synthesize natural sounding speech and allowed us to remove the speech parameter generation step from the synthesis pipeline.

Future work includes evaluations of the proposed architecture on mobile devices, comparing the unidirectional and bidirectional LSTM-RNNs for TTS, and combining LSTM-RNNs with mixture density output layer. Evaluating bidirectional LSTM-RNNs with dynamic features and recurrent output layer is also necessary.

5. ACKNOWLEDGEMENTS

The authors would like to thank Drs. Andrew Senior, Alex Graves, Keiichi Tokuda, and Yannis Agiomyrgiannakis for helpful comments and discussions.

6. REFERENCES

[1] H. Zen, K. Tokuda, and A. Black, "Statistical parametric speech synthesis," *Speech Commn.*, vol. 51, no. 11, pp. 1039–1064, 2009.

[2] A. Hunt and A. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proc. ICASSP*, 1996, pp. 373–376.

[3] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Process. Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[4] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. ICASSP*, 2013, pp. 7962–7966.

[5] R. Fernandez, A. Rendel, B. Ramabhadran, and R. Hoory, "f_0 contour prediction with a deep belief network-Gaussian process hybrid model," in *Proc. ICASSP*, 2013, pp. 6885–6889.

[6] H. Lu, S. King, and O. Watts, "Combining a vector space representation of linguistic context with a deep neural network for text-to-speech synthesis," in *Proc. ISCA SSW8*, 2013, pp. 281–285.

[7] Y. Qian, Y. Fan, W. Hu, and F. Soong, "On the training aspects of deep neural network (DNN) for parametric TTS synthesis," in *Proc. ICASSP*, 2014, pp. 3857–3861.

[8] K. Tokuda, H. Zen, and A. Black, "An HMM-based speech synthesis system applied to English," in *Proc. IEEE Speech Synthesis Workshop*, 2002, CD-ROM Proceeding.

[9] K. Yu, H. Zen, F. Mairesse, and S. Young, "Context adaptive training with factorized decision trees for HMM-based statistical parametric speech synthesis," *Speech Commn.*, vol. 53, no. 6, pp. 914–923, 2011.

[10] H. Zen, "Deep learning in speech synthesis," in *Keynote speech given at ISCA SSW8*, 2013, `http://research.google.com/pubs/archive/41539.pdf`.

[11] H. Zen and A. Senior, "Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis," in *Proc. ICASSP*, 2014, pp. 3872–3876.

[12] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. ICASSP*, 2000, pp. 1315–1318.

[13] T. Toda, A. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Trans. Audio Speech Lang. Process.*, vol. 15, no. 8, pp. 2222–2235, 2007.

[14] A. Robinson and F. Fallside, "Static and dynamic error propagation networks with application to speech coding," in *Proc. NIPS*, 1988, pp. 632–641.

[15] M. Schuster, *On supervised learning from sequential data with applications for speech recognition*, Ph.D. thesis, Nara Institute of Science and Technology, 1999.

[16] C. Tuerk and T. Robinson, "Speech synthesis using artificial neural networks trained on cepstral coefficients," in *Proc. Eurospeech*, 1993, pp. 1713–1716.

[17] O. Karaali, G. Corrigan, I. Gerson, and N. Massey, "Text-to-speech conversion with neural networks: A recurrent TDNN approach," in *Proc. Eurospeech*, 1997, pp. 561–564.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[19] Y. Fan, Y. Qian, and F. Soong, "TTS synthesis with bidirectional LSTM based recurrent neural networks," in *Proc. Interspeech*, 2014, pp. 1964–1968.

[20] H. Zen, "Statistical parametric speech synthesis," in *Tutorial given at UKSpeech Conference*, 2014, `http://research.google.com/pubs/archive/42624.pdf`.

[21] H. Zen, H. Sak, A. Graves, and A. Senior, "Statistical parametric speech synthesis based on recurrent neural networks," in *Poster presentation given at UKSpeech Conference*, 2014.

[22] R. Fernandez, A. Rendel, B. Ramabhadran, and R. Hoory, "Prosody contour prediction with long short-term memory, bi-directional, deep recurrent neural networks," in *Proc. Interspeech*, 2014.

[23] T. Muramatsu, Y. Ohtani, T. Toda, H. Saruwatari, and K. Shikano, "Low-delay voice conversion based on maximum likelihood estimation of spectral parameter trajectory," in *Proc. Interspeech*, 2008, pp. 1076–1079.

[24] M. Shannon, H. Zen, and W. Byrne, "Autoregressive models for statistical parametric speech synthesis," *IEEE Trans. Acoust. Speech Lang. Process.*, vol. 21, no. 3, pp. 587–597, 2013.

[25] X. Na, X. Xie, and J. Kuang, "Low latency parameter generation for real-time speech synthesis system," in *Proc. ICME*. IEEE, 2014, pp. 1–6.

[26] A. Graves, M. Abdel-rahman, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. ICASSP*, 2013, pp. 6885–6889.

[27] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. Interspeech*, 2014.

[28] M. Liwicki, A. Graves, H. Bunke, and J. Schmidhuber, "A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks," in *Proc. ICDAR*, 2007, pp. 367–371.

[29] H. Zen, T. Toda, M. Nakamura, and T. Tokuda, "Details of the Nitech HMM-based speech synthesis system for the Blizzard Challenge 2005," *IEICE Trans. Inf. Syst.*, vol. E90-D, no. 1, pp. 325–333, 2007.

[30] T. Fukada, K. Tokuda, T. Kobayashi, and S. Imai, "An adaptive algorithm for mel-cepstral analysis of speech," in *Proc. ICASSP*, 1992, pp. 137–140.

[31] K. Yu and S. Young, "Continuous F0 modelling for HMM based statistical parametric speech synthesis," *IEEE Trans. Audio Speech Lang. Process.*, vol. 19, no. 5, pp. 1071–1079, 2011.

[32] M. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q.-V. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, and G. Hinton, "On rectified linear units for speech processing," in *Proc. ICASSP*, 2013, pp. 3517–3521.

[33] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. ASRU*. IEEE, 2011, pp. 24–29.

[34] A. Senior, G. Heigold, M. Ranzato, and K. Yang, "An empirical study of learning rates in deep neural networks for speech recognition," in *Proc. ICASSP*, 2013, pp. 6724–6728.

[35] D.E. Rumelhart and J.L. McCelland, *Parallel distributed processing*, MIT Press, 1986.

[36] R. Williams and J. Peng, "An efficient gradient-based algorithm for on-line training of recurrent network trajectories," *Neural Comput.*, vol. 2, no. 4, pp. 490–501, 1990.

[37] T. Toda and K. Tokuda, "A speech parameter generation algorithm considering global variance for HMM-based speech synthesis," *IEICE Trans. Inf. Syst.*, vol. E90-D, no. 5, pp. 816–824, 2007.

[38] M. Shannon and W. Byrne, "Fast, low-artifact speech synthesis considering global variance," in *Proc. ICASSP*. IEEE, 2013, pp. 7869–7873.

[39] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Incorporation of mixed excitation model and postfilter into HMM-based text-to-speech synthesis," *IEICE Trans. Inf. Syst.*, vol. J87-D-II, no. 8, pp. 1563–1571, 2004.

[40] Y. Agiomyrgiannakis, "Vocaine the vocoder and applications is speech synthesis," in *Submitted to ICASSP*, 2015.