IIITH Submission for Blizzard Challenge 2017: A BLSTM based SPSS System using MatNN

Sivanand Achanta, Anandaswarup Vadapalli, and Suryakanth V Gangashetty

Speech Processing Lab, KCIS
International Institute of Information Technology Hyderabad, India
{sivanand.a, anandaswarup.vadapalli}@research.iiit.ac.in, svg@iiit.ac.in

Abstract

In this paper, text-to-speech (TTS) system submitted by IIITH team for Blizzard Challenge 2017 is described. This year's Blizzard Challenge is a continuation to previous year's task of synthesizing children's audio books. Our TTS system is based on statistical parametric speech synthesis (SPSS) paradigm. It is quite challenging to statistically model large variations in prosody that are unique to the expressive audio book data given in the challenge. We have explored two neural architectures for acoustic modeling in SPSS to model them, they are: (1) bidirectional long short-term memory (BLSTM) recurrent neural networks and (2) a deep hybrid architecture consisting of two feedforward layers, followed by four highway layers with a BLSTM stacked on top. Based on the objective scores on a held out test set, we have submitted the former system for the challenge. Details of various architectural choices and training are presented. From the results, it is clear that our system convincingly outperforms the baseline deep neural network and hidden Markov model based SPSS systems in most evaluations with statistical significant value (p) set to 0.01.

Index Terms: BLSTM, Blizzard Challenge, SPSS

1. Introduction

Statistical parametric speech synthesis (SPSS), has become the dominant approach for text-to-speech synthesis (TTS) over the last decade [1]. The key factors for the success of this paradigm are: (1) Compactness: Since the statistical models learn to predict speech, we no longer need to store the original waveforms as in unit selection and (2) Flexibility: The parameters of statistical models can be suitably transformed to obtain desired variations in the synthesized speech [2]. Several statistical models have been investigated for SPSS in the last decade and can be broadly classified into two approaches: (1) Generative Models: These explicitly model speech dynamics and use a different model for predicting speech parameters from text. Examples of this approach are hidden Markov model - Gaussian mixture model (HMM-GMM) [3], linear dynamical models [4], and restricted Boltzman machine [5] based SPSS systems, and (2) Conditional Models: The text-to-speech regression is modeled directly rather than using two separate models as in the former approach. Classification and regression trees [6], random forests [7], Gaussian processes [8], and neural network (NN) [9] based SPSS belong to this category.

Of these, deep neural network (DNN) based approaches have emerged as state-of-the-art in SPSS [10]. The improvements in naturalness over HMM based synthesis has been attributed to (1) powerful regression capabilites of DNNs and (2) the frame-level modeling [11, 12]. In addition to improving acoustic modeling, DNNs have demosntrated flexibility which is one of the key factors for the success of SPSS paradigm. Modeling variations like multiple speakers, languages, ages using a single neural network was shown in [13, 14, 15].

However, the naturalness of the synthetic speech using SPSS is still low and there is a scope for improvement. In a DNN based SPSS approach, the improvements can be made to following components of text-to-speech pipeline (1) acoustic modeling (2) text features (3) speech parametrization (vocoding), (4) post-filtering and (5) duration modeling. Advancements in acoustic modeling were made using more appropriate architectures for time-series like recurrent neural networks (RNN) [16, 17, 18] and modeling higher-order statistics using mixture density networks [19]. Speech parametrization using neural network approaches like [20, 21] seem to be performing better than the conventional signal processing vocoders like STRAIGHT. Similarly in post-filtering, generatively trained NNs [22, 23] have resulted in improved speech quality over traditional signal processing based post-filtering methods. Duration modeling could be explicit or implicit. Previously, most NN based SPSS systems relied on an external duration predictor [24]. Of late, there are some studies where duration modeling is carried out simultaneously with acoustic modeling [25, 26, 27].

Recently, another class of models called Conditional-Generative models have emerged as state-of-the-art in TTS systems [26, 28]. These combine above two approaches of SPSS into a single approach and are relatively newer. Also, these are statistical speech synthesis systems and *not* SPSS systems, as they do not use parametric representations of speech to synthesize waveforms but directly predict the samples.

Our TTS system is based on statistical parametric speech synthesis (SPSS) paradigm. It is quite challenging to statistically model large variations in prosody that are unique to the expressive audio book data given in the challenge. We have explored two neural architectures for acoustic modeling in SPSS to model them, they are: (1) bidirectional long short-term memory (BLSTM) recurrent neural networks and (2) a deep hybrid architecture consisting of two feedforward layers, followed by four highway layers with a BLSTM stacked on top. Further description of model training, hyper-parameter fine-tuning and architecture selection are given in the below sections.

The paper is organised as follows: Section 2 gives the details about the toolkit used to build NN models. In Section 3, details of the dataset are given. In Sections 4, 5 and 6, acoustic modeling, experimental setup and objective evaluation of models are discussed respectively. Results are presented in Section 7. Concluding remarks and future work are presented in Section 8.

2. MatNN Toolbox

This section describes the toolbox used to build our SPSS system. The MatNN toolbox[1] is developed in MATLAB for training advanced neural networks for various tasks, but mainly tested on SPSS. The toolbox now contains DNN, DNN with attention (DNN-WA) [29, 30], dropout [31], highway network [32], various RNNs including ERNN [33], GRU [34], SLSTM [35], LSTM [36], BLSTM [37] and the recently introduced hybrid architecture convolution bank highway GRU (shortly CBHG) [38]. It also contains sequence-to-sequence with attention architectures like seq2seq [39], Tacotron [27] and Transformer [40]. All the RNNs are trained using full back propagation through time (BPTT) and currently there is no support for truncated BPTT [41].

While the DNN can run on both GPU and CPU, the recurrent architectures run on CPU alone. Also, the recurrent architectures use a single sequence at a time and hence run using pure SGD than the more popular minibatch SGD. But pure SGD gives us the advantage that the local minima reached is more generalizable than that of those obtained using minibatch. This feature may be very useful when using small datasets like the Blizzard Challenge 2017 dataset.

This toolkit is not written to create an alternative to the existing high-performance (interms of speed in training) toolkits like Tensorflow or Theano or Torch. The main purpose is to be able to make the core neural net code of complex architectures as simple as possible. The core neural network code (i.e., the forward and back propagation of any architecture) is made very transparent without getting mired under several layers of generic classes. The code is easily modifiable to create a new architectures. For all the architectures gradient check is carried out by comparing with numerical gradients to make sure back propagation is correct [42].

3. Dataset

In this section, we describe the data used for building our models. Seven new audio books were added to the previous year's Blizzard Challenge data. This new data was manually segmented at the sentence level. The special indicator sounds of the start and end of the chapter were manually removed. Since the some of the text was in PDF format, it was passed through an online OCR tool to convert into a suitable format for building prompts in Festival followed by alignment using EHMM tool [43].

We used all the data without any trimming for building models. Three stories namely (1) The Boy Who Cried the Wolf, (2) The Enormous Turn Tip, and (3) Goldilocks and the Three Bears, were used as validation set and the rest of the data for training. This split was adapted from CSTR Edinburgh's last year Blizzard Challenge paper [44]. The best performing model on the validation data interms of objective metrics was chosen to synthesize the test utterances. Table 1 shows the number of utterances used for training and validation and corresponding amount of data in hours.

4. Acoustic Modeling

4.1. BLSTM

At the core of acoustic model is BLSTM which is briefly described here. While bidirectional version is used in the experiments, for simplicity, below we give the unidirectional LSTM equations.

[1] https://github.com/SivanandAchanta/nntoolbox

Table 1: *Description of speech corpus used*

	Total	Train	Val
# Utts.	7252	6999	253
# Hrs	5.4	5.2	0.2

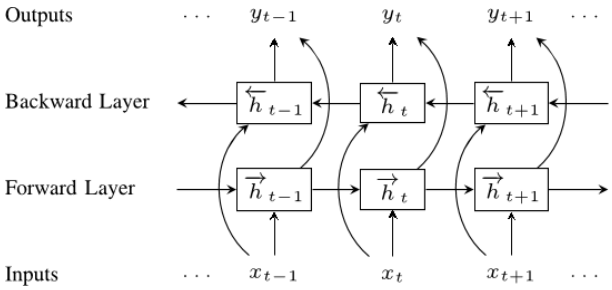

Figure 1: *Bidirectional Long-Short Term Memory*

The implementation of LSTM is rather standard including forget gate and peephole connections (shown in Fig. 1). The forward pass equations of our LSTM implementation based on [45] are given below:

$$\begin{aligned}
\mathbf{z_t} &= \mathrm{g}(\mathbf{W_z x_t} + \mathbf{R_z h_{t-1}} + \mathbf{b_z}) \\
\mathbf{i_t} &= \sigma(\mathbf{W_i x_t} + \mathbf{R_i h_{t-1}} + \mathbf{p_i} \odot \mathbf{c_{t-1}} + \mathbf{b_i}) \\
\mathbf{f_t} &= \sigma(\mathbf{W_f x_t} + \mathbf{R_f h_{t-1}} + \mathbf{p_f} \odot \mathbf{c_{t-1}} + \mathbf{b_f}) \\
\mathbf{c_t} &= \mathbf{i_t} \odot \mathbf{z_t} + \mathbf{f_t} \odot \mathbf{c_{t-1}} \\
\mathbf{o_t} &= \sigma(\mathbf{W_o x_t} + \mathbf{R_o h_{t-1}} + \mathbf{p_o} \odot \mathbf{c_t} + \mathbf{b_o}) \\
\mathbf{h_t} &= \mathbf{o_t} \odot \mathrm{h}(\mathbf{c_t})
\end{aligned} \tag{1}$$

where g and h are *tanh* activation functions and σ denotes *sigmoid* activation function. $\mathbf{W_z}$, $\mathbf{W_i}$, $\mathbf{W_f}$, $\mathbf{W_o}$ are the weights from input and $\mathbf{R_z}$, $\mathbf{R_i}$, $\mathbf{R_f}$, $\mathbf{R_o}$ are the weights from previous state at unit-input, input gate, forget gate, and output gate, respectively. $\mathbf{p_i}$, $\mathbf{p_f}$ and $\mathbf{p_o}$ are the peep-hole connections and $\odot$ indicates element-wise multiplication. $\mathbf{x_t}$, $\mathbf{h_t}$ are the input and hidden state at time t.

The bidirectional version has another LSTM operating in the reverse direction. The output $\mathbf{y_t}$ is predicted using the forward hidden state $\overrightarrow{\mathbf{h_t}}$ and the backward hidden state $\overleftarrow{\mathbf{h_t}}$ as

$$\mathbf{y_t} = \mathbf{U_f}\overrightarrow{\mathbf{h_t}} + \mathbf{U_b}\overleftarrow{\mathbf{h_t}} + \mathbf{b_u} \tag{2}$$

Where $\mathbf{U_f}, \mathbf{U_b}, \mathbf{b_u}$ represent output layer forward, backward weights and the bias.

5. Experimental Setup

In this section, details of the experimental conditions are given. A neural network with one bidirectional long short-term memory layer is used [36, 37]. The forward and backward hidden layers had 500 units each with $tanh$ non-linearity. The weights of the network are randomly initialized from a Gaussian distribution with variance scaled to 0.01 and biases are initialized to zero excepting the forget-gate bias which is initialized to 1 [46]. The model was trained using pure stochastic gradient descent with ADAM optimizer [47]. We use full back propagation through time [41]. The learning rate was set to $\alpha = 0.0003$ and

decay rate was set to $\beta_1 = 0.9$. The input and output feature for all the models are mean and variance normalized. The source code for replicating our experiments is available online[2].

5.1. Phoneset and Alignment

We have used two different phonesets for aligning the data using EHMM namely the *mrpa* and *unilex* phonesets. Both of them are UK based English pronunciation phonesets. We found that using *unilex* phoneset improved the log-likelihood of the data implying that the alignment of the data was better using the latter phoneset. We chose to build the voice with the *unilex* phoneset.

5.2. Input Features

Input features are composed of: (1) categorical features (pentaphone identities, vowel identity in current syllable, etc), (2) numerical features (# of syllables in word, # words in phrase and so on), (3) durational features. The dimension of the input feature vector is 334.

5.3. Output Features

50 dimensional Mel cepstral features (MCEP) and 26 dimensional band-aperiodicities (BAP) were extracted with a frame-shift of 5 ms for all the speech utterances along with their deltas and double-deltas. This feature extraction is followed from HTS-STRAIGHT demo available online. F_0, Δ F_0, $\Delta\Delta$ F_0 and a binary voiced/unvoiced flag are used for modeling pitch and voicing features. During testing, voiced/unvoiced threshold was set using the best threshold computed from the training data. The dimension of the output feature was 235.

5.4. Duration Modeling

The durations were modeled using another BLSTM with 50 $tanh$ units. The input was same as the one described above except the duration features and the duration of the current phone in seconds was the output. Both the input and output are standardized.

5.5. Synthesis

During synthesis time, durations are predicted using a separate BLSTM. The predicted output features are un-normalized using training data mean and variance before passing to the parameter generation module [3]. The resultant features are post-filtered using the global variance technique proposed in [48] and then synthesized using STRAIGHT vocoder [49].

6. Objective Evaluation

In this section, we discuss how various hyper-parameter settings have been arrived at. Below we report objective metrics for each hyper-parameter setting on the validation set and the best architecture has been chosen for submission.

6.1. Acoustic Modeling

Table 2 shows objective scores for two neural architectures namely (1) BLSTM and (2) MLP-HW-BLSTM. The BLSTM architecture is 334L500N235L and MLP-HW-BLSTM architecture is 334L500R250R250R250R250R250R250N235L where L - linear units and N - $tanh$ units. Clearly the BLSTM architecture performs slightly better than the hybrid architecture and hence has been chosen for final submission.

Table 2: *Objective scores of various systems*

System	MCD (dB)	F_0 RMSE (Hz)	VUVE (%)	BAPD (dB)
BLSTM	**5.33**	68.98	**11.54**	**26.50**
MLP-HW-BLSTM	5.35	**68.80**	11.56	26.63

7. Results

In this section, we discuss the results of our system and contrast it with the performance of baselines. Our system identifier is M.

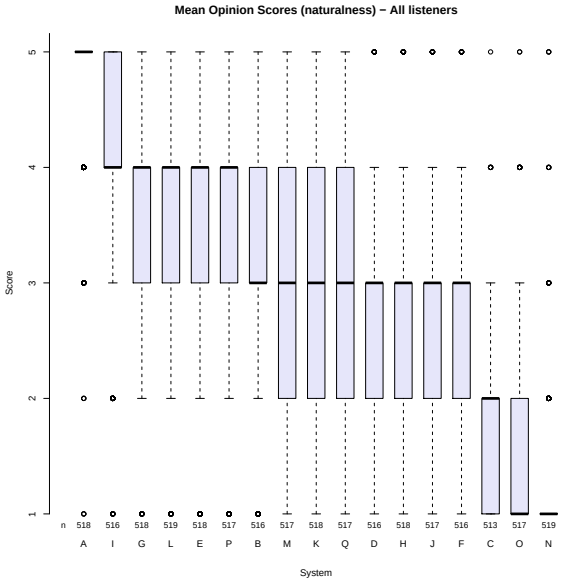

Figure 2: *Our system (M) performance in the naturalness of synthesized speech with all listeners*

7.1. Naturalness

We first consider the results for naturalness. Mean opinion scores for naturalness from all listeners on book sentences are shown in Figure 2. Our system outperformed the two SPSS based baseline systems (C and D) and is very close the strong unit selection based Festival baseline system (B). The pairwise Wilcoxon signed rank tests (with alpha Bonferoni corrected) with p = 0.01, reveals that there *is* significant difference between baseline systems C, D and our system (M), while there is *no* significant difference between baseline system B and our system (M). The same trend can be seen across the scores made by paid listeners, speech experts and on-line volunteers.

7.2. Speaker Similarity

We now consider mean opinion scores for speaker similarity. The mean opinion scores for speaker similarity from all listeners on book sentences are shown in Figure 3. Considering ratings from all listeners (or any other listener group), our system seems to be performing well. Again, the pairwise Wilcoxon

[2] `https://www.goo.gl/87tQGp`

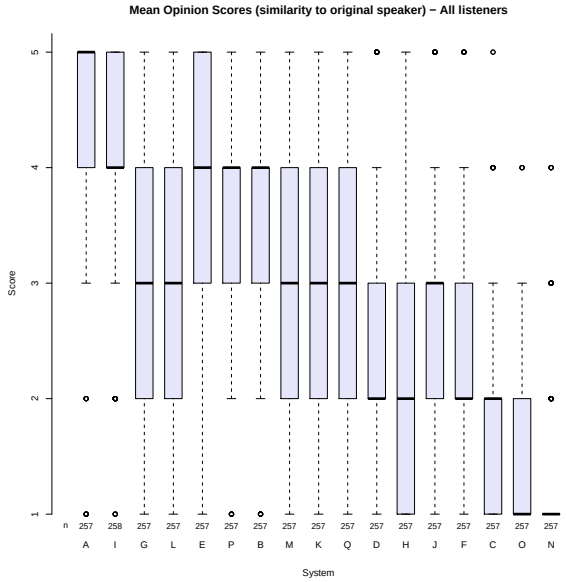

Figure 3: *Our system (M) performance in the similarity to the original speaker with all listeners*

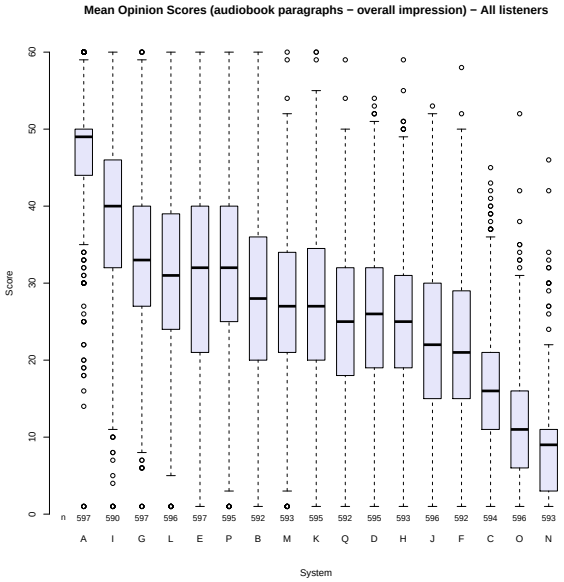

Figure 4: *Our system (M) performance in the overall performance of synthesized speech at paragraph level with all listeners*

signed rank tests (with alpha Bonferoni corrected) with p = 0.01, reveals that there *is* significant difference between baseline systems C, D and our system (M). However, in speaker similarity unlike naturalness there *is* significant difference between baseline system B and our system (M) implying that B being a unit-seletion based system has convincingly higher speaker similarity than our SPSS based system.

7.3. Evaluation of Audiobook Paragraphs

We now consider the results for evaluation of audiobook paragraphs that have been evaluated on several other factors like stress, intonation, emotion, pleasantness, listening effort, speech pauses and overall impression. Considering ratings from all listeners on overall impression are shown in Figure 4, our system showed similar performance as in the case of the isolated sentence evaluation of naturalness and speaker similarity.

7.4. Intelligibility

Our systems performance is most notable in word error rate (WER) on semantically unpredictable sentences (SUS). Clearly Fig. 5 shows that our systems WER performance is amongst top performing systems and is around 32 %. The pairwise Wilcoxon signed rank tests (with alpha Bonferoni corrected) with p = 0.01, reveals that there is no significant difference between system D,I,J,L,G and our system (M).

8. Conclusions and Future Work

The main challenge of synthesizing expressive audio book data involves appropriately modeling large variations both in prosody (i.e, duration of phonemes, speech pauses, pitch contour) and spectrum. While we believe that our BLSTM system could model these to a certain extent, it is clear from the synthesized audio samples that we would have to explore better input/output representations and advanced statistical models to improve the performance. For instance, due to heavy averaging there was muffledness in the output even after post filtering pointing to the fact that better input representations could have reduced the averaging effect. Also using spectrogram as the speech representation as in [27] instead of the MCEP, F_0 and BAP features may increase the robustness to F_0 errors in the synthesis.

As for the statistical modeling, using autoregressive models like [27] or [26] might be worthwhile investigating into. However, after conducting some preliminary experiments we suspect that a larger database might be required for such models than the one provided for the challenge.

9. Acknowledgements

The authors would like to thank Dr. Kishore Prahallad for his constant motivation to submit a system for Blizzard Challenge. The authors would also like to thank TCS for partially funding first author for his PhD. The authors also thank Srikanth Ronanki for helpful comments on preliminary results of our system and sharing *Unilex* lexicon [50].

10. References

[1] S. King, "Measuring a decade of progress in text-to-speech," *Loquens*, vol. 1, no. 1, 2014.

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

[3] 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.

[4] V. Tsiaras, R. Maia, V. Diakoloukas, Y. Stylianou, and V. Digalakis, "Linear dynamical models in speech synthesis," in *Proc. ICASSP*, 2014, pp. 300–304.

[5] Z.-H. Ling, L. Deng, and D. Yu, "Modeling spectral envelopes using restricted boltzmann machines and deep belief networks for

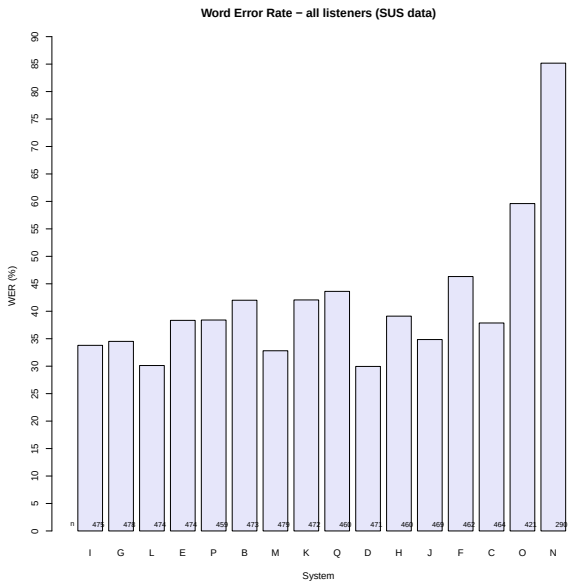

Figure 5: *Our system (M) performance in the SUS-WER with all listeners*

statistical parametric speech synthesis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2129–2139, 2013.

[6] A. W. Black, "CLUSTERGEN: A statistical parametric synthesizer using trajectory modeling," in *Proc. ICSLP*, 2006, pp. 1762–1765.

[7] A. W. Black and P. K. Muthukumar, "Random forests for statistical speech synthesis," in *Proc. INTERSPEECH*, 2015, pp. 1211–1215.

[8] T. Koriyama, T. Nose, and T. Kobayashi, "Statistical parametric speech synthesis based on Gaussian process regression," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 2, pp. 173–183, Apr. 2014.

[9] E. Raghavendra, P. Vijayaditya, and K. Prahallad, "Speech synthesis using artificial neural networks," in *Proc. National Conference on Communications*, 2010, pp. 1–5.

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

[11] O. Watts, G. E. Henter, T. Merritt, Z. Wu, and S. King, "From HMMs to DNNs: Where do the improvements come from?" in *Proc. ICASSP*, March 2016, pp. 5505–5509.

[12] K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "The effect of neural networks in statistical parametric speech synthesis," in *Proc. ICASSP*, April 2015, pp. 4455–4459.

[13] S. Pascual and A. Bonafonte, "Multi-output RNN-LSTM for multiple speaker speech synthesis and adaptation," in *Proc. EUSIPCO*, 2016, pp. 2325–2329.

[14] Y. Fan, Y. Qian, F. K. Soong, and L. He, "Unsupervised speaker adaptation for DNN-based TTS synthesis," in *Proc. ICASSP*, 2016, pp. 5135–5139.

[15] S. Achanta and S. V. Gangashetty, "Polyglot synthesis for Indian languages using zero-shot learning with recurrent neural networks," *Submitted to Signal Processing Letters, IEEE*, 2017.

[16] H. Zen and H. Sak, "Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis," in *Proc. ICASSP*, 2015, pp. 4470–4474.

[17] Y. Fan, Y. Qian, F.-L. Xie, and F. K. Soong, "TTS synthesis with bidirectional LSTM based recurrent neural networks," in *Proc. INTERSPEECH*, 2014, pp. 1964–1968.

[18] S. Achanta, T. Godambe, and S. V. Gangashetty, "An investigation of recurrent neural network architectures for statistical parametric speech synthesis," in *Proc. INTERSPEECH*, 2015, pp. 2524–2528.

[19] H. Zen and A. Senior, "Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis," in *Proc. ICASSP*, May 2014, pp. 3844–3848.

[20] P. K. Muthukumar and A. W. Black, "A deep learning approach to data-driven parameterizations for statistical parametric speech synthesis," *CoRR*, vol. abs/1409.8558, 2014. [Online]. Available: http://arxiv.org/abs/1409.8558

[21] Z. Wu, S. Takaki, and J. Yamagishi, "Deep denoising auto-encoder for statistical speech synthesis," *arXiv preprint arXiv:1506.05268*, 2015.

[22] T. Kaneko, H. Kameoka, N. Hojo, Y. Ijima, K. Hiramatsu, and K. Kashino, "Generative adversarial network-based postfilter for statistical parametric speech synthesis," in *Proc. ICASSP*, 2017.

[23] L.-H. Chen, T. Raitio, C. Valentini-Botinhao, Z.-H. Ling, and J. Yamagishi, "A deep generative architecture for postfiltering in statistical parametric speech synthesis," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 11, pp. 2003–2014, 2015.

[24] G. E. Henter, S. Ronanki, O. Watts, M. Wester, Z. Wu, and S. King, "Robust TTS duration modelling using DNNs," in *Proc. ICASSP*, March 2016, pp. 5130–5134.

[25] K. Tokuda, K. Hashimoto, K. Oura, and Y. Nankaku, "Temporal modeling in neural network based statistical parametric speech synthesis."

[26] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," *CoRR*, vol. abs/1609.03499, 2016. [Online]. Available: http://arxiv.org/abs/1609.03499

[27] Y. Wang, R. J. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. V. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: A fully end-to-end text-to-speech synthesis model," *CoRR*, vol. abs/1703.10135, 2017. [Online]. Available: http://arxiv.org/abs/1703.10135

[28] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. Courville, and Y. Bengio, "Char2wav: End-to-end speech synthesis," in *Proc. ICLR*, 2017.

[29] C. Raffel and D. P. W. Ellis, "Feed-forward networks with attention can solve some long-term memory problems," *CoRR*, vol. abs/1512.08756, 2015. [Online]. Available: http://arxiv.org/abs/1512.08756

[30] K. V. Mounika, S. Achanta, H. R. Lakshmi, S. V. Gangashetty, and A. K. Vuppala, "An investigation of deep neural network architectures for language recognition in Indian languages," in *Proc. INTERSPEECH*, 2016, pp. 2930–2933.

[31] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[32] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," *arXiv preprint arXiv:1505.00387*, 2015.

[33] S. Achanta and S. V. Gangashetty, "Deep Elman recurrent neural networks for statistical parametric speech synthesis," *Submitted to Speech Communication*, 2017.

[34] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014. [Online]. Available: http://arxiv.org/abs/1412.3555

[35] Z. Wu and S. King, "Investigating gated recurrent networks for speech synthesis," in *Proc. ICASSP*, 2016, pp. 5140–5144.

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

[37] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[38] J. Lee, K. Cho, and T. Hofmann, "Fully character-level neural machine translation without explicit segmentation," *CoRR*, vol. abs/1610.03017, 2016. [Online]. Available: http://arxiv.org/abs/1610.03017

[39] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. ICLR*, 2014.

[40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," *ArXiv e-prints*, Jun. 2017.

[41] R. J. Williams and J. Peng, "An efficient gradient-based algorithm for on-line training of recurrent network trajectories," *Neural Computation*, vol. 2, pp. 490–501, 1990.

[42] C. M. Bishop, *Neural networks for pattern recognition*. Oxford university press, 1995.

[43] K. Prahallad, "Automatic building of synthetic voices from audio books," Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, USA, 2010.

[44] T. Merritt, S. Ronanki, Z. Wu, and O. Watts, "The CSTR entry to the Blizzard Challenge 2016," in *Proc. Blizzard Challenge workshop*, 2016.

[45] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *arXiv preprint arXiv:1503.04069*, 2015.

[46] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *Proc. ICML*, 2015, pp. 2342–2350.

[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2014.

[48] T. Nose, "Efficient implementation of global variance compensation for parametric speech synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 10, pp. 1694–1704, Oct. 2016.

[49] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveign, "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. 34, pp. 187–207, 1999.

[50] S. Fitt and S. Isard, "Synthesis of regional English using a keyword lexicon." in *Proc. INTERSPEECH*, 1999, pp. 823–826.