Interspeech 2018
2-6 September 2018, Hyderabad

On The Application and Compression of Deep Time Delay Neural Network for Embedded Statistical Parametric Speech Synthesis

Yibin Zheng[1,2], *Jianhua Tao*[1,2], *Zhengqi Wen*[1], *Ruibo Fu*[1,2]

[1] National Laboratory of Pattern Recognition, Institute of Automation, CAS, China
[2] School of Artificial Intelligence, University of Chinese Academy of Science, China
{yibin.zheng, jhtao, zqwen, ruibo.fu}@nlpr.ia.ac.cn

Abstract

Acoustic models based on long short-term memory (LSTM) recurrent neural networks (RNNs) were applied to statistical parametric speech synthesis (SPSS) and shown significant improvements. However, the model complexity and inference time cost of RNNs are much higher than feed-forward neural networks (FNN) due to the sequential nature of the learning algorithm, thus limiting its usage in many runtime applications. In this paper, we explore a novel application of deep time delay neural network (TDNN) for embedded SPSS, which requires low disk footprint, memory and latency. The TDNN could model long short-term temporal dependencies with inference cost comparable to standard FNN. Temporal subsampling enabled by TDNN could reduce computational complexity. Then we compress deep TDNN using singular value decomposition (SVD) to further reduce model complexity, which are motivated by the goal of building embedded SPSS systems which can be run efficiently on mobile devices. Both objective and subjective experimental results show that, the proposed deep TDNN with SVD compression could generate synthesized speech with better speech quality than FNN and comparable speech quality to LSTM, while drastically reduce model complexity and speech parameter generation time.

Index Terms: deep TDNN, SVD, acoustic model, embedded statistical parametric speech synthesis

1. Introduction

Statistical parametric speech synthesis (SPSS) [1] based on deep neural networks (DNNs) have become dominant in text-to-speech (TTS) research area in recent years [2-21]. DNN-based acoustic models offer an efficient representation of complex dependencies between linguistic and acoustic features, and have advanced the perceived naturalness of synthesized speech [2-10]. Recurrent neural networks (RNNs) [22], especially long short-term memory (LSTM) [23], which use a dynamic changing context window over all of the sequence history rather than a fixed context window have shown their advantages in capturing long-term dependencies in sequential data, and turn out to be a great success in SPSS [9]. However, due to the recurrent connections in the RNNs, the model complexity and inference cost of RNNs are much higher than feed-forward neural networks (FNNs).

As speech synthesis technologies continue to improve, they are becoming increasingly ubiquitous on mobile devices: voices assistants such as Apple's Siri and Microsoft's Cortana could now synthesize human-like speech. Though the traditional models for these applications have been to synthesize speech remotely on large servers, there have been growing interest in developing TTS technologies that could synthesize speech directly "on-device" [24]. Some of the main challenges in this regard are the disk footprint, memory and computational constrains imposed by these devices.

Different from FNNs that maps a fixed input within a small context window to a fixed output, a time delay neural network (TDNN) [25] is able to deal with long-term temporal contexts. This architecture employs a modular and incremental design to create larger network so that the lower layers focus on modeling narrow context information, while the higher layers learn from wider temporal context information [26]. More importantly, the TDNN preserves the feed-forward structure so that the training and inference time cost are comparable with FNN. Moreover, the computation cost of TDNN could be reduced by sub-sampling its temporal connections [27]. In this paper, we propose to use TDNN for embedded statistical parametric speech synthesis. We first train a deep TDNN based acoustic model with sub-sampling to reduce TDNN complexity. Then we investigate the influences of different temporal context windows in each TDNN layer to the performance of systems. After that, we propose to apply singular value decomposition (SVD) to further reduce the ranks of affine transform matrices [28], which requires further regularization on TDNN training. With SVD compression, we could train a larger deep TDNN first, then compress it to meet the model complexity requirements afterwards. For comparison, we have trained FNN and LSTM based speech synthesis systems as baselines. Experimental results show the proposed deep TDNN with SVD compression generates synthesized speech with better speech quality than FNN and comparable speech quality to LSTM, while drastically reduce model complexity and speech parameter generation time.

The rest of this paper is organized as follows. Section 2 introduces our baseline SPSS systems. Section 3 describes the proposed deep SVD-TDNN based speech synthesis system, including sub-sampling approach and how SVD is applied for compression. Experiments and results are presented in Section 4. Conclusion remarks are shown in the final Section.

2. DNN based SPSS System

Neural networks have re-emerged as a potential powerful acoustic model for SPSS. In DNN-based SPSS, DNN is trained as a regression model to map input linguistic features into output acoustic features. In [6], a feed-forward neural network (FNN) was employed to map a linguistic representation derived from input text directly to acoustic features. However, the temporal sequence nature of speech is not explicitly modeled in the FNN architectures. In [23], a LSTM was employed to map a sequence of linguistic features to corresponding sequence of acoustic features and achieved

10.21437/Interspeech.2018-1970

great improvements. In this paper, both FNN and LSTM are used as the baseline systems.

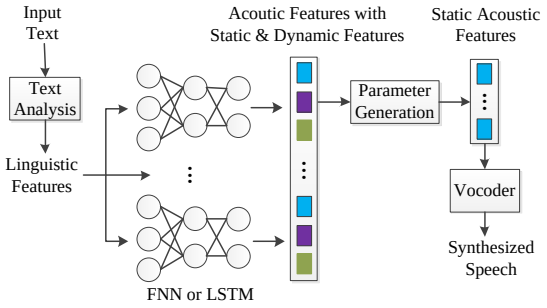


Figure1: *Architecture of FNN or LSTM based SPSS.*

Fig.1 shows the overview of the streaming SPSS architecture using FNN or LSTM. The input text is first converted into linguistic features through the text analysis, then a FNN or LSTM is employed to model the mapping between the input linguistic features and the output acoustic features. In order to generate smooth parameter trajectories, dynamic features are constraints in speech parameter generation, where predicted features are used as means vectors and the global variances of the training data are adopted for generating speech parameters by maximizing the probability. Finally, speech waveforms are generated by a vocoder with generated speech parameters.

3. Deep SVD-TDNN based embedded speech synthesis system

The proposed deep SVD-TDNN based embedded SPSS is similar to LSTM/FNN based SPSS that described in Section 2, except we replace the LSTM/FNN with deep SVD-TDNN. The SVD compression employed here follows the ideas in [28-30]. In this section, we will first review the basic architectures of TDNN, and then demonstrate how to use the deep TDNN for acoustic modeling. After that, we show how to apply the SVD compression on the deep TDNN. Finally, we give a summary of our training procedure.

3.1. Deep TDNN Architecture

3.1.1. TDNN

The basic unit in the FNN computes the weighted sum of its inputs and then pass this sum through a nonlinear function, most commonly a sigmoid or tanh function. However, in the TDNN [25], this basic unit is modified by introducing delays D_L through D_R as shown in Fig.2. The *j-th* inputs of such a unit now will be multiplied by several weights. In this way, a TDNN unit has the ability to relate and compare current input to the past history of events. The activation function for node i at time t in such a network is given by:

$$y_i^t = h(\sum_{j=1}^{i-1}\sum_{k=D_L}^{D_R} y_j^{t-k} w_{ijk}) \quad (1)$$

where y_i^t is the output of node i at time t, w_{ijk} is the connection strength to node i from output of node j at time $t-k$, and h is the activation function. In this paper, we use rectifier linear units (ReLUs) [31] as activation functions:

$$h(\cdot) = relu(x) = \begin{cases} x & x > 0 \\ 0 & x \le 0 \end{cases} \quad (2)$$

Compared with other activation fucntions, such as sigmoid or tanh, the computation cost of ReLUs is much cheaper: there is no need for computing the exponential function in ReLUs, and only comparison operation is required. As a results, it would be beneficial to improve computation effectiveness of the system.

3.1.2. Deep TDNN

To learn the complex mappings between the linguistic features and acoustic features, we can stack the TDNN to form a deep TDNN architecture. The architecture of deep TDNN used for embedded SPSS is shown in Fig.3, which is able to see longer temporal dependencies hierarchically. The figure shows the time steps at which activations are computed, at each layer, and dependencies between activations across layers. When processing a wider temporal context, the initial transforms are learnt on narrow contexts and the deeper layers process the hidden activation from a wider temporal context. Therefore, the higher layers are able to learn wider temporal relationships. Each layer in the deep TDNN operates at a different temporal resolution, which increases as we go to higher layers of the network. The parameters of each layer of TDNN are tied across different time stamps [29-30].

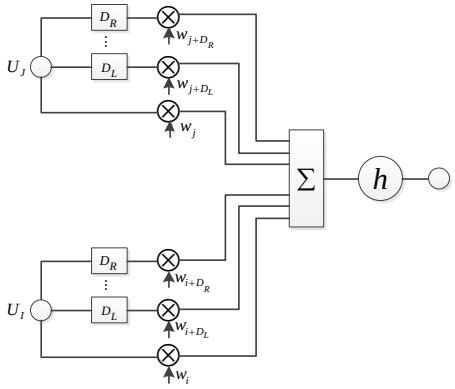


Figure 2: *A time delay neural network (TDNN) unit.*

3.2. Sub-sampling

In the regular TDNN, hidden activations are computed at all time delay steps from D_L to D_R. However, the DNN based acoustic models predict features at frame level. Thus there are large overlaps between input contexts of the activations computed at neighboring time steps. To take advantages of this property, sub-sampling [27] is employed to skip some adjacent frames in this paper. More importantly, by using sub-sampling, we could reduce the complexity of TDNN.

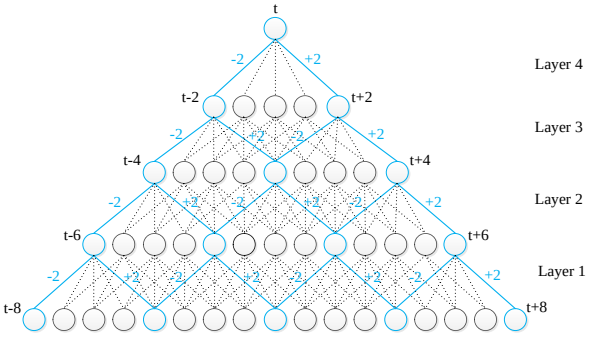
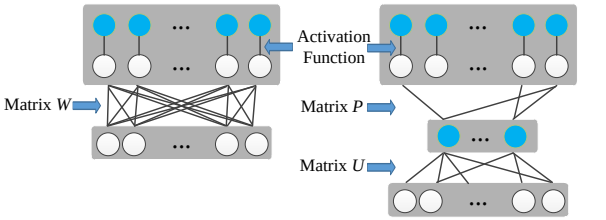


Figure 3: *Computation in TDNN with sub-sampling (blue).*

More concretely, instead of splicing together contiguous temporal windows of frames at each layer, we allow gaps between the frames. For example, the notation $\{D_L, D_R\} = \{-2, 2\}$ means we splice together the input at current frame minus 2 and the current frame plus 2. Fig.3 shows this pictorially. The temporal context windows could be set independent for each layer in deep TDNN. From our primary experimental results, we find a similar conclusion as in [32] that the model works best to splice together increasingly wide context as we go to higher layers of the network.

3.3. SVD

In this paper, we propose to further reduce the model size of deep TDNN using SVD. Fig.4 show how SVD is applied to a deep TDNN. For a $M \times N$ weight matrix W in the deep TDNN architecture, if we apply SVD on it, we get:



(A) One layer in regular TDNN model

(B) Two corresponding layers in after SVD compression

Figure 4: *Model compression in a restructured TDNN by SVD.*

$$W_{M\times N} = U_{M\times C}\sum_{C\times C} V^T_{C\times N} \quad (3)$$

where $\sum$ is a diagonal matrix with W's singular values on the diagonal in the decreasing order. We then truncate, retaining only the top K singular values and the corresponding singular vectors from $U_{M\times C}$ and $V^T_{C\times N}$ (donated by $U_{M\times K}$ and $V^T_{K\times N}$, respectively):

$$W_{M\times N} \approx U_{M\times K}\sum_{K\times K} V^T_{K\times N} = U_{M\times K} P_{K\times N} \quad (4)$$

where $P_{K\times N} = \sum_{K\times K} V^T_{K\times N}$. In this way, the weight matrix W is decomposed into two smaller matrix U and P. Fig.4(A) shows a layer in original TDNN with weight matrix $W_{M\times N}$. After SVD reconstruction, a bottleneck SVD layer of K nodes is inserted between two large hidden layers, shown in Fig.4(b). With K properly chosen, the number of multiplications as well as the model parameters could be reduced from $M \times N$ to $(M + N) \times K$. In this paper, we apply SVD to approximate all hidden layers of the proposed deep TDNN network, including the input layer.

3.4. Training procedure

The training procedure for the deep TDNN based acoustic model with SVD approximation is as follows.

- Train a full rank TDNN based acoustic model.
- Add bottleneck layers initialized by SVD of the full-rank weight matrices to the deep TDNN, layer by layer, starting from the input layer.
- Fine-turn the SVD compressed deep TDNN to compensate the precision loss caused by SVD.

The advantages of the proposed method compared to the regular full-rank TDNN is obvious, since we could train a larger TDNN first, then compress it by controlling the nodes of bottleneck layers to meet the model complexity requirements afterwards.

4. Experimental Results

4.1. Experimental setups

A Chinese Mandarin speech corpus recorded by a professional female broadcaster, both phonetically and prosodically rich, is used in our experiments. The database contains 20,000 utterances, with each utterance having around 13 words. The training/validation/test split is 8:1:1 for all the experiments. The speech signal is sampled at 16 kHz. The 60-dim line spectral pairs (LSP) features, 1-dim band aperiodicity (BAP) feature, 1-dim logarithmic fundamental frequency (log F0) together with their delta and delta-delta deviation, and voiced/unvoiced (V/UV) flag are extracted with frame shift 5-ms, and frame length 25-ms using WORLD [33]. The input features used here are the encoded 379 dimensional one-hot and numerical linguistic features obtained from our speech synthesis front-end. Both the input and output features are normalized to the zero-mean and unit-variance before model training.

To decrease the computation cost, the ReLUs function is employed as the activation function for deep TDNN training. All the full-rank deep TDNN based systems have four hidden layers in our experiments. We use exponential decaying learning rate scheduling for TDNN related systems' training. The initial learning rate is set to be 0.002 for full-rank TDNN training, and 0.0001 for SVD-compressed TDNN training. The learning rate decays by a factor 0.5 for the first few epochs, and increased to 0.9 for annealing in the training epoch.

For comparison purposes, two types of systems (described in Section 2), which are FNN and LSTM respectively, are built. These two systems serve as two baselines in this paper. The FNN based systems consist of four hidden layers, with each layer having 256 nodes. For LSTM based systems, we use three LSTM layers with 128 memory blocks for each layer to capture the long time span contextual effect of the training data. The model size of baseline FNN and LSTM are around 1.30 and 1.78 MB, respectively.

For testing, the outputs of all the systems are fed into a parameter generation module to generate smooth feature parameters with the dynamic constraints. LSP based formant enhancement is used to improve the quality of synthesized speech.

We evaluate the performance of the systems both objectively and subjectively. For objective evaluation, log spectral distance (LSD), BAPs distortion (BAPD), V/UV error rate and root mean squared error (RMSE) of the log F0 are measured. For subjective evaluation, we use both AB preference test and mean of score (MOS) test. 14 native listeners with no hearing difficulties participate in the evaluation using headphones.

4.2. Evaluation of TDNN with various temporal contexts

Since TDNN has not been applied to TTS tasks ever before, we start exploration from a deep TDNN system which is short-sighted. As donated by TDNN-A, the system uses a narrow window of {-2, -2} in each hidden layers. We then increase the temporal context windows successively, finally reaching a context window of 30 frames. Tab.1 summarizes the objective measures for different systems. For fair comparison, the model size of the full-rank deep TDNN related systems (from TDNN-A to TDNN-D) are controlled at

Table 1: *Objective measures of different systems with various layerwise temporal context windows.*

Systems	Network Context	Layerwise Context Window					Objective Measures			
		1	2	3	4	5	LSD (dB)	BAPD (dB)	V/UV Error (%)	RMSE Log F0
FNN	[-8, 8]	[-8,8]	{0}	{0}	{0}	{0}	5.013	0.186	3.561	0.113
LSTM	---	---	---	---	---	---	4.772	0.163	3.186	0.103
TDNN-A	[-8, 8]	{-2,2}	{-2,2}	{-2,2}	{-2,2}	{0}	4.873	0.179	3.401	0.111
TDNN-B	[-10, 10]	{-2,2}	{-2,2}	{-3,3}	{-3,3}	{0}	4.852	0.176	3.325	0.109
TDNN-C	[-15, 10]	{-2,2}	{-3,2}	{-5,3}	{-5,3}	{0}	4.821	0.171	3.248	0.106
TDNN-D	[-18, 12]	{-3,2}	{-3,2}	{-6,4}	{-6,4}	{0}	4.810	0.167	3.194	0.105
SVD-TDNN-C	[-15, 10]	{-2,2}	{-3,2}	{-5,3}	{-5,3}	{0}	4.831	0.172	3.306	0.106

around 1.30 MB as well. Comparing TDNN-A through TDNN-D, we could find that all the objective measures consistently drop with the increasing of the length of temporal context window. With context window equals to [-18, 12], system TDNN-D beats baseline FNN in all objective measures. Meanwhile, it also achieves comparable performance to LSTM, but with less than 73% model size, and 2.09 times (Compared to LSTM) speech parameter generation speed in our experimental results.

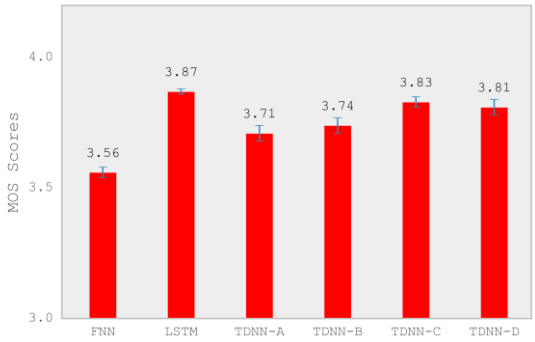


Figure 5: *Naturalness MOS results of different systems.*

We also conduct the subjective MOS evaluations to compare the performance of different systems, and the results are presented in Fig.5. From the results, we could find that system TDNN-C achieves comparable performance to both system LSTM and TDNN-D. This indicates longer time window does not bring any further benefits in the subjective evaluation. Such suggests that we definitely need to model long-term relations between speech samples in TTS tasks, but the length of dependency modeling does not need to be as long as the length of input sequence. This conclusion is consistent with that in [34]. Besides the comparable MOS score with the baseline system LSTM, system TDNN-C generates 2.34 times faster than LSTM, making it very competitive in embedded production environments. Therefore, TDNN-C is selected as the initial systems for SVD-compressed deep TDNN training.

4.3. Evaluation of SVD-compressed TDNN

We then apply SVD on all the hidden layers of TDNN-C to further reduce the footprint and computation cost of the model (donated as SVD-TDNN-C). The objective evaluation results are shown in Tab.1. It's seen the differences of the objective measures between the TDNN-C and SVD-TDNN-C are not obvious. Meanwhile, from our experimental results, the model size is further reduced (from 1.30 MB to 0.74 MB) and the generation of speech parameter becomes much faster (from 2.34 times to 2.45 times) after the SVD compression.

Table 2: *Preference scores (%) of subjective evaluation with confidence level of 0.95.*

SVD-TDNN-C	TDNN-C	LSTM	FNN	*p-value*
51.9	48.1			0.782
47.8		52.2		0.768
67.7			32.3	0.001

We further conduct an AB preference listening test to evaluate the influence of SVD from Tab.2, we could see that there is no preference among system SVD-TDNN-C, TDNN-C and LSTM. Meanwhile, system SVD-TDNN-C receives much more preference than FNN system. Also, it's worth noticing that system SVD-TDNN-C drastically reduces model complexity and speech parameter generation time.

5. Conclusions

In this paper, we present our work of building a deep TDNN based embedded SPSS, which requires low disk footprint, memory and latency. The TDNN could model long short-term temporal dependencies with inference cost comparable to standard FNN. We use temporal subsampling enabled by deep TDNN to reduce computational complexity. Then we compress deep TDNN using SVD to further reduce model complexity, which are motivated by the goal of building embedded SPSS systems which can be run efficiently on mobile devices. Our experimental results show the proposed deep TDNN with SVD compression could generate synthesized speech with better speech quality than FNN and comparable speech quality to LSTM, while drastically reduce model complexity and speech parameter generation time.

6. Acknowledgements

This work is supported by the National Natural Science Foundation of China (NSFC) (No.61425017, No. 61773379, No. 61603390, No. 61771472), the National Key Research \& Development Plan of China (No. 2017YFC0820602) and Inria-CAS Joint Research Project (173211KYSB20170061).

7. References

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

[2] H. Zen, Andrew Senior, and Mike Schuster, "Statistical parametric speech synthesis using deep neural networks," in *International Conference on Acoustics, Speech, & Signal Processing, ICASSP*, 2013.

[3] Y. Fan, Y. Qian, F.K. Soong, and L. He, "Multi-speaker modeling and speaker adaptation for DNN-based TTS synthesis," in *International Conference on Acoustics, Speech, & Signal Processing, ICASSP*, 2015, pp. 4475 – 4479.

[4] Z.Z. Wu, C. V. Botinhao, O. Watts, and S. King, "Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis," in *nternational Conference on Acoustics, Speech, & Signal Processing, ICASSP*, 2015, pp. 4460 – 4464.

[5] B. Potard, P. Motlicek, and D. Imseng, "preliminary work on speaker adaptation for DNN-based speech synthesis," *Idiap, Tech. Rep.*, 2015.

[6] J. Tao, Y. Zheng, Z. Wen, Y. Li, and B. Liu, "A BLSTM Guided Unit Selection Synthesis System for Blizzard Challenge 2016," in *Blizzard Challenge Workshop of 2016*, 2016, pp. 1 – 6.

[7] Y.Q. Yuchen Fan, F. K. Song, and L. He, "Unsupervised speaker adaptation for DNN-based tts synthesis," in *international Conference on Acoustics, Speech, & Signal Processing, ICASSP*, 2015, pp. 5135 – 5139.

[8] Y. Fan, Y. Qian, F.K. Soong, and L. He, "Speaker and language factorization in DNN-based TTS synthesis," in *international Conference on Acoustics, Speech, & Signal Processing, ICASSP*, 2016, pp. 1005–1008.

[9] H. Zen and H. Sak, "Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis," in *international Conference on Acoustics, Speech, & Signal Processing, ICASSP*, 2015, pp. 4470 – 4474.

[10] Q. Yu, P. Liu and L. Cai, "Learning cross-lingual information with multilingual BLTSM for speech synthesis of low-resource language," in *international Conference on Acoustics, Speech, & Signal Processing, ICASSP*, 2016, pp. 1233–1236.

[11] V. Digalakis, D. Rtischev, and L. Neumeyer, "Speaker adaptation using constrained reestimation of Gaussian mixtures," *IEEE Trans. Speech Audio Process.*, vol. 3, no. 5, pp. 357–366, Sep. 1995.

[12] K. Tokuda and H. Zen, "Directly modeling speech waveforms by neural networks for statistical parametric speech synthesis," in *international Conference on Acoustics, Speech, & Signal Processing, ICASSP*, 2015, pp. 4215 – 4219.

[13] Z. Wu and S. King, "Investigating gated recurrent networks for speech synthesis," in *international Conference on Acoustics, Speech, & Signal Processing, ICASSP*, 2016.

[14] Z. Z. Wu, P. Swietojanski, C. Veaux, S. King, "A study of speaker adaptation for DNN-based speech synthesis," in *INTERSPEECH 2015 – 16th Annual Conference of the International Speech Communication Association*, *Proceedings*, 2015.

[15] Y. Zheng, Y. Li, Z. Wen, et al, "Improving Prosodic Boundaries Prediction for Mandarin Speech Synthesis by Using Enhanced Embedding Feature and Model Fusion Approach," in *INTERSPEECH 2016 – 17th Annual Conference of the International Speech Communication Association*, *Proceedings*, 2016, pp. 3201–3205.

[16] A. Rosenberg, R. Fernandez, and B. Ramabhadran, "Modeling phrasing and prominence using deep recurrent learning," in *INTERSPEECH 2015 – 16th Annual Conference of the International Speech Communication Association*, *Proceedings*, 2015, pp. 3066–3070.

[17] G.E. Henter, S. Ronanki, O. Watts, et al. "Robust TTS duration modelling using DNN," in *International Conference on Acoustics, Speech, & Signal Processing, ICASSP*, 2016, pp. 5130-5134.

[18] J. Tao, Y. Zheng, Z. Wen, Y. L, et al. "A BLSTM Guided Unit Selection Synthesis System for Blizzard Challenge 2016", in *Proceeding of Blizzard Challenge 2016*.

[19] Y. Zheng, J. Tao, Z. Wen, Y. L, et al., "Investigating Efficient Feature Representation Methods and Training Objective for BLSTM-Based Phone Duration Prediction", in *INTERSPEECH 2017 – 18th Annual Conference of the International Speech Communication Association*, *Proceedings*, 2017, pp. 784-788.

[20] S. Arik, G. Diamos, A. Gibiansky, et at., "Deep Voice 2: Multi-Speaker Neural Text-to-Speech", *arXiv:*1705.08947, 2017.

[21] B. Chen, T. Bian, K. Yu, "Discrete Duration Model For Speech Synthesis", in *INTERSPEECH 2017 – 18th Annual Conference of the International Speech Communication Association*, *Proceedings*, 2017, pp. 789-793.

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

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

[24] J. Schalkwyk, D. Beeferman, F. Beaufays, "Your Word is my Command: Google Search by Voice: A Case Study," in *Advances in Speech Recognition*. Springer US, 2010:61-90.

[25] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE transactions on acoustics, speech, and signal processing*, vol. 37, no. 3, pp. 328–339, 1989.

[26] A. Waibel, "Modular construction of time-delay neural networks for speech recognition," *Neural computation*, vol. 1, no. 1, pp. 39–46, 1989.

[27] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts." In *INTERSPEECH 2015 – 16th Annual Conference of the International Speech Communication Association*, *Proceedings*, 2015, pp. 3214–3218.

[28] A. Senior and X. Lei, "Fine context, low-rank, softplus deep neural networks for mobile speech recognition," in *International Conference on Acoustics, Speech, & Signal Processing, ICASSP*, 2014, pp. 7644–7648.

[29] G. Tucker, M. Wu, M. Sun, S. Panchapagesan, G. Fu, and S. Vitaladevuni, "Model compression applied to small-footprint keyword spotting," in *INTERSPEECH 2016 – 17th Annual Conference of the International Speech Communication Association*, *Proceedings*, 2016.

[30] M. Sun, D. Snyder, Y. Gao, et al, "Compressed Time Delay Neural Network for Small-Footprint Keyword Spotting", in *INTERSPEECH 2017 – 18th Annual Conference of the International Speech Communication Association*, *Proceedings*, 2017, pp. 3607-3611.

[31] X. Glorot, A. Bordes, Y. Bengio, X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *International Conference on Artificial Intelligence and Statistics*, 2011, pp. 315–323.

[32] V. Peddinti， D. Povey， S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *INTERSPEECH 2015 – 16th Annual Conference of the International Speech Communication Association*, *Proceedings*, 2015, pp. 3214-3218.

[33] M. Morise, F, Yokomori, K. Ozawa, "WORLD, A Vocoder-Based High-Quality Speech Synthesis System for Real-Time Applications,", *Ieice Transactions on Information & Systems*, 2016, 99(7):1877-1884.

[34] M. B, H. Lu, S. Zhang, et al, "Deep Feed-forward Sequential Memory Networks for Speech Synthesis," in *international Conference on Acoustics, Speech, & Signal Processing, ICASSP*, 2018.