GATING RECURRENT MIXTURE DENSITY NETWORKS FOR ACOUSTIC MODELING IN STATISTICAL PARAMETRIC SPEECH SYNTHESIS

Wenfu Wang *Shuang Xu* *Bo Xu*

Interactive Digital Media Technology Research Center
Institute of Automation, Chinese Academy of Sciences, Beijing, P.R.China
{wangwenfu2013,shuang.xu,xubo}@ia.ac.cn

ABSTRACT

Though recurrent neural networks (RNNs) using long short-term memory (LSTM) units can address the issue of long-span dependencies across the linguistic inputs and have achieved the state-of-the-art performance for statistical parametric speech synthesis (SPSS), another limitation of the intrinsic uni-Gaussian nature of mean square error (MSE) objective function still remains. This paper proposes a gating recurrent mixture density network (GRMDN) architecture to jointly address these two problems in neural network based SPSS. What's more, the gated recurrent unit (GRU), which is much simpler and has more intelligible work mechanism than LSTM, is also investigated as an alternative gating unit in RNN based acoustic modeling. Experimental results show that the proposed GRMDN architecture can synthesize more natural speech than its MSE-trained counterpart and both the two gating units (LSTM and GRU) show comparable performance.

Index Terms— Statistical parametric speech synthesis, gating units, GRU, gating recurrent mixture density network

1. INTRODUCTION

HMM-based acoustic modeling [1] has been the mainstream approach in statistical parametric speech synthesis (SPSS) for decades of years. Even though it has flexible and robust advantages [2] over unit selection [3] approaches, the naturalness of synthesized speech is still unsatisfying. Recently, deep neural networks (DNN) based acoustic modeling techniques [4] have achieved state-of-the-art performance in SPSS. In this approach, a deep network with many stacked hidden layers is used to directly model the complex, nonlinear mapping from linguistic inputs to acoustic outputs. The success is attributed to its deep architecture that an HMM doesn't possess. A number of attempts based on DNNs for acoustic modeling have been made [5, 6]. However, a feedforward DNN has its limitation that the sequential nature of speech is ignored. Recurrent neural networks (RNNs) with long short term memory (LSTM) units [7, 8], which have capacities to capture long-term dependencies across the input sequences, have been recently employed to acoustic modeling in SPSS [9, 10, 11].

The elaborately designed gating mechanism of LSTM is the major contributor to its great success in many machine learning fields, including speech recognition [12, 13], speech synthesis [9] and statistical machine translation [14]. Recently, a novel gated recurrent unit (GRU) [15] has been proposed for RNN-based encoder in neural machine translation research. The new unit uses two gates (a reset gate and an update gate) to adaptively control the flow of information. Since its simpler architecture and more intelligible work mechanism than LSTM's, more attentions have been drawn from the community [16, 17]. Comparable performance was reported compared to LSTM and both the gating units have demonstrated noticeable superiority over the traditional sigmoid/tanh activation function. However, it has not been employed in SPSS yet.

In neural network based SPSS, there are at least two problems affecting the acoustic accuracy: *long-span dependencies in linguistic sequences* and *distribution hypothesis of acoustic features*. RNNs with LSTM or GRU can address the first problem naturally. As for the second problem, however, most of the RNN-based models in SPSS are trained by minimizing the cost function of mean square error (MSE) [9-11], which assumes the conditional distribution of output acoustic features is a single Gaussian. This is problematic for the prediction of acoustic features. It is known that the distribution of acoustic features is multimodal as human speech can vary in different styles given the same text. The conventional training approaches of neural networks using MSE cannot learn to model any more complex distributions of acoustic features than unimodal Gaussian distribution. A mixture density network (MDN) [18], which consists of a feed-forward neural network whose outputs determine the parameters of a mixture density model conditioned on the input vector to the neural networks, can alleviate the limited assumption and represent more accurate probability density functions of output features. Zen et al [19] have investigated the use of DNN based MDN (DNN-MDN) as an alternative acoustic model for SPSS and improved the naturalness of the synthesized speech. Though the limited distribution hypothesis problem was addressed using MDN, the long-span sequential problem still remains in the DNN-MDN model.

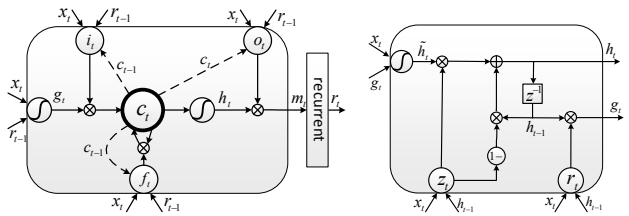
This paper proposes a novel gating recurrent mixture density network (GRMDN), which combines the gating units (LSTM and GRU) based RNNs with a mixture density model, to jointly address the two problems mentioned above. To our knowledge, this is the first time that these two problems are jointly addressed. Another novelty of our work is that GRU is investigated for the first time in SPSS for its simpler architecture than LSTM. Experimental results demonstrate that improved accuracy of GRU-RNN based acoustic modeling over DNN is achieved and both the two gating recurrent networks, GRU-RNN and LSTM-RNN, show comparable performance using MSE training criterion. To demonstrate the superiority of the proposed GRMDN, we further systematically compare the modeling capacity of GRMDNs (GRU-MDN and LSTM-MDN) with their conventional counterparts trained using MSE and DNN-MDN, respectively for SPSS. Experimental results show that the proposed architecture achieves the best naturalness of synthesized speech among all the investigated systems.

The work is supported by 973 Program in China, grant No. 2013CB329302.

The rest of the paper is organized as follows. Section 2 describes the GRMDN acoustic modeling technique for SPSS. Experimental results and analysis are presented in Section 3, and Section 4 gives the conclusions.

2. GATING RECURRENT MIXTURE DENSITY NETWORKS

A standard recurrent network (SRN) with sigmoid or hyperbolic tangent activation functions has the potential to model time sequences. However, the vanishing problem [20] caused by these activation functions prevents an SRN from learning long-span dependencies across the sequential inputs. The elaborately designed gating units, LSTM[1] and GRU as shown in Fig. 1, have demonstrated their capacity in sequential tasks.



(a) LSTM (b) GRU

Fig. 1. Architecture of LSTM and GRU.

2.1. Gated Recurrent Unit

The GRU was proposed by Cho et al [15] for encoder-decoder in neural machine translation. Fig. 1(b) illustrates the GRU implementation adopted in this paper. The activations are updated by iterating the following equations:

$$\boldsymbol{z}_t = sigm(\boldsymbol{W_{zx}}\boldsymbol{x}_t + \boldsymbol{W_{zh}}\boldsymbol{h}_{t-1} + \boldsymbol{b_z}) \tag{1}$$

$$\boldsymbol{r}_t = sigm(\boldsymbol{W_{rx}}\boldsymbol{x}_t + \boldsymbol{W_{rh}}\boldsymbol{h}_{t-1} + \boldsymbol{b_r}) \tag{2}$$

$$\boldsymbol{g}_t = \boldsymbol{r}_t \circ \boldsymbol{h}_{t-1} \tag{3}$$

$$\tilde{\boldsymbol{h}}_t = tanh(\boldsymbol{W_{hx}}\boldsymbol{x}_t + \boldsymbol{W_{hh}}\boldsymbol{g}_t + \boldsymbol{b_h}) \tag{4}$$

$$\boldsymbol{h}_t = (\boldsymbol{1} - \boldsymbol{z}_t) \circ \boldsymbol{h}_{t-1} + \boldsymbol{z}_t \circ \tilde{\boldsymbol{h}}_t \tag{5}$$

where the $\boldsymbol{W}$ terms denote weight matrices, the $\boldsymbol{b}$ terms denote bias vectors, $\boldsymbol{r}$ and $\boldsymbol{z}$ are respectively the reset gate and update gate, $\circ$ is the elementwise multiplication; the output vector $\boldsymbol{h}_t$ of the GRUs at time t is a linear interpolation between the previous output $\boldsymbol{h}_{t-1}$ and the current candidate output $\tilde{\boldsymbol{h}}_t$; $\boldsymbol{g}_t$ is the gated feedback from the previous output. One advantage of GRU is that it has less amount of parameters than LSTM without degrading performance.

2.2. Mixture Density Network

As shown in Fig. 2, an MDN can be seen as using a neural network to generate the parameters of a mixture model. Given a training sample $(\boldsymbol{x}, \boldsymbol{y})$, the conditional probability density $p(\boldsymbol{y}|\boldsymbol{x}, \mathcal{M})$[2] of the target

[1]This paper adopts the LSTM implementation in [8], where a recurrent projection layer is appended after the LSTM cells. Due to limited space, we don't give a detailed description of LSTM here since the updating equations are straightforward.

[2]For notation simplicity, the model symbol $\mathcal{M}$ in the following equations is omitted.

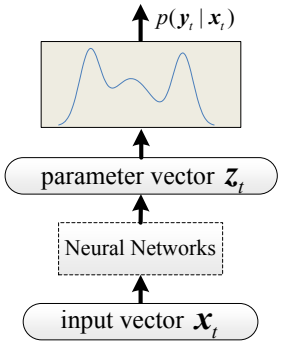


Fig. 2. Mixture Density Network.

data modeled by a mixture density network $\mathcal{M}$ can be represented as a combination of kernel functions in the form

$$p(\boldsymbol{y}|\boldsymbol{x}) = \sum_{i=1}^{M} \alpha_i(\boldsymbol{x})\phi_i(\boldsymbol{y}|\boldsymbol{x}) \tag{6}$$

where M is the number of components in the mixture, $\alpha_i(\boldsymbol{x})$ is the mixing coefficient of the i-th component. This paper constrains the kernel function to be Gaussian of the form

$$\phi_i(\boldsymbol{y}|\boldsymbol{x}) = N(\boldsymbol{y}; \boldsymbol{\mu}_i(\boldsymbol{x}), \boldsymbol{\sigma}_i^2(\boldsymbol{x})) \tag{7}$$

where $\boldsymbol{\mu}_i(\boldsymbol{x})$, $\boldsymbol{\sigma}_i^2(\boldsymbol{x})$ correspond to respectively the mean, variance of the i-th Gaussian. Given input $\boldsymbol{x}$, the parameters of Gaussian Mixture Model (GMM) can be achieved using following equations:

$$\alpha_i(\boldsymbol{x}) = \frac{\exp(z_i^{\alpha}(\boldsymbol{x}))}{\sum_{j=1}^{M}\exp(z_j^{\alpha}(\boldsymbol{x}))} \tag{8}$$

$$\boldsymbol{\mu}_i(\boldsymbol{x}) = \boldsymbol{z}_i^{\boldsymbol{\mu}}(\boldsymbol{x}) \tag{9}$$

$$\boldsymbol{\sigma}_i(\boldsymbol{x}) = \exp(\boldsymbol{z}_i^{\boldsymbol{\sigma}}(\boldsymbol{x})) \tag{10}$$

where $z_i^{\alpha}(\boldsymbol{x})$, $\boldsymbol{z}_i^{\boldsymbol{\mu}}(\boldsymbol{x})$, $\boldsymbol{z}_i^{\boldsymbol{\sigma}}(\boldsymbol{x})$ represent the corresponding network outputs. The use of softmax function in Eq. (8) ensures that the mixing coefficients are positive and sum to 1. Likewise, the exponential function applied in Eq. (10) constrains the deviations to be positive. Unlike the conventional MSE criterion that minimizes the square errors between the real outputs and targets, the MDN is trained to maximizing the log likelihood of training data. Thus the cost function is:

$$E = \sum_{n=1}^{N} \ln p(\boldsymbol{y}^n|\boldsymbol{x}^n, \mathcal{M}) \tag{11}$$

where N is the number of training samples, n is the sample index.

2.3. Gating Recurrent Mixture Density Networks based SPSS

A GRMDN combines a gating recurrent network with a mixture density model (e.g. GMM). The gating recurrent network is used to capture long-span dependencies in the linguistic context and the mixture density model can give a complete probability density of the acoustic features. In SPSS, given a linguistic input sequence $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)$ and the corresponding acoustic feature sequence $\boldsymbol{y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T)$, we aim at maximizing the conditional probability $p(\boldsymbol{y}|\boldsymbol{x})$, which factorizes as

$$p\left(\boldsymbol{y}|\boldsymbol{x}\right) = p\left(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\right)$$

$$= p(\boldsymbol{y}_1|\boldsymbol{x}_1) \cdot p\left(\boldsymbol{y}_2|\boldsymbol{y}_1, \boldsymbol{x}_1, \boldsymbol{x}_2\right) \cdots p\left(\boldsymbol{y}_T|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{T-1}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\right) \tag{12}$$

under the assumption that each frame of acoustic feature is dependent only on the current and past inputs. Further assuming output frames are conditional independent of each other, we rewrite $p(\boldsymbol{y}|\boldsymbol{x})$ as

$$p\left(\boldsymbol{y}|\boldsymbol{x}\right) = p\left(\boldsymbol{y}_1|\boldsymbol{x}_1\right) \cdot p\left(\boldsymbol{y}_2|\boldsymbol{x}_1, \boldsymbol{x}_2\right) \cdots p\left(\boldsymbol{y}_T|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\right)$$

$$= \prod_{t=1}^{T} p(\boldsymbol{y}_t|\boldsymbol{x}_{\leq t}) \tag{13}$$

where $p\left(\boldsymbol{y}_t|\boldsymbol{x}_{\leq t}\right)$ can be efficiently modeled by a GRMDN. However, a DNN-MDN or an MSE-trained RNN does not have the power.

The GRMDN based SPSS can be outlined as follows. First, a text to be synthesized is converted into a sequence of phoneme-level linguistic features through text analysis. Next, the frame-level features of each phoneme are predicted using the duration model. Then the phoneme-level and frame-level features are spliced together as inputs to the well-trained GRMDN. Through propagation, a set of parameters of GMM, the probability density over acoustic features including spectral and excitation parameters and their dynamic counterparts, can be obtained conditioned on each input vector. At each frame, the mean and variance of the component that has the highest predicted mixing coefficient are selected to form a sequence of acoustic features. Then the speech parameter generation algorithm [21] can generate smooth trajectory of speech parameters which satisfy the statistics of static and dynamic features. Finally, the speech parameters are directly fed into a vocoder to synthesize speech.

3. EXPERIMENTS

3.1. Experimental Setups

A Mandarin speech database recorded by a female professional speaker, both phonetically and prosodically rich, was used in our experiments. The database consisted of 7266 training utterances (around 7 hours, 90% as training set and the rest as development set) and 38 extra utterances for evaluation. The speech data was downsampled from 44.1 kHz to 16 kHz, then 41 line spectral pairs (LSPs), 25 band aperiodicities (BAPs) and logarithmic fundamental frequency (log F_0) were extracted every 5-ms using STRAIGHT [22].

For all the neural networks based (NN-based) systems in this paper, the input feature vector contained 462 binary features for categorical linguistic contexts (e.g. phonemes identities) and 64 numeric features for numerical linguistic contexts (e.g. the number of phonemes in current word). In addition to the phoneme-level linguistic contexts, five binary features for states indices and a numeric feature for the position of a frame in current state were appended to form frame-level identities. Each acoustic feature vector included 41 LSPs, 25 BAPs, and interpolated log F_0 , and their dynamic counterparts. A voiced/unvoiced flag was also added to the output vector to indicate the voicing condition of the current frame. For the training of NN-based systems, the input and output features were time-aligned using an HMM aligner, which was first trained using maximum likelihood criterion and then refined by minimum generation error (MGE) training to minimize the generation error between predicted and original parameter trajectories of the training data. Both the input and output features were normalized to the range of [0.01, 0.99].

For comparison, three types of architectures, which were DNN, LSTM and GRU respectively, were established, and each type was trained with and without mixture density model respectively. The DNN-related architectures were 5 hidden-layer, 1024 units per layer, with tanh activation functions. Both the LSTM- and GRU-related architectures, had two hidden layers; each layer of the LSTM-related models contained 800 memory blocks with 512 recurrent projection units while the GRU-related models had 800 units per layer. Linear and mixture density output layers were used for MSE-trained and MDN-based networks, respectively. The parameters of all the NN-based systems were first pre-trained using layerwise backpropagation, and then optimized with a mini-batch stochastic gradient descent (SGD)-based algorithm. For software implementation, the Kaldi toolkit [23] was used and training was conducted on a Tesla K40 GPU.

At synthesis time for testing utterances, the speech parameter generation algorithm[3] can generate smooth speech parameter trajectories. The conventional MSE-trained systems used the predicted output features as mean vectors and the global variances precomputed from all the training data as covariance matrices, while the MDN-based systems used the mean and covariance vectors of the component that had the highest predicted mixing coefficient at each frame.

Table 2. Preference scores (%) of different compared pairs of systems. Due to limited space, the system ID corresponding to the system in Table 1 is used. The confidence level of t-test is 95%.

Group	Compared Systems	The Former	The Latter	Neutral	p-value
1	1 vs 2	20.3	**35.0**	44.7	$< 10^{-4}$
	1 vs 3	18.0	**33.7**	48.3	$< 10^{-4}$
	2 vs 3	**23.3**	21.0	55.7	0.492
2	1 vs 6	13.3	**35.7**	51.0	$< 10^{-6}$
	2 vs 10	13.6	**29.7**	56.7	$< 10^{-4}$
	3 vs 14	12.3	**23.3**	64.4	$< 10^{-4}$
3	10 vs 14	**14.7**	10.0	75.3	0.083
4	9 vs 10	10.0	**19.3**	70.7	$< 10^{-2}$
	10 vs 11	12.0	**22.0**	66.0	$< 10^{-2}$
	13 vs 14	14.7	**27.3**	58.0	$< 10^{-3}$
	14 vs 15	17.0	**30.3**	52.7	$< 10^{-3}$

3.2. Experimental Results and Analysis

We evaluated all these NN-based systems both objectively and subjectively. 38 utterances not included in the training data were tested. To objectively evaluate the synthesis quality, voiced/unvoiced error rate, root mean squared error (RMSE) of log F_0, LSPs distortion and BAPs error were used. Though these criteria are not highly correlated to perceived quality, they provide measurable errors between predicted and target values. When performing objective test, the phoneme durations of original speech were achieved by forced alignment using the HMM aligner. The results of objective measures of different systems are presented in Table 1. The subjective evaluation was an AB preference test between a pair of synthesized speech from two chosen systems. 15 native listeners with no hearing difficulties participated in the evaluation using headphones. Each subject evaluated 20 pairs and each pair was evaluated by 10 subjects at most. After listening to each pair of synthesized speech, the subjects were asked to choose a preferred one: 1) the former was

[3] GV [24] was not considered in this experiment.

Table 1. Objective results for the MSE-trained and MDN-based systems. Totally 15 systems of different configurations are ingestigated. To facilitate later use, the systems are numbered, showed in "System ID" column. "n mix" means the mixture density model has n components. The MDN-based systems share the same architectures as their corresponding MSE-trained systems except the last output layers.

System		System ID	Architecture	LSP Distortion (dB)	BAP Error (dB)	V/UV Error Rate (%)	RMSE of log F_0
MSE	DNN	1	5×1024	1.0355	2.0556	3.861	0.1135
	LSTM	2	$2 \times (800 + 512)$	**1.0302**	**2.0452**	3.691	**0.1029**
	GRU	3	2×800	1.0319	2.0501	**3.658**	0.1061
MDN	DNN-MDN	4	2 mix	1.0361	2.0593	3.942	0.1146
		5	4 mix	1.0350	2.0556	3.754	0.1119
		6	8 mix	1.0339	2.0520	**3.666**	0.1111
		7	16 mix	**1.0327**	**2.0519**	3.728	**0.1086**
	LSTM-MDN	8	2 mix	1.0286	2.0478	3.548	0.1031
		9	4 mix	1.0263	2.0423	3.721	0.1028
		10	8 mix	1.0251	**2.0384**	3.717	0.1009
		11	16 mix	**1.0223**	2.0390	**3.375**	**0.1007**
	GRU-MDN	12	2 mix	1.0358	2.0537	3.809	0.1055
		13	4 mix	1.0356	2.0551	3.515	0.1091
		14	8 mix	1.0299	2.0456	3.518	**0.1036**
		15	16 mix	**1.0230**	**2.0424**	**3.382**	0.1040

preferred; 2) the latter was preferred; 3) neutral (the two utterances were difficult to distinguish). Table 2 shows the results of subjective preference tests. When performing subjective evaluations, totally four groups of comparison were made to comprehensively evaluate different components of the GRMDN acoustic models.

We first established three conventional MSE-trained systems, which were respectively DNN-, LSTM- and GRU-based systems (System 1, 2, 3 showed in Table 1), where the gating units based recurrent networks acoustic models were evaluated against the DNN baseline. By comparing the objective results, it can be seen that both LSTM and GRU help to predict more accurate acoustic features (e.g. LSPs) than DNN does[4]. This can be due to the strong capacity of gating units in sequence modeling. The same conclusion can be drawn from subjective test results of Group 1 in Table 2. The difference between LSTM and GRU in preference test is not statistically significant (p value of paired t-test is greater than 0.05), showing comparable performance between the two kind of gating units.

Next, the three MSE-trained systems built above were treated as baselines to assess the performance of MDN-based systems. Three MDN based models, named as DNN-MDN, LSTM-MDN and GRU-MDN corresponding to their MSE-trained versions respectively, were established. To demonstrate the superiority of MDN, each pair of models (e.g. MSE-trained LSTM based and LSTM-MDN based ones) shared the same architecture except the last output layer. Objective evaluation results of System 1 and 4 in Table 1 and subjective preference scores in Table 2 both demonstrate that DNN-MDN improves naturalness of the synthesized speech, which is consistent with that reported in [19]. The superiority of mixture density model is confirmed by comparing LSTM-MDN, GRU-MDN with LSTM and GRU respectively (Group 2 in Table 2), though the improvements are not such significant as that of DNN-MDN over DNN. We conjecture that LSTM or GRU can give context-rich outputs through capturing long-span inputs so the improvements are relatively less significant. Further, we compared the performance of LSTM-MDN and GRU-MDN. From Group 3 of subjective evaluations, it can be seen that the two kind of GRMDNs (LSTM-MDN and GRU-MDN) show comparable synthesized naturalness. This suggests a scalable superiority of LSTM or GRU over DNN whenever there is a mixture model.

The effect of having different number of mixing components in the mixture density model was investigated on GRMDNs. It can be seen from Table 1 that as the number grows, the LSP distortion, BAP error and RMSE of log F_0 all show declining trends in general. Subjective preference tests of Group 4 in Table 2 also show that having more components gives more natural synthesized speech. This demonstrates that more variability can be generated through increasing the mixing number.

[4]The contrasted result between LSTM and DNN is consistent with that in [9]

4. CONCLUSIONS

This paper proposes a novel GRMDN architecture to jointly address the two problems that affecting acoustic accuracy in neural networks based SPSS: long-span dependencies in linguistic sequences and distribution hypothesis of acoustic features. Besides, GRU is employed in RNN-based SPSS for the first time. Among all the investigated systems, the proposed GRMDN model achieves the best performance both subjectively and objectively, demonstrating the superiority over DNN-MDN or the MSE-trained counterpart. Furthermore, experimental results show that GRU exhibits comparable modeling capacity with LSTM with simpler architecture. We also explored the effect of having different number of mixing components in the mixture density model, experimental results suggest that having more components give more natural synthesized speech.

Our future work includes the application of mixture density model to bidirectional gating units based recurrent neural networks for SPSS on a larger dataset.

5. REFERENCES

[1] Takayoshi Yoshimura, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthe-

sis," in *Proceedings of 6th European Conference on Speech Communication and Technology*, 1999, pp. 2347–2350.

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

[3] Andrew J Hunt and Alan W Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proc. ICASSP*. IEEE, 1996, vol. 1, pp. 373–376.

[4] Heiga Zen, Andrew Senior, and Mike Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. ICASSP*. IEEE, 2013, pp. 7962–7966.

[5] Zhizheng Wu, Cassia Valentini-Botinhao, Oliver Watts, and Simon King, "Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis," in *Proc. ICASSP*. IEEE, 2015, pp. 4460–4464.

[6] Yuchen Fan, Yao Qian, Frank K Soong, and Lei He, "Sequence generation error (SGE) minimization based deep neural networks training for text-to-speech synthesis," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[7] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[8] Haşim Sak, Andrew Senior, and Françoise Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," *arXiv preprint arXiv:1402.1128*, 2014.

[9] Yuchen Fan, Yao Qian, Fenglong Xie, and Frank K Soong, "TTS synthesis with bidirectional LSTM based recurrent neural networks," in *Proc. Interspeech*, 2014, pp. 1964–1968.

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

[11] Peilu Wang, Yao Qian, Frank K Soong, Lei He, and Hai Zhao, "Word embedding for recurrent neural network based TTS synthesis," in *Proc. ICASSP*. IEEE, 2015, pp. 4879–4883.

[12] Yajie Miao, Mohammad Gowayyed, and Florian Metze, "EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding," *arXiv preprint arXiv:1507.08240*, 2015.

[13] Alex Graves and Navdeep Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1764–1772.

[14] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[15] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[16] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[17] Mohammad Pezeshki, "Sequence modeling using gated recurrent neural networks," *arXiv preprint arXiv:1501.00299*, 2015.

[18] Christopher M Bishop, "Mixture density networks," 1994.

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

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

[21] Keiichi Tokuda, Takayoshi Yoshimura, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. ICASSP*. IEEE, 2000, vol. 3, pp. 1315–1318.

[22] Hideki Kawahara, Ikuyo Masuda-Katsuse, and Alain De Cheveigne, "Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," *Speech communication*, vol. 27, no. 3, pp. 187–207, 1999.

[23] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukáš Burget, Ondřej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlíček, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," 2011.

[24] Toda Tomoki and Keiichi Tokuda, "A speech parameter generation algorithm considering global variance for HMM-based speech synthesis," *IEICE TRANSACTIONS on Information and Systems*, vol. 90, no. 5, pp. 816–824, 2007.