DEEP MIXTURE DENSITY NETWORKS FOR ACOUSTIC MODELING IN STATISTICAL PARAMETRIC SPEECH SYNTHESIS

Heiga Zen, Andrew Senior

Google

{heigazen,andrewsenior}@google.com

ABSTRACT

Statistical parametric speech synthesis (SPSS) using deep neural networks (DNNs) has shown its potential to produce naturally-sounding synthesized speech. However, there are limitations in the current implementation of DNN-based acoustic modeling for speech synthesis, such as the unimodal nature of its objective function and its lack of ability to predict variances. To address these limitations, this paper investigates the use of a mixture density output layer. It can estimate full probability density functions over real-valued output features conditioned on the corresponding input features. Experimental results in objective and subjective evaluations show that the use of the mixture density output layer improves the prediction accuracy of acoustic features and the naturalness of the synthesized speech.

Index Terms— Statistical parametric speech synthesis; hidden Markov models; deep neural networks; mixture density networks;

1. INTRODUCTION

Statistical parametric speech synthesis (SPSS) [1] based on hidden Markov models (HMMs) [2] offers various advantages over concatenative speech synthesis [3]. However, the naturalness of the synthesized speech from SPSS is still as 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 the acoustic models [1]. There have been many attempts to improve the accuracy, such as trajectory HMMs [4], autoregressive HMMs [5], minimum generation error (MGE) training [6], product of experts (PoEs) [7,8], Gaussian process regression (GPR) [9], and restricted Boltzmann machines (RBMs) [10, 11].

Recently, deep neural networks (DNNs) [12], which are feed-forward artificial neural networks (ANNs) with many hidden layers, have achieved significant improvement in many machine learning areas. They were also introduced as acoustic models for SPSS [13–15]. In SPSS, a number of linguistic features that affect speech, including phonetic, syllabic, and grammatical ones, have to be taken into account in acoustic modeling to achieve naturally sounding synthesized speech. In a typical implementation, there are normally around 50 different types of linguistic features [16], which is much more than those used in acoustic modeling for speech recognition. Effective modeling of these complex context dependencies is one of the most critical problems for SPSS. In DNN-based SPSS, a DNN is trained to represent the mapping function from linguistic features (inputs) to acoustic features (outputs), which are modeled by decision tree-clustered context-dependent HMMs in HMM-based SPSS [2]. DNN-based acoustic models offer an efficient and distributed representation of complex dependencies between linguistic and acoustic features [17] and have shown the potential to produce naturally-sounding synthesized speech [13, 15].

However, there are limitations in DNNs used for acoustic modeling in speech synthesis. This paper addresses the following two limitations:

- It is known that the distributions of acoustic features given linguistic features can be multimodal since humans can speak the same text in many different ways. It is also known that the outputs of an ANN trained by minimizing the squared loss function approximates the conditional mean of the outputs in the training data [18,19]. This is problematic as the average of the outputs (acoustic features) may actually be close to none of the modes of the distribution. The DNN-based acoustic model in [13], which uses the mean squared error (MSE) as its objective function to optimize its weights, does not have the power to model distributions of outputs any more complex than a unimodal Gaussian distribution.

- The outputs of an ANN provide the mean values only. The speech parameter generation algorithm [20], which has been used in SPSS, uses both the means and variances of acoustic features to find the most probable acoustic feature trajectories under the constraints between static and dynamic features. Although it has been shown experimentally that having precise variances had less impact on the naturalness of the synthesized speech than having precise means in HMM-based SPSS [21], variances are still useful to generate better acoustic feature trajectories. Furthermore, advanced generation algorithms such as the speech parameter generation algorithm considering global variance [22] relies more heavily on the variance information.

To address these limitations, this paper investigates the use of a mixture density network (MDN) [18] as an acoustic model for SPSS. MDNs can give full probability density functions over real-valued output features conditioned on the corresponding input features. This is achieved by modeling the conditional probability distribution of output features given input features with a Gaussian mixture model (GMM), where its parameters are generated using an ANN trained with a log likelihood-based loss function. The use of the MDNs allows us to do multimodal regression as well as to predict variances. In the speech synthesis-related area, MDNs have been successfully applied to articulatory-acoustic inversion mapping [23, 24].

The rest of this paper is organized as follows. Section 2 describes the MDN. Experimental results in objective and subjective evaluations are presented in Section 3. Concluding remarks are shown in the final section.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

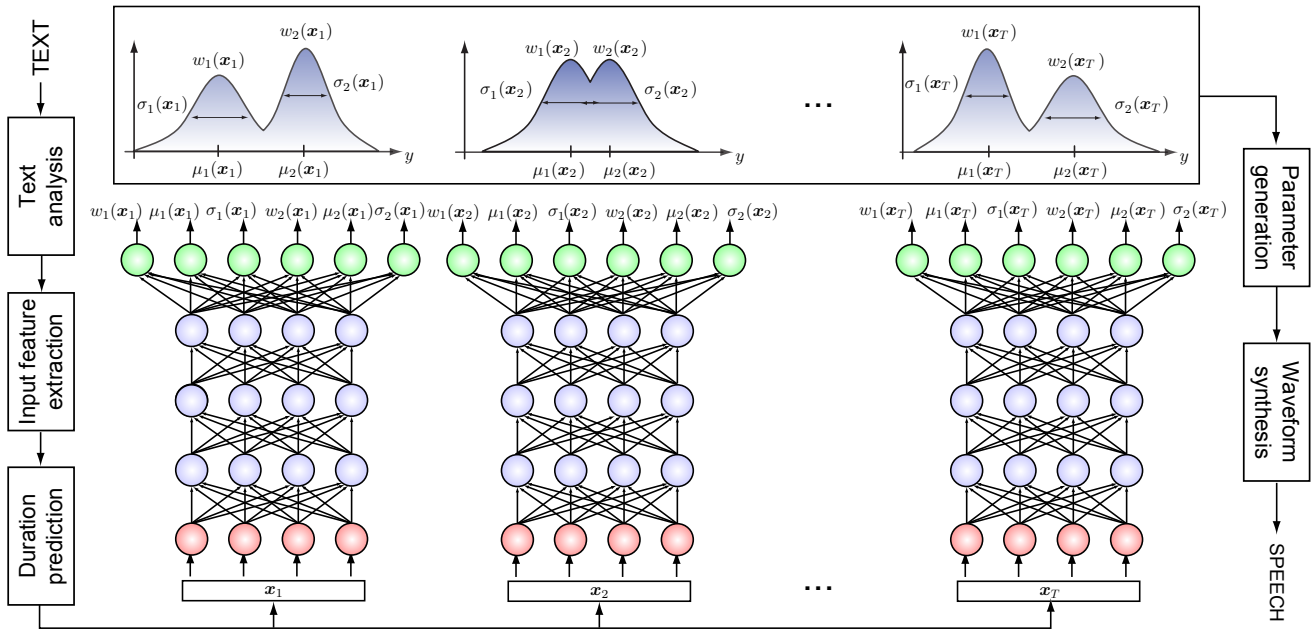

Fig. 1. Overview of the proposed SPSS framework using a deep MDN (DMDN). The red, blue, and green circles are the input, hidden, and output units, respectively. The DMDN in this example has 3 hidden layers with 4 units per hidden layer, and a mixture density output layer with 2 Gaussian components.

2. MIXTURE DENSITY NETWORK

An MDN combines a mixture model with an ANN [18]. This paper utilizes a Gaussian mixture model (GMM)-based MDN. An MDN $\mathcal{M}$ maps a set of input features $\boldsymbol{x}$ to the parameters of a GMM (mixture weights w_m, mean μ_m, and variance σ_m^2), which in turn gives a full probability density function of an output feature y, conditioned on the input features, $p(y \mid \boldsymbol{x}, \mathcal{M})$.[1] It takes the form of a GMM given as

$$p\left(y \mid \boldsymbol{x}, \mathcal{M}\right) = \sum_{m=1}^{M} w_m(\boldsymbol{x}) \cdot \mathcal{N}\left(y; \mu_m\left(\boldsymbol{x}\right), \sigma_m^2\left(\boldsymbol{x}\right)\right), \quad (1)$$

where M is the number of mixture components and $w_m(\boldsymbol{x})$, $\mu_m(\boldsymbol{x})$, and $\sigma_m^2(\boldsymbol{x})$ correspond to the mixture weight, mean, and variance of the m-th Gaussian component of the GMM, given $\boldsymbol{x}$. The GMM parameters can be derived from the MDN as

$$w_m(\boldsymbol{x}) = \frac{\exp\left(z_m^{(w)}(\boldsymbol{x}, \mathcal{M})\right)}{\sum_{l=1}^{M} \exp\left(z_l^{(w)}(\boldsymbol{x}, \mathcal{M})\right)}, \quad (2)$$

$$\sigma_m(\boldsymbol{x}) = \exp\left(z_m^{(\sigma)}(\boldsymbol{x}, \mathcal{M})\right), \quad (3)$$

$$\mu_m(\boldsymbol{x}) = z_m^{(\mu)}(\boldsymbol{x}, \mathcal{M}), \quad (4)$$

where $z_m^{(w)}(\boldsymbol{x}, \mathcal{M})$, $z_m^{(\sigma)}(\boldsymbol{x}, \mathcal{M})$, and $z_m^{(\mu)}(\boldsymbol{x}, \mathcal{M})$ are the activations of the output layer of the MDN corresponding to the mixture weight, variance, and mean for the m-th Gaussian component in the GMM, given $\boldsymbol{x}$ and $\mathcal{M}$, respectively [18]. The use of the softmax function in Eq. (2) constrains the mixture weights to be positive and sum to 1. Similarly, Eq. (3) constrains the standard deviations to be positive.

Training of the MDN aims to maximize the log likelihood of $\mathcal{M}$ given the data as

$$\hat{\mathcal{M}} = \arg\max_{\mathcal{M}} \sum_{n=1}^{N} \sum_{t=1}^{T^{(n)}} \log p\left(y_t^{(n)} \mid \boldsymbol{x}_t^{(n)}, \mathcal{M}\right) \quad (5)$$

where

$$\mathcal{D} = \left\{\left(\boldsymbol{x}_1^{(1)}, y_1^{(1)}\right), \ldots, \left(\boldsymbol{x}_{T^{(1)}}^{(1)}, y_{T^{(1)}}^{(1)}\right), \ldots, \left(\boldsymbol{x}_1^{(N)}, y_1^{(N)}\right), \ldots, \left(\boldsymbol{x}_{T^{(N)}}^{(N)}, y_{T^{(N)}}^{(N)}\right)\right\}, \quad (6)$$

is the set of input/output pairs in the training data, N is the number of utterances in the training data, and $T^{(n)}$ is the number of frames in the n-th training utterance.

Figure 1 illustrates a speech synthesis framework based on a deep MDN (DMDN). First, a text to be synthesized is converted to a sequence of linguistic features $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$. Second, the durations of each speech unit (e.g., phoneme) are predicted by a duration prediction module. Then probability distributions (GMMs) over acoustic features including spectral and excitation parameters and their dynamic features given linguistic features are predicted by a trained DMDN using forward propagation. From the sequence of the predicted GMMs, the speech parameter generation algorithm [20] can generate smooth trajectories of acoustic features which satisfy the statistics of both static and dynamic features. Finally, a waveform synthesis module outputs a synthesized waveform given the acoustic features.

[1] For simplicity of notation, here the output feature is assumed to be a scalar value. The extension to a vector is straightforward.

3. EXPERIMENTS

3.1. Experimental Conditions

Speech data in US English from a female professional speaker was used for training speaker-dependent HMM-, DNN-, and DMDN-based SPSS. The training data consisted of about 33 000 utterances. The speech analysis conditions and model topologies were similar to those used for the Nitech-HTS 2005 [25] system. The speech data was downsampled from 48 kHz to 16 kHz sampling, then 40 mel-cepstral coefficients [26], logarithmic fundamental frequency ($\log F_0$) values, and 5-band aperiodicities (0–1, 1–2, 2–4, 4–6, 6–8 kHz) [25] were extracted every 5 ms. Each observation vector consisted of 40 mel-cepstral coefficients, $\log F_0$, and 5 band aperiodicities, and their velocity and acceleration features ($3 \times (40 + 1 + 5) = 138$). Five-state, left-to-right, no-skip hidden semi-Markov models (HSMMs) [27] were used. To model $\log F_0$ sequences consisting of voiced and unvoiced observations, a multi-space probability distribution (MSD) was used [28]. The number of questions for the decision tree-based context clustering was 2 554. The sizes of decision trees in the HMM-based systems were controlled by changing the scaling factor α for the model complexity penalty term of the minimum description length (MDL) criterion [29] ($\alpha = 1$). The numbers of leaf nodes for mel-cepstrum, $\log F_0$, and band aperiodicities were 12 578, 32 847, and 436, respectively.

The input features for the DNN- and DMDN-based systems included 342 binary features for categorical linguistic contexts (*e.g.* phonemes identities, stress marks) and 25 numerical features for numerical linguistic contexts (*e.g.* the number of syllables in a word, position of the current syllable in a phrase). In addition to the linguistic contexts-related input 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. The output features were basically the same as those used in the HMM-based systems. To model $\log F_0$ sequences, the continuous F_0 with explicit voicing modeling approach [30] was used; voiced/unvoiced binary value was added to the output features and $\log F_0$ values in unvoiced frames were interpolated. To reduce the computational cost, 80% of silence frames were removed from the training data. The weights of the networks were initialized randomly (no pretraining was performed), then optimized; The weights of the DNN-based systems were trained to minimize the mean squared error between the output features of the training data and predicted values using, whereas those of the DMDN-based systems were trained to maximize the log likelihood of the model given the training data. A GPU implementation of a minibatch stochastic gradient descent (SGD)-based back-propagation algorithm was used. To schedule the learning rate of the minibatch stochastic gradient descent (SGD)-based back-propagation algorithm, AdaDec [31] was used.[2] Both input and output features in the training data were normalized; 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 rectifier linear activation function (ReLU) [33] was used in hidden layers.[3] Linear and mixture density output layers were used for the DNN- and DMDN-based systems, respectively. In each case a single network was trained to model both spectral and excitation parameters.

Speech parameters for the evaluation sentences were generated from the models using the speech parameter generation algorithm [20].[4] The DNN-based systems used the per-dimension variances computed from all training data whereas the DMDN-based systems used the ones predicted by the network. While generating the acoustic features from the HMMs and DMDNs, the mixture component that had the highest predicted mixture weight was selected at each frame.[5] Spectral enhancement based on post-filtering in the cepstral domain [34] was applied to improve the naturalness of the synthesized speech. From the generated speech parameters, speech waveforms were synthesized using the source-filter model.

To objectively evaluate the performance of the HMM-, DNN-, DMDN-based systems, mel-cepstral distortion (dB) [35], linear aperiodicity distortion (dB), voiced/unvoiced error rate (%), and root mean squared error (RMSE) in $\log F_0$ were used.[6] Phoneme durations from natural speech were used while performing objective and subjective evaluations. To subjectively evaluate the performance of the systems, preference and mean opinion score (MOS) tests were also conducted. 173 utterances not included in the training data were used for evaluation. One subject could evaluate a maximum of 30 pairs in the preference tests and 30 stimuli in the MOS tests. Each pair was evaluated by five subjects in the preference tests, whereas each stimulus was evaluated by three subjects in the MOS tests. 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).

Table 2. Preference scores (%) between speech samples from the DNN (4 hidden layers, 1024 units per hidden layer) and DMDNs (4 hidden layers, 1024 units per hidden layer, mixture density output layer with 1, 4, or 16 mixture components).

DNN	DMDN					
	1mix	4mix	16mix	Neutral	p-value	z-score
11.6	**17.9**	–	–	70.5	$< 10^{-3}$	-3.5
8.8	–	–	**28.3**	62.9	$< 10^{-6}$	-11.1
–	6.7	**16.1**	–	77.2	$< 10^{-6}$	-6.3
–	9.2	–	**18.3**	72.5	$< 10^{-6}$	-5.4

3.2. Experimental Results

Table 1 shows the objective measures and the mean opinion scores for all architectures. Table 2 also shows the results of the subjective preference listening tests to evaluate the effect of the mixture density output layer.

[2] AdaDec is a variant of AdaGrad [32], which can manage the learning rate on per-parameter basis. Preliminary experiments showed that AdaGrad and AdaDec gave faster convergence and more stable optimization while training MDNs, which had heterogeneous parameter types (means, standard deviations, and mixture weights) requiring different learning rates.

[3] A preliminary experiment showed that DNNs with the ReLU activation functions in hidden layers achieved better objective measures.

[4] The generation algorithm considering global variance [22] was not investigated in this experiment.

[5] Although the case 3 algorithm of [20], which is based on the EM algorithm and can marginalize hidden variables, can also be used, a preliminary experiment showed that the differences in the objective measures between choosing the most probable mixture component and marginalizing them was negligible. Furthermore, the case 3 algorithm requires more computations.

[6] These criteria are not highly correlated to the naturalness of synthesized speech. However they have been used to objectively measure the prediction accuracy of acoustic models.

Table 1. Voiced/unvoiced error rates (%), root mean squared errors (RMSEs) in log F_0, mel-cepstral distortions (dB), band aperiodicity distortions (dB), and 5-scale MOSs of the HMM-, DNN-, and DMDN-based systems with different architectures. In this table, "$L \times X$" means L hidden layers with X units per hidden layer, and "M mix" means M components at the mixture density output layer.

Model	Architecture	Number of parameters ($\times 10^6$)	V/UV error rates (%)	log F_0 RMSE	Mel-cepstral distortion (dB)	Band aperiodicity distortion (dB)	5-scale MOS
HMM	1 mix	3.267	4.293	**0.1232**	**4.820**	**1.263**	**3.537 ± 0.113**
	2 mix	6.548	**4.275**	0.1275	4.895	**1.263**	3.397 ± 0.115
DNN	4×1024	3.673	3.505	0.1243	4.794	1.222	3.635 ± 0.127
	5×1024	4.723	**3.411**	0.1225	4.542	1.199	**3.681 ± 0.109**
	6×1024	5.772	3.477	**0.1221**	**4.526**	**1.198**	3.652 ± 0.108
	7×1024	6.822	3.495	0.1225	4.537	1.200	3.637 ± 0.129
DMDN (4×1024)	1 mix	3.818	3.752	0.1217	4.637	1.204	3.654 ± 0.117
	2 mix	3.962	3.342	0.1191	**4.541**	1.201	3.796 ± 0.107
	4 mix	4.251	**3.399**	0.1193	4.565	**1.200**	3.766 ± 0.113
	8 mix	4.829	3.340	0.1190	4.553	1.202	**3.805 ± 0.113**
	16 mix	5.986	3.383	**0.1188**	4.543	1.203	3.791 ± 0.102

3.2.1. Having variances

The effect of having variances can be seen by comparing the DNN (4×1024) and the DMDN (4×1024, 1 mix). Although there was no significant difference between them in the mean opinion scores, the preference test results show that the DMDN (4×1024, 1 mix) was more preferred to the DNN (4×1024). As DNNs were trained to minimize the squared error between data and predicted values and DMDNs were trained to maximize the log likelihood of the model given data, the DMDN had to get worse in the squared error-based measures. However, it can be seen from the tables that having variances was helpful in predicting mel-cepstra and band aperiodicity and improved the naturalness of the synthesized speech. This can be due to the speech parameter generation algorithm, which determines smoothly-varying acoustic feature trajectories using both means and variances. To check this, an additional experiment was conducted. The variances predicted by the DMDN (4×1024, 1 mix) rather than global ones were used with the means predicted by the DNN (4×1024, 1 mix) as inputs of the speech parameter generation algorithm. Table 3 shows experimental results. It can be seen from the tables that the use of the variances predicted by the DMDN with the means predicted by the DNN achieved small improvements. However, it was not as good as the DMDN.

Table 3. RMSE in log F_0, mel-cepstral distortion (dB), and band aperiodicity distortion (dB) of the DNN-based system (4×1024) with variances predicted by the DMDN-based system (4×1024, 1 mix).

log F_0 RMSE	Mel-cepstral distortion (dB)	Band aperiodicity distortion (dB)
0.1240	4.783	1.221

3.2.2. Having multiple components

The effect of having multiple Gaussian components can be found by contrasting the DMDN with 1 mixture component and those with multiple mixture components. It can be seen from the table that having multiple components was helpful in predicting log F_0 and improved the naturalness of the synthesized speech. This is reasonable as there can be multiple possible naturally-sounding F_0 contours for the same texts. Having multiple components can help capturing such phenomena. It can also be seen from the preference and MOS test results that having multiple components improved the naturalness of the synthesized speech significantly. The MOS test results also showed that having mixture density output layer is more efficient than having more layers. For example, although DNN (5×1024) and DMDN (4×1024, 4 mix) had the similar numbers of parameters, the DMDN (4×1024, 4 mix) achieved better mean opinion score than the DNN (5×1024).

Overall, the DMDN (4×1024, 8 mix) achieved 3.803 in the 5-scale MOS, which was 0.266 better than the standard HMM-based system.

4. CONCLUSIONS

This paper has extended DNN-based SPSS by introducing mixture density networks (MDNs). The proposed DMDN-based approach can relax the limitations in the DNN-based acoustic modeling for speech synthesis: the lack of variances and the unimodal nature of the objective function. Objective and subjective evaluations showed that having variances and multiple mixture components by using a mixture density output layer was helpful in predicting acoustic features more accurately and improved the naturalness of the synthesized speech significantly.

Future work includes exploring better network architectures and optimization algorithms to train networks. Evaluation of DMDNs with the speech parameter generation algorithm considering global variance is also necessary.

5. ACKNOWLEDGEMENTS

The authors would like to thank Dr. Mike Schuster 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] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," in *Proc. Eurospeech*, 1999, pp. 2347–2350.

[3] 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.

[4] H. Zen, K. Tokuda, and T. Kitamura, "Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic features," *Comput. Speech Lang.*, vol. 21, no. 1, pp. 153–173, 2007.

[5] 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.

[6] Y.-J. Wu and R.-H. Wang, "Minimum generation error training for HMM-based speech synthesis," in *Proc. ICASSP*, 2006, pp. 89–92.

[7] Y. Qian, Z.-Z. Wu, B.-Y. Gao, and F. Soong, "Improved prosody generation by maximizing joint probability of state and longer units," *IEEE Trans. Acoust. Speech Lang. Process.*, vol. 19, no. 6, pp. 1702–1710, 2011.

[8] H. Zen, M. Gales, Y. Nankaku, and K. Tokuda, "Product of experts for statistical parametric speech synthesis," *IEEE Trans. Audio Speech Lang. Process.*, vol. 20, no. 3, pp. 794–805, 2012.

[9] T. Koriyama, T. Nose, and T. Kobayashi, "Statistical parametric speech synthesis based on Gaussian process regression," *IEEE Journal of Selected Topics in Signal Process.*, 2013.

[10] S.-Y. Kang, X.-J. Qian, and H. Meng, "Multi-distribution deep belief network for speech synthesis," in *Proc. ICASSP*, 2013, pp. 8012–8016.

[11] Z.-H. Ling, L. Deng, and D. Yu, "Modeling spectral envelopes using restricted Boltzmann machines and deep belief networks for statistical parametric speech synthesis," *IEEE Trans. Acoust. Speech Lang. Process.*, vol. 21, no. 10, pp. 2129–2139, 2013.

[12] 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.

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

[14] 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.

[15] 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.

[16] 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.

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

[18] C. Bishop, "Mixture density networks," Tech. Rep. NCRG/94/004, Neural Computing Research Group, Aston University, 1994.

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

[20] 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.

[21] K. Oura, H. Zen, Y. Nankaku, A. Lee, and K. Tokuda, "Tying covariance matrices to reduce the footprint of HMM-based speech synthesis systems," in *Proc. Interspeech*, 2009, pp. 1759–1762.

[22] 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.

[23] K. Richmond, "Trajectory mixture density networks with multiple mixtures for acoustic-articulatory inversion," in *Advances in Nonlinear Speech Processing*, pp. 263–272. Springer, 2007.

[24] B. Uria, I. Murray, S. Renals, and K. Richmond, "Deep architectures for articulatory inversion," in *Proc. Interspeech*, 2012, pp. 867–870.

[25] 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.

[26] 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.

[27] H. Zen, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "A hidden semi-Markov model-based speech synthesis system," *IEICE Trans. Inf. Syst.*, vol. E90-D, no. 5, pp. 825–834, 2007.

[28] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi, "Multi-space probability distribution HMM," *IEICE Trans. Inf. Syst.*, vol. E85-D, no. 3, pp. 455–464, 2002.

[29] K. Shinoda and T. Watanabe, "Acoustic modeling based on the MDL criterion for speech recognition," in *Proc. Eurospeech*, 1997, pp. 99–102.

[30] 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.

[31] 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.

[32] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *The Journal of Machine Learning Research*, pp. 2121–2159, 2011.

[33] 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.

[34] 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.

[35] 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.