STATISTICAL PARAMETRIC SPEECH SYNTHESIS USING DEEP NEURAL NETWORKS

Heiga Zen, Andrew Senior, Mike Schuster

Google

{heigazen,andrewsenior,schuster}@google.com

ABSTRACT

Conventional approaches to statistical parametric speech synthesis typically use decision tree-clustered context-dependent hidden Markov models (HMMs) to represent probability densities of speech parameters given texts. Speech parameters are generated from the probability densities to maximize their output probabilities, then a speech waveform is reconstructed from the generated parameters. This approach is reasonably effective but has a couple of limitations, *e.g.* decision trees are inefficient to model complex context dependencies. This paper examines an alternative scheme that is based on a deep neural network (DNN). The relationship between input texts and their acoustic realizations is modeled by a DNN. The use of the DNN can address some limitations of the conventional approach. Experimental results show that the DNN-based systems outperformed the HMM-based systems with similar numbers of parameters.

Index Terms— Statistical parametric speech synthesis; Hidden Markov model; Deep neural network;

1. INTRODUCTION

Statistical parametric speech synthesis based on hidden Markov models (HMMs) [1] has grown in popularity in the last decade. This approach has various advantages over the concatenative speech synthesis approach [2], such as the flexibility to change its voice characteristics, [3–6], small footprint [7–9], and robustness [10]. However its major limitation is the quality of the synthesized speech. Zen *et al.* [11] highlighted three major factors that degrade the quality of the synthesized speech: vocoding, accuracy of acoustic models, and over-smoothing. This paper addresses the accuracy of acoustic models.

A number of contextual factors that affect speech including phonetic, linguistic, and grammatical ones have been taken into account in acoustic modeling for statistical parametric speech synthesis. In a typical system, there are normally around 50 different types of contexts [12]. Therefore, effective modelling of these complex context dependencies is one of the most critical problems for statistical parametric speech synthesis. The standard approach to handling contexts in HMM-based statistical parametric speech synthesis is to use a distinct HMM for each individual combination of contexts, referred to as a context-dependent HMM. The amount of available training data is normally not sufficient for robustly estimating all context-dependent HMMs since there is rarely sufficient data to cover all of the context combinations required. To address these problems, top-down decision tree based context clustering is widely used [13]. In this approach, the states of the context-dependent HMMs are grouped into "clusters" and the distribution parameters within each cluster are shared. The assignment of HMMs to clusters is performed by examining the context combination of each HMM through a binary decision tree, where one context-related binary question is associated with each non-terminal node. The number of clusters, namely the number of terminal nodes, determines the model complexity. The decision tree is constructed by sequentially selecting the questions which yield the largest log likelihood gain of the training data. The size of the tree is controlled using a pre-determined threshold of log likelihood gain, a model complexity penalty [14, 15], or cross validation [16, 17]. With the use of context-related questions and state parameter sharing, the unseen contexts and data sparsity problems are effectively addressed. As the method has been successfully used in speech recognition, HMM-based statistical parametric speech synthesis naturally employs a similar approach to model very rich contexts.

Although the decision tree-clustered context-dependent HMMs work reasonably effectively in statistical parametric speech synthesis, there are some limitations. First, it is inefficient to express complex context dependencies such as *XOR*, parity or multiplex problems by decision trees [18]. To represent such cases, decision trees will be prohibitively large. Second, this approach divides the input space and use separate parameters for each region, with each region associated with a terminal node of the decision tree. This results in fragmenting the training data and reducing the amount of the data that can be used in clustering the other contexts and estimating the distributions [19]. Having a prohibitively large tree and fragmenting training data will both lead to overfitting and degrade the quality of the synthesized speech.

To address these limitations, this paper examines an alternative scheme that is based on a deep architecture [20]. The decision trees in HMM-based statistical parametric speech synthesis perform mapping from linguistic contexts extracted from text to probability densities of speech parameters. Here decision trees are replaced by a deep neural network (DNN). Until recently, neural networks with one hidden layer were popular as they can represent arbitrary functions if they have enough units in the hidden layer. Although it is known that neural networks with multiple hidden layers can represent some functions more efficiently than those with one hidden layer, learning such networks was impractical due to its computational costs. However, the recent progress both in hardware (*e.g.* GPU) and software (*e.g.* [21]) enables us to train a DNN from a large amount of training data. Deep neural networks have achieved large improvements over conventional approaches in various machine learning areas including speech recognition [22] and acoustic-articulatory inversion mapping [23]. Note that NNs have been used in speech synthesis since the 90s (e.g. [24]).

This paper is organized as follows. Section 2 contrasts the difference between the decision tree and DNNs. Section 3 describes the DNN-based statistical parametric speech synthesis framework. Experimental results are presented in Section 4. Concluding remarks are shown in the final section.

2. DEEP NEURAL NETWORK

Here the depth of architecture refers to the number of levels of composition of non-linear operations in the function learned. It is known that most conventional learning algorithms correspond to *shallow* architectures (≤ 3 levels) [20]. For example, both the decision tree and neural network with 1 hidden layer can be seen as having 2 levels.[1] Boosting [25], tree intersections [19, 26, 27], or product of decision tree-clustered experts [28] add one level to the base learner (*i.e.* 3 levels). A DNN, which is a neural network with multiple hidden layers, is a typical implementation of a *deep* architecture. We can have a deep architecture by adding multiple hidden layers to a neural network (adding one layer results in having one more level).

The properties of the DNN are contrasted with those of the decision tree as follows;

- Decision trees are inefficient to express complicated functions of input features, such as *XOR*, d-bit parity function, or multiplex problems [18]. To represent such cases, decision trees will be prohibitively large. On the other hand, they can be compactly represented by DNNs [20].
- Decision trees rely on a partition of the input space and using a separate set of parameters for each region associated with a terminal node. This results in reduction of the amount of the data per region and poor generalization. Yu *et al.* showed that "weak" input features such as word-level emphasis in reading speech were thrown away while building decision trees [29]. DNNs provide better generalization as weights are trained from all training data. They also offer incorporation of high-dimensional, disparate features as inputs.
- Training a DNN by back-propagation usually requires a much larger amount of computation than building decision trees. At the prediction stage, DNNs require a matrix multiplication at each layer but decision trees just need traversing trees from their root to terminal nodes using a subset of input features.
- The decision trees induction can produce interpretable rules while weights in a DNN are harder to interpret.

3. DNN-BASED SPEECH SYNTHESIS

Inspired by the human speech production system which is believed to have layered hierarchical structures in transforming the information from the linguistic level to the waveform level [30], this paper applies a deep architecture to solve the speech synthesis problem.

Figure 1 illustrates a speech synthesis framework based on a DNN. A given text to be synthesized is first converted to a sequence of input features $\{x_n^t\}$, where x_n^t denotes the n-th input feature at frame t. The input features include binary answers to questions about linguistic contexts (*e.g.* is-current-phoneme-aa?) and numeric values (*e.g.* the number of words in the phrase, the relative position of the current frame in the current phoneme, and durations of the current phoneme).

Then the input features are mapped to output features $\{y_m^t\}$ by a trained DNN using forward propagation, where y_m^t denotes the m-th output feature at frame t. The output features include spectral and excitation parameters and their time derivatives (dynamic features) [31]. The weights of the DNN can be trained using pairs of input and output features extracted from training data. In the same fashion as the HMM-based approach, it is possible to generate speech parameters; By setting the predicted output features from the DNN as mean vectors and pre-computed variances of output features from all training data as covariance matrices, the speech parameter generation algorithm [32] can generate smooth trajectories of speech parameter features which satisfy both the statistics of static and dynamic features. Finally, a waveform synthesis module outputs a synthesized waveform given the speech parameters.

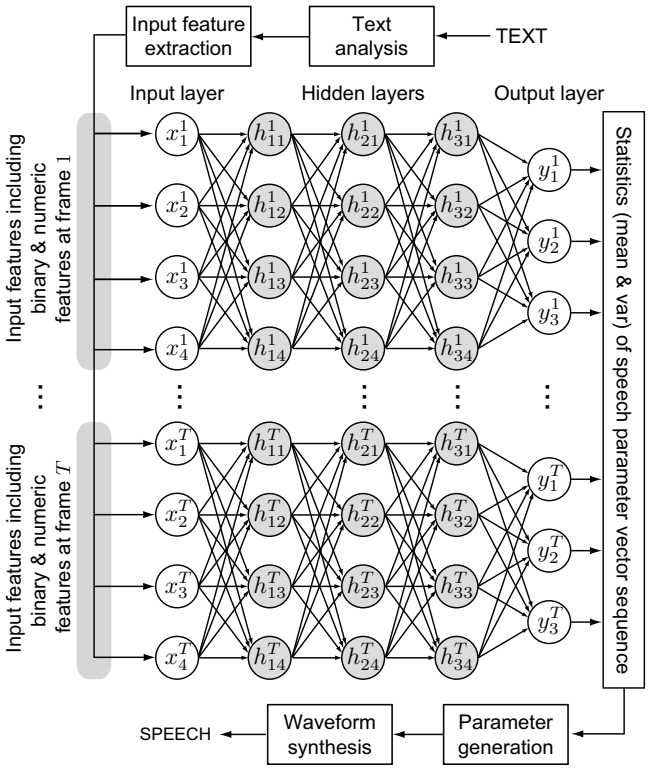

Fig. 1. A speech synthesis framework based on a DNN.

Note that the text analysis, speech parameter generation, and waveform synthesis modules of the DNN-based system can be shared with the HMM-based one, *i.e.* only the mapping module from context-dependent labels to statistics needs to be replaced.

4. EXPERIMENTS

4.1. Experimental conditions

Speech data in US English from a female professional speaker was used for training speaker-dependent HMM-based and DNN-based statistical parametric speech synthesizers. 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 [33] system. The speech data was downsampled from 48 kHz to 16 kHz sampling, then 40 Mel-cepstral coefficients [34], logarithmic fundamental frequency ($\log F_0$) values, and 5-band aperiodicities (0–1, 1–2, 2–4, 4–6, 6–8 kHz) [33] were extracted every 5 ms. Each observation vector consisted of 40 Mel-cepstral coefficients, $\log F_0$, and 5 band aperiodicities, and their delta and delta-delta features ($3 \times (40 + 1 + 5) = 138$). Five-state, left-to-right, no-skip hidden semi-Markov models (HSMMs) [35] were used. To model $\log F_0$ sequences consisting of voiced and unvoiced observa-

[1] Partition of an input feature space by a decision tree can be represented by a composition of *OR* and *AND* operation layers.

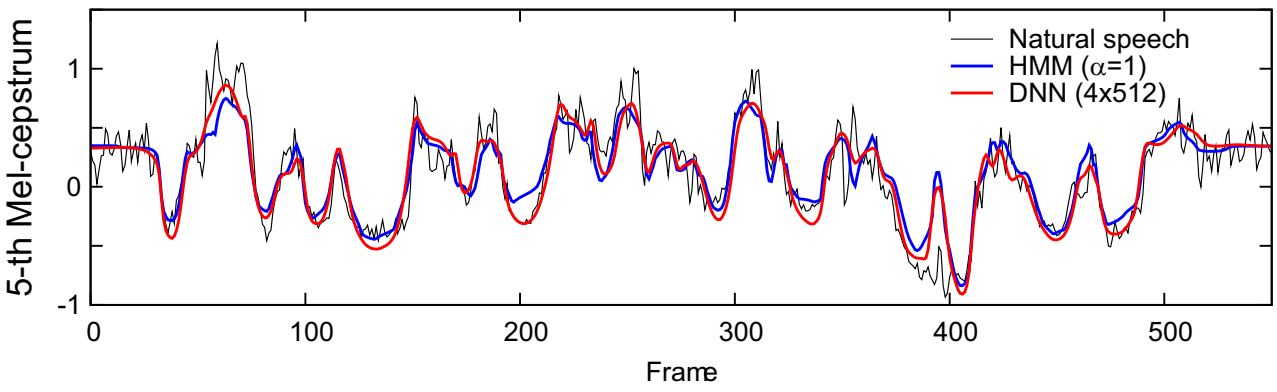

Fig. 2. Trajectories of 5-th Mel-cepstral coefficients of natural speech and those predicted by the HMM and DNN-based systems.

tions, a multi-space probability distribution (MSD) was used [36]. 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 complexiby penalty term of the minimum description length (MDL) criterion [14] ($\alpha = 16, 8, 4, 2, 1, 0.5, 0.375,$ or 0.25).[2] When $\alpha = 1$, the number of leaf nodes for Mel-cepstrum, log F_0, and band aperiodicities were 12 342, 26 209, and 401, respectively (3 209 991 parameters[3] in total).

The input features for the DNN-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).[4] 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 by a DNN, the continuous F_0 with explicit voicing modeling approach [37] 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 DNN were initialized randomly, then optimized to minimize the mean squared error between the output features of the training data and predicted values using a GPU implementation of a minibatch stochastic gradient descent (SGD)-based back-propagation algorithm. Both input and output features in the training data for the DNN 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 sigmoid activation function was used for hidden and output layers.[5] A single network which modeled both spectral and excitation parameters was trained.

Speech parameters for the evaluation sentences were generated from the models using the speech parameter generation algorithm [32].[6] Spectral enhancement based on post-filtering in the cepstral domain [39] 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 and DNN-based systems, Mel-cepstral distortion (dB) [40], linear aperiodicity distortion (dB), voiced/unvoiced error rate (%), and root mean squared error (RMSE) in log F_0 were used.[7] Segmentations (phoneme durations) from natural speech were used while performing objective and subjective evaluations.[8] 173 utterances not included in the training data were used for evaluation.

4.2. Objective evaluation

Figure 2 plots the trajectories of 5-th Mel-cepstral coefficients of natural speech and predicted by the HMM and DNN-based systems. It can be seen from the figure that both systems could predict reasonable speech parameter trajectories for a given text.

In the objective evaluation we investigated the relationship between the prediction performance and architecture of the DNN; the number of layers (1, 2, 3, 4, or 5) and units per layer (256, 512, 1 024, or 2 048). Figure 3 plots the experimental results. The DNN-based systems consistently outperformed the HMM-based ones in voiced/unvoiced classification and aperiodicity prediction. The DNN-based systems with many layers were similar to or better than the HMM-based ones in Mel-cepstral distortion. On the other hand, the HMM-based systems outperformed the DNN-based ones in log F_0 prediction in most cases. Currently all unvoiced frames were interpolated and modeled as voiced frames. We expect that this scheme degrades the prediction performance for log F_0 as these interpolated frames may introduce a bias to the estimated DNN. For Mel-cepstrum and aperiodicity prediction, having multiple layers tended to work better than having more units per layer.

[2] As α increases, the sizes of decision trees decrease. Typical HMM-based speech synthesis systems use $\alpha = 1$.

[3] Each leaf node for Mel-cepstrum, log F_0, and band aperiodicities had 240, 9, and 30 parameters (means, variances, and MSD weights), respectively.

[4] We also tried to encode numerical features to binary ones by applying questions such as "is-the-number-of-words-in-a-phrase-less-than-5". A preliminary experiment showed that using numerical features directly worked better and more efficiently than encoding them to binary ones.

[5] Although the linear activation function is popular in DNN-based regression, our preliminary experiments showed that the DNN with the sigmoid activation function at the output layer consistently outperformed those with the linear one.

[6] The generation algorithm considering global variance [38] was not investigated in this experiment.

[7] 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.

[8] Durations can also be predicted by a separate DNN.

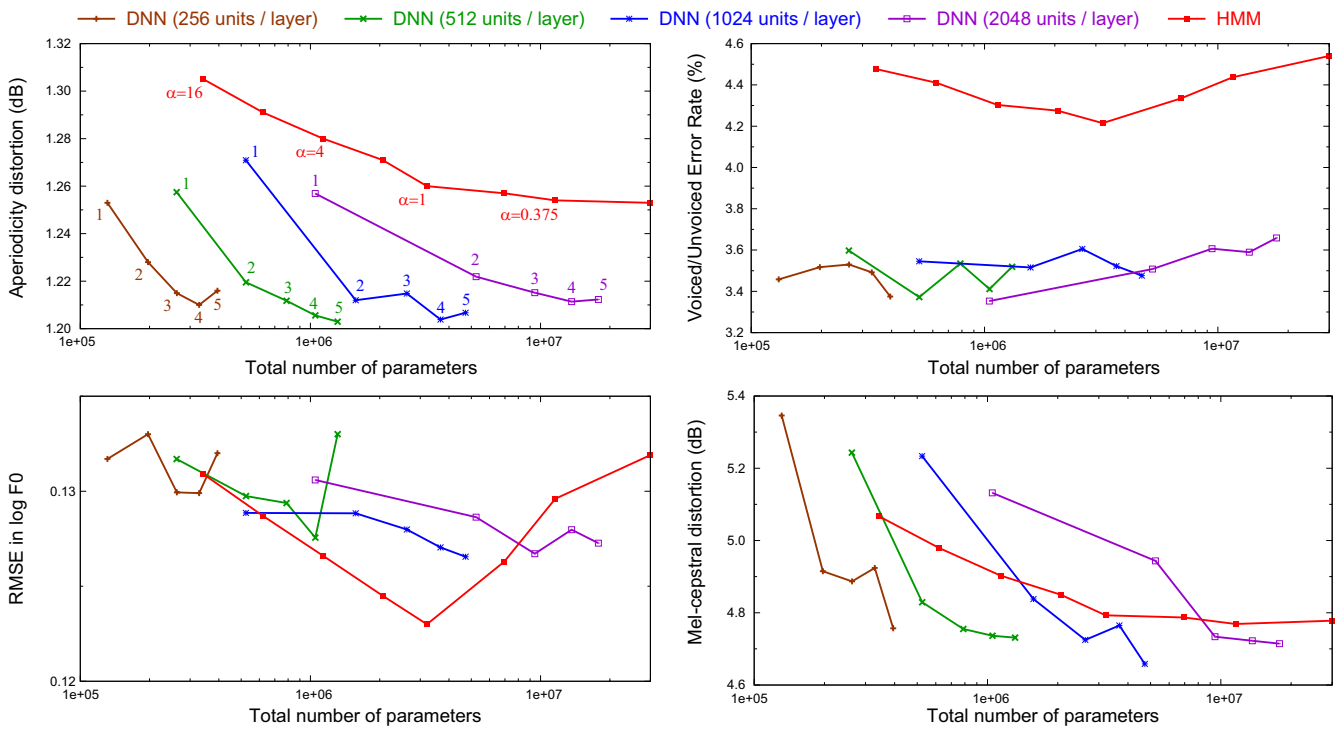

Fig. 3. Band aperioditicy distortions (dB), voiced/unvoiced error rates (%), root mean squared errors (RMSEs) in log F_0, and Mel-cepstral distortions (dB) of speech parameters predicted by the HMM-based and the DNN-based systems. Note that the numbers associated with the points on the lines plotting the DNN-based systems denote the numbers of layers.

4.3. Subjective evaluation

To compare the performance of the DNN-based systems with the HMM-based ones, a subjective preference listening test was conducted. The total number of test sentences was 173. One subject could evaluate a maximum of 30 pairs, they were randomly chosen from the test sentences for each subject. Each pair was evaluated by five subjects. The subjects used headphones. 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 this experiment, the HMM-based and DNN-based systems with similar numbers of parameters were compared. The DNN-based systems had four hidden layers with different number of units per layer (256, 512, or 1 024).

Table 1. Preference scores (%) between speech samples from the HMM and DNN-based systems. The systems which achieved significantly better preference at $p < 0.01$ level are in the bold font.

HMM (α)	DNN (#layers × #units)	Neutral	p value	z value
15.8 (16)	**38.5** (4 × 256)	45.7	$< 10^{-6}$	-9.9
16.1 (4)	**27.2** (4 × 512)	56.8	$< 10^{-6}$	-5.1
12.7 (1)	**36.6** (4 × 1 024)	50.7	$< 10^{-6}$	-11.5

Table 1 shows the experimental results. It can be seen from the table that the DNN-based systems were preferred significantly to the HMM-based ones in all three model sizes. The subjects reported that the DNN-based systems were less muffled. We expect that better prediction of Mel-cepstral coefficients by the DNN-based systems contributed to the preference.

5. CONCLUSIONS

This paper examined the use of the DNNs to perform speech synthesis. The DNN-based approach has a potential to address the limitations in the conventional decision tree-clustered context-dependent HMM-based approach, such as inefficiency in expressing complex context dependencies, fragmenting the training data, and completely ignoring linguistic input features which did not appear in the decision trees. The objective evaluation showed that the use of a deep architecture improved the performance of the neural network-based system for predicting spectral and excitation parameters. Furthermore, the DNN-based systems achieved better preference over the the HMM-based systems with a similar numbers of parameters in the subjective listening test. These experimental results showed the potential of the DNN-based approach for statistical parametric speech synthesis.

One of the advantages of the HMM-based system over the DNN-based one is the reduced computational cost. At synthesis time, the HMM-based systems traverse decision trees to find statistics at each state. On the other hand, the DNN-based system in this paper performs mapping from inputs to outputs which includes a number of arithmetic operations at each frame.[9] Future work includes the reduction of computations in the DNN-based systems, adding more input features including weak features such as emphasis, and exploring a better log F_0 modeling scheme.

[9]Switching to state or phoneme is also possible by changing the encoding scheme for time information.

6. REFERENCES

[1] 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.

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

[3] M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi, "Adaptation of pitch and spectrum for HMM-based speech synthesis using MLLR," in *Proc. ICASSP*, 2001, pp. 805–808.

[4] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Speaker interpolation in HMM-based speech synthesis system," in *Proc. Eurospeech*, 1997, pp. 2523–2526.

[5] K. Shichiri, A. Sawabe, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Eigenvoices for HMM-based speech synthesis," in *Proc. ICSLP*, 2002, pp. 1269–1272.

[6] T. Nose, J. Yamagishi, T. Masuko, and T. Kobayashi, "A style control technique for HMM-based expressive speech synthesis," *IEICE Trans. Inf. Syst.*, vol. E90-D, no. 9, pp. 1406–1413, 2007.

[7] Y. Morioka, S. Kataoka, H. Zen, Y. Nankaku, K. Tokuda, and T. Kitamura, "Miniaturization of HMM-based speech synthesis," in *Proc. Autumn Meeting of ASJ*, 2004, pp. 325–326, (in Japanese).

[8] S.-J. Kim, J.-J. Kim, and M.-S. Hahn, "HMM-based Korean speech synthesis system for hand-held devices," *IEEE Trans. Consum. Electron.*, vol. 52, no. 4, pp. 1384–1390, 2006.

[9] A. Gutkin, X. Gonzalvo, S. Breuer, and P. Taylor, "Quantized HMMs for low footprint text-to-speech synthesis," in *Proc. Interspeech*, 2010, pp. 837–840.

[10] J. Yamagishi, T. Nose, H. Zen, Z.-H. Ling, T. Toda, K. Tokuda, S. King, and S. Renals, "Robust speaker-adaptive HMM-based text-to-speech synthesis," *IEEE Trans. Audio Speech Lang. Process.*, vol. 17, no. 6, pp. 1208–1230, 2009.

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

[12] 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.

[13] J. Odell, *The use of context in large vocabulary speech recognition*, Ph.D. thesis, Cambridge University, 1995.

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

[15] W. Chou and W. Reichl, "Decision tree state tying based on penalized Bayesian information criterion," in *Proc. ICASSP*, 1999, vol. 1, pp. 345–348.

[16] T. Shinozaki, "HMM state clustering based on efficient cross-validation," in *Proc. ICASSP*, 2006, pp. 1157–1160.

[17] H. Zen and M.J.F. Gales, "Decision tree-based context clustering based on cross validation and hierarchical priors," in *Proc. ICASSP*, 2011, pp. 4560–4563.

[18] S. Esmeir and S. Markovitch, "Anytime learning of decision trees," *J. Mach. Learn. Res.*, vol. 8, pp. 891–933, 2007.

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

[20] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[21] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. Le, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Ng, "Large scale distributed deep networks," in *Proc. NIPS*, 2012.

[22] 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.

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

[24] O. Karaali, G. Corrigan, and I. Gerson, "Speech synthesis with neural networks," in *Proc. World Congress on Neural Networks*, 1996, pp. 45–50.

[25] Y. Qian, H. Liang, and F. Soong, "Generating natural F0 trajectory with additive trees," in *Proc. Interspeech*, 2008, pp. 2126–2129.

[26] Y. Nankaku, K. Nakamura, H. Zen, and K. Tokuda, "Acoustic modeling with contextual additive structure for HMM-based speech recognition," in *Proc. ICASSP*, 2008, pp. 4469–4472.

[27] K. Saino, *A clustering technique for factor analysis-based eigenvoice models*, Master thesis, Nagoya Institute of Technology, 2008, (in Japanese).

[28] 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.

[29] K. Yu, F. Mairesse, and S. Young, "Word-level emphasis modelling in HMM-based speech synthesis," in *Proc. ICASSP*, 2010, pp. 4238–4241.

[30] D. Yu and L. Deng, "Deep learning and its applications to signal and information processing," *IEEE Signal Process. Magazine*, vol. 28, no. 1, pp. 145–154, 2011.

[31] S. Furui, "Speaker independent isolated word recognition using dynamic features of speech spectrum," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 34, pp. 52–59, 1986.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

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

[40] 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.