Zhen-Hua Ling, Shi-Yin Kang, Heiga Zen, Andrew Senior, Mike Schuster, Xiao-Jun Qian, Helen Meng, and Li Deng

Deep Learning for Acoustic Modeling in Parametric Speech Generation

A systematic review of existing techniques and future trends

Hidden Markov models (HMMs) and Gaussian mixture models (GMMs) are the two most common types of acoustic models used in statistical parametric approaches for generating low-level speech waveforms from high-level symbolic inputs via intermediate acoustic feature sequences. However, these models have their limitations in representing complex, nonlinear relationships between the speech generation inputs and the acoustic features. Inspired by the intrinsically hierarchical process of human speech production and by the successful application of deep neural networks (DNNs) to automatic speech recognition (ASR), deep learning techniques have also been applied successfully to speech generation, as reported in recent literature. This article systematically reviews these emerging speech generation approaches, with the dual goal of helping readers gain a better understanding of the existing techniques as well as stimulating new work in the burgeoning area of deep learning for parametric speech generation.

In speech signal and information processing, many applications have been formulated as machine-learning tasks. ASR is a typical classification task that predicts word sequences from speech waveforms or feature sequences. There are also many regression tasks in speech processing that are aimed to generate speech signals from various types of inputs. They are referred to as *speech generation* tasks in this article. Speech generation covers a wide range of research topics in speech processing, such as text-to-speech (TTS) synthesis (generating speech from text), voice conversion (modifying nonlinguistic information of the input speech), speech enhancement (improving speech quality by noise reduction or other processing), and articulatory-to-acoustic mapping (converting articulatory movements to acoustic features). These

Digital Object Identifier 10.1109/MSP.2014.2359987
Date of publication: 6 April 2015

topics have the common goal of generating speech signals and differ in the forms of inputs. Statistical parametric speech generation (SPSG), which combines statistical acoustic models and vocoding techniques to generate speech waveforms, has been the mainstream approach for solving the speech generation problems. This approach first builds statistical acoustic models representing either the conditional probability density function (PDF) of output acoustic features given the input features or joint PDFs between the input and output features. The model structure is usually task dependent, but the parameters are estimated from a training database consisting of pairs of inputs and output acoustic features. At the speech-generation stage, the input features are given, which could be texts for TTS and noisy speech for speech enhancement. Then, the conditional distribution of the output acoustic features given the input features can be derived from the trained acoustic models. The output acoustic features are predicted from the conditional distribution under a certain criterion, e.g., maximizing the output probability, and are subsequently sent to a vocoder to reconstruct a speech waveform. In SPSG, vocoders are used to extract acoustic features, such as spectral [e.g., Mel-cepstral coefficients (MCCs)] and excitation (e.g., fundamental frequency and aperiodicity) features, from the raw waveforms of training data and to reconstruct speech waveforms from the generated acoustic features at synthesis time. Although both vocoder and acoustic modeling are essential for SPSG systems, this article focuses on acoustic modeling techniques for SPSG.

GMMs and HMMs with single Gaussian (or GMM) state-output PDFs are the two most popular acoustic models for SPSG [1], [2]. HMMs can represent nonstationary distributions of acoustic features using a sequence of hidden states, which are associated with linguistic features.

GMMs are widely used in frame-by-frame mapping for several speech-generation tasks, such as voice conversion, speech enhancement, and articulatory-to-acoustic mapping. The SPSG approaches using these two types of models have been shown to generate highly intelligible and smooth speech [2]–[4]. However, the generated speech sounds are noticeably muffled compared to recorded speech. Inadequate acoustic modeling is one of the main reasons for this deficiency [2], [5].

Take HMM-based speech synthesis, for example. In this approach, decision-tree-clustered, context-dependent phoneme HMMs are typically used to represent distributions of acoustic features given linguistic features [6]. The PDF of the acoustic features associated with each leaf node of the decision trees is typically a single Gaussian distribution with a diagonal covariance matrix.

At training time, parameters of the HMMs are usually estimated based on the maximum likelihood (ML) criterion. At synthesis time, given an input sentence and the trained parameters of the HMMs, the most likely acoustic features are predicted using the speech parameter-generation algorithm [7]. Since single Gaussian distributions are used as state-output PDFs, the outputs of the speech parameter-generation algorithm tend to distribute near the means of the Gaussian distributions, which are estimated by averaging all observations associated with a given decision tree leaf node. Although this averaging process improves the robustness of parameter estimation and generation, the detailed characteristics of the speech parameters are often lost. Therefore, the reconstructed spectral envelopes are typically oversmoothed, which leads to the muffled voice quality of the synthetic speech. In recent years, many techniques have been proposed to alleviate the oversmoothing problem by introducing better acoustic models (e.g., the trajectory HMM [8], product of experts [9], and Gaussian process regression [10]), improving the model training criterion (e.g., minimum generation error training [11], [12]), or modifying the speech parameter-generation algorithm (e.g., integrating a global variance model [13], using segment-wise representation [14], and minimizing Kullback–Leibler divergences [15]).

Since 2006, deep learning has emerged as a new area of machine-learning research [16], [17] and has also attracted the attention of many signal processing researchers. Deep learning refers to a class of machine-learning techniques that exploit many layers of nonlinear information processing for supervised or unsupervised feature extraction and transformation, and for pattern analysis and classification. Both unconditional deep architectures [e.g., restricted Boltzmann machines (RBMs) [19], deep belief networks (DBNs) [16], denoising autoencoders (DAEs) [20], [21], deep Boltzmann machines [18], and conditional deep architectures, e.g., DNNs] [17], have been intensively studied and explored by signal processing researchers in recent years. Strictly speaking, an RBM is a shallow graphical model with only one layer of hidden units; it is the constituent of many deep models (e.g., DBNs and DNNs). As a density model, RBMs perform much better than the conventional shallow structures (e.g., GMMs) [18]. Considering its intrinsic relationship and similarity to other deep models, RBMs are included as an example of deep generative models in this article.

One example is the successful application of DNNs to the acoustic modeling of ASR. In this approach, DNNs are introduced to replace GMMs for evaluating the fit between a frame of acoustic observations and each HMM state [22]. Deep learning techniques have also been applied to the acoustic modeling of speech generation very recently to deal with the limitations of the conventional approaches [23]–[40]. Different from the deep learning in ASR where DNN-HMM is the dominant model structure, these emerging acoustic modeling approaches for speech generation adopted various model structures. Some of them focus on improving the density functions of HMM states or GMM mixtures using RBMs or DBNs [23], [24], [27]. While some others use DBNs or DNNs to model the entire mapping process from input to output feature sequences directly [25], [26], [28]–[35].

This article first reviews the conventional and popular statistical framework for speech generation, including HMM-based speech synthesis and GMM-based voice conversion, focusing on acoustic modeling and not on the vocoder. It then analyzes the limitations of these approaches. The key models and techniques of deep learning as relevant to speech generation, including RBMs, DBNs, and DNNs, are also introduced.

Subsequently, emerging speech generation approaches using deep learning techniques for acoustic modeling are reviewed systematically, with an analysis of their motivations and a

description of their implementations. Finally, we discuss the remaining issues associated with current deep learning methods for parametric speech generation and point to future directions in this area.

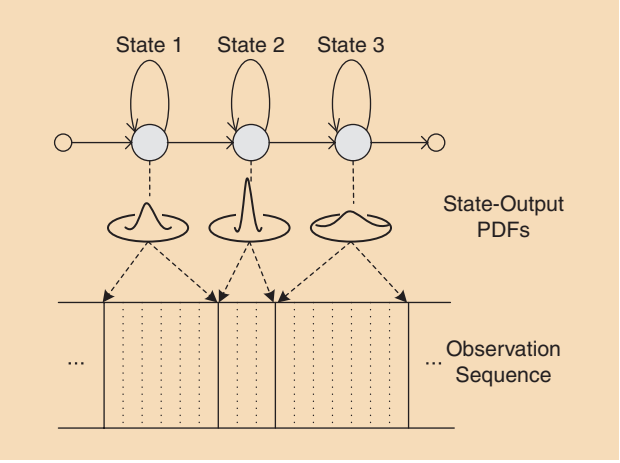
CONVENTIONAL ACOUSTIC MODELING USING HMMs AND GMMs FOR SPSG

HMM-BASED SPEECH SYNTHESIS

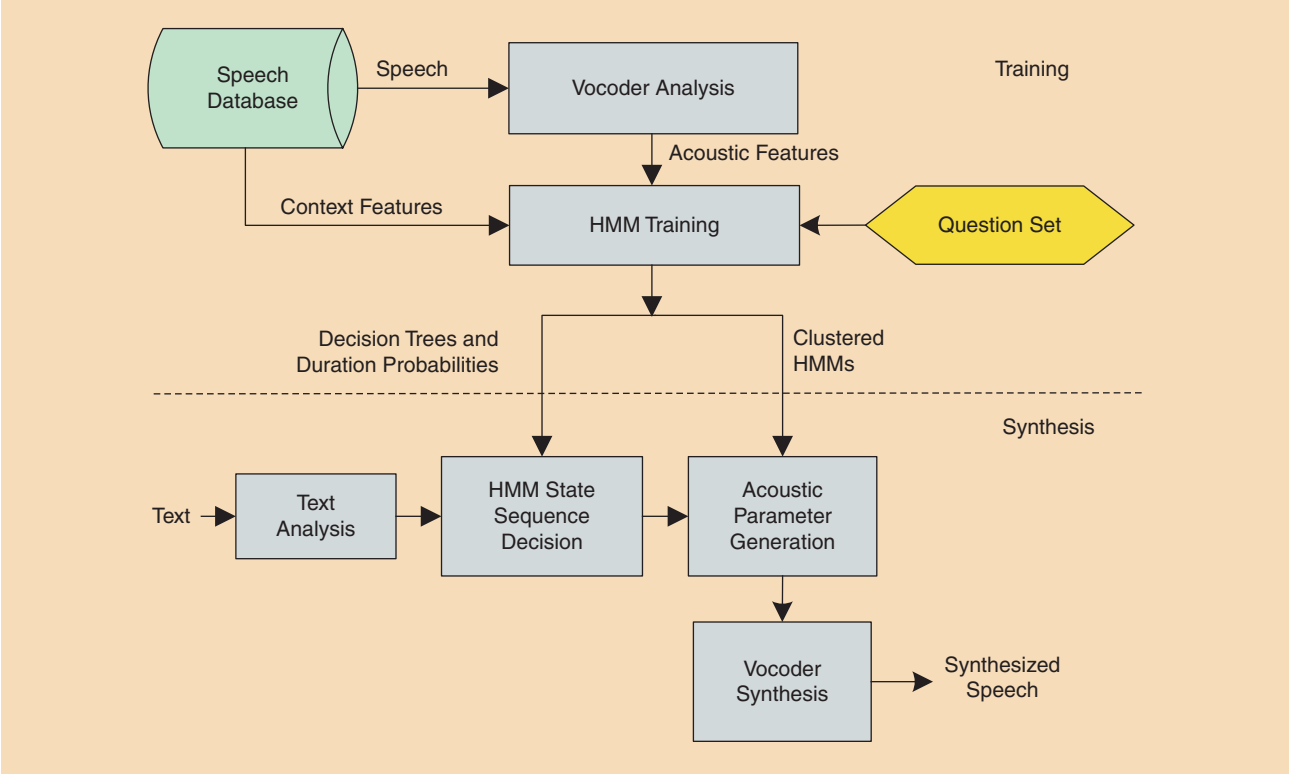
Statistical parametric speech synthesis (SPSS) [5] emerged in the mid-1990s [6], [41]. In this approach, the relationship between text and its acoustic realizations is modeled using a set of stochastic generative acoustic models. Decision-tree-clustered, context-dependent phoneme HMMs with single Gaussian state-output PDFs are the most popular generative acoustic model used in SPSS [6]. This approach is known as *HMM-based speech synthesis*. An HMM is a generative model that generates an observation sequence using a discrete and hidden state sequence. An example of a three-state left-to-right HMM is illustrated in Figure 1. In an HMM, state-output PDFs describe the distribution of observed features belonging to corresponding states and the transition among states is characterized by state-transition probabilities.

HMM-based speech synthesis is able to synthesize highly intelligible and smooth speech sounds. In addition, this model-based approach makes speech synthesis far more flexible compared to the conventional unit selection and waveform concatenation approach. Model adaptation, interpolation, and manipulation methods have been applied to control the HMM's parameters and thus diversify the characteristics of the generated speech [42]–[49]. Figure 2 shows the diagram of a typical HMM-based speech synthesis system. At the training stage, acoustic features of speech, including vocal tract and vocal source parameters, are extracted from the speech waveforms in a training database. Context features are also derived from the segmental and prosodic labels of the texts corresponding to the waveforms. Then, a set of parameters of context-dependent HMMs λ^* is estimated based on the ML criterion as

$$\lambda^* = \arg\max_{\lambda} p(\boldsymbol{y} \mid \boldsymbol{x}, \lambda), \tag{1}$$



[FIG1] **An example of a three-state, left-to-right HMM.**



[FIG2] **A block diagram of a typical HMM-based speech synthesis system.**

where $p(\cdot)$ is used to denote a PDF (continuous) in this article, $\boldsymbol{y} = [\boldsymbol{y}_1^\top, \boldsymbol{y}_2^\top, \ldots, \boldsymbol{y}_T^\top]^\top$ denotes a sequence of acoustic features with T frames, $\boldsymbol{y}_t$ is the acoustic feature at frame t, $\boldsymbol{x} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ is a sequence of linguistic context features for $\boldsymbol{y}$ that are derived from text automatically or annotated manually, N is the number of phonemes, and $(\cdot)^\top$ denotes the matrix transposition operation. The acoustic feature vector at each frame typically consists of static acoustic parameters $\boldsymbol{y}_{st} \in \mathcal{R}^{D_y}$ and their velocity and acceleration components, $\Delta \boldsymbol{y}_{st}$ and $\Delta^2 \boldsymbol{y}_{st}$, as

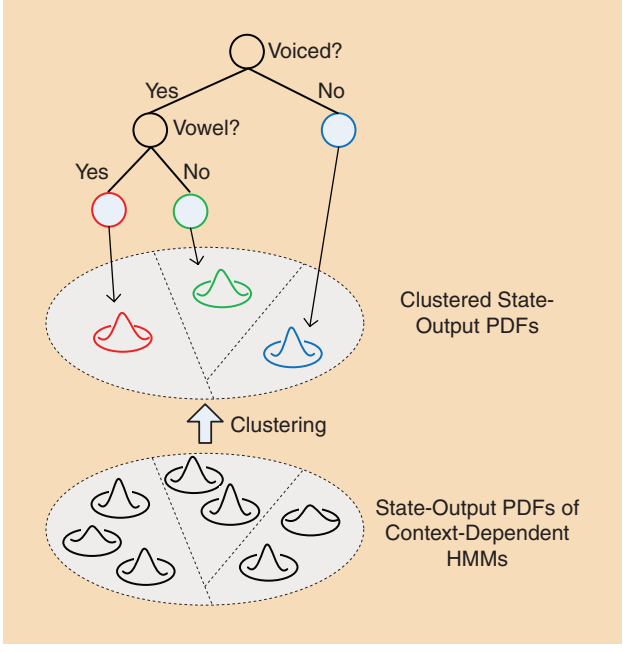
$$\boldsymbol{y}_t = [\boldsymbol{y}_{st}^\top, \Delta \boldsymbol{y}_{st}^\top, \Delta^2 \boldsymbol{y}_{st}^\top]^\top. \tag{2}$$

Therefore, the complete acoustic feature sequence $\boldsymbol{y}$ can be considered a linear transform of the static feature sequence $\boldsymbol{y}_s = [\boldsymbol{y}_{s_1}^\top, \boldsymbol{y}_{s_2}^\top, \ldots, \boldsymbol{y}_{s_T}^\top]^\top$ as

$$\boldsymbol{y} = \boldsymbol{M}_y \boldsymbol{y}_s, \tag{3}$$

where $\boldsymbol{M}_y$ is determined by the velocity and acceleration calculation functions used in (2) [7].

An HMM-based speech synthesis system typically contains a large number of context-dependent HMMs with linguistic context features that are far more extensive and can express far more fine-grained distinctions than those used in HMM-based ASR systems [50], [51]. This leads to data sparsity problems, such as overfitting in context-dependent models that have only few training examples available and the problem that many valid combinations of linguistic context features will be absent from the training database. To deal with this issue, a decision-tree-based clustering technique [52] is applied after the initial training to cluster state-output PDFs of the context-dependent HMMs as shown in Figure 3, where the state-output PDFs of the context-dependent HMMs with similar context descriptions are represented by a shared distribution. The question set for decision tree constriction is designed considering the characteristics of the language being processed. Next, the state alignment results using the trained HMMs are utilized to train context-dependent state-duration PDFs [6]. A single Gaussian distribution is also used to model the state-duration PDF at each state. A decision-tree-based model clustering technique is similarly applied to these state-duration PDFs [54]. Joint training of state-output and state-duration PDFs based on hidden semi-Markov models have also been used [53].



[FIG3] A decision-tree-based modeling clustering for HMM-based speech synthesis.

The acoustic model $p(\boldsymbol{y} \mid \boldsymbol{x}, \lambda)$ used in HMM-based speech synthesis can be rewritten as

$$p(\boldsymbol{y} \mid \boldsymbol{x}, \lambda) = \sum_{\forall \boldsymbol{q}} p(\boldsymbol{y}, \boldsymbol{q} \mid \boldsymbol{x}, \lambda), \tag{4}$$

$$= \sum_{\forall \boldsymbol{q}} P(\boldsymbol{q} \mid \boldsymbol{x}, \lambda) p(\boldsymbol{y} \mid \boldsymbol{q}, \lambda), \tag{5}$$

$$= \sum_{\forall \boldsymbol{q}} P(\boldsymbol{q} \mid \boldsymbol{x}, \lambda) \prod_{t=1}^{T} p(\boldsymbol{y}_t \mid q_t, \lambda), \tag{6}$$

where $P(\cdot)$ is used to denote a probability mass function (discrete) in this article, $p(\boldsymbol{y}_t \mid q_t, \lambda)$ is a state-output PDF associated with the q_tth state, which is typically a single Gaussian distribution with a diagonal covariance matrix and $\boldsymbol{q} = \{q_1, \ldots, q_T\}$ is an HMM state sequence. Note that the derivation from (5) to (6) is based on the assumption of HMMs that the frame observations are independent from each other given the state sequence.

To perform synthesis, the result of front-end linguistic analysis on input text is used to get the context features $\tilde{\boldsymbol{x}}$ for synthesis, as shown in Figure 2. In the HMM state sequence decision step, a sentence HMM corresponding to the input text is composed, with its parameters derived from the training stage.

In the step of acoustic parameter-generation, the acoustic features that maximize their output probabilities given the sentence HMM are determined under the constraints between static and dynamic features [7] as

$$\boldsymbol{y}_s^* = \arg\max_{\boldsymbol{y}_s} p(\boldsymbol{y} \mid \tilde{\boldsymbol{x}}, \lambda^*)\Big|_{\boldsymbol{y} = \boldsymbol{M}_y \boldsymbol{y}_s}. \tag{7}$$

The solution to (7) can be simplified if only the optimal state sequences in (5) is considered; optimization is approximated as two sequential steps

$$\boldsymbol{q}^* = \arg\max_{\boldsymbol{q}} P(\boldsymbol{q} \mid \tilde{\boldsymbol{x}}, \lambda^*), \tag{8}$$

$$\boldsymbol{y}_s^* = \arg\max_{\boldsymbol{y}_s} \prod_{t=1}^{T} p(\boldsymbol{y}_t \mid q_t^*, \lambda^*)\Big|_{\boldsymbol{y} = \boldsymbol{M}_y \boldsymbol{y}_s}. \tag{9}$$

Then, the closed-form solution of $\boldsymbol{y}_s^*$ can be derived by setting the partial derivative of (9) with respect to $\boldsymbol{y}_s$ to **zero** once the state sequence $\boldsymbol{q}^*$ is given [7]. Finally, these generated parameters are sent to a vocoder to reconstruct the speech waveforms.

GMM-BASED VOICE CONVERSION

The aim of voice conversion is to modify the nonlinguistic information (e.g., speaker characteristics) of input speech while

keeping the linguistic information unchanged. Different from the linguistic features, which are used as inputs for speech synthesis, the input features for voice conversion are typically continuous acoustic representations of a source voice. Many statistical approaches to voice conversion have been studied since the late 1980s, such as codebook mapping [55], GMM [2], [56], frequency warping [57], neural networks [58], partial least square regression [59], noisy channel model [60], etc. Among them, GMM-based voice conversion is the most popular [2], [56]. Figure 4 is a diagram of a typical GMM-based voice conversion system with parallel training data, which means that the training database contains the speech waveforms uttered by the source and target voices for the same texts. At the training stage, the acoustic features of the source and target speech in the training database are extracted by a vocoder and are aligned frame by frame by dynamic time warping. Then, the aligned pairs of the source acoustic feature vector $\boldsymbol{x}_t$ and the target acoustic feature vector $\boldsymbol{y}_t$ are concatenated to construct a joint feature vector $\boldsymbol{z}_t = [\boldsymbol{x}_t^\top, \boldsymbol{y}_t^\top]^\top$. Similar to HMM-based speech synthesis, the acoustic features $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ consist of static and dynamic components. Therefore, the acoustic feature sequences $\boldsymbol{x} = [\boldsymbol{x}_1^\top, \boldsymbol{x}_2^\top, \ldots, \boldsymbol{x}_T^\top]^\top$ and $\boldsymbol{y} = [\boldsymbol{y}_1^\top, \boldsymbol{y}_2^\top, \ldots, \boldsymbol{y}_T^\top]^\top$ can also be written as a linear transform from the static feature sequences $\boldsymbol{x}_s = [\boldsymbol{x}_{s_1}^\top, \boldsymbol{x}_{s_2}^\top, \ldots, \boldsymbol{x}_{s_T}^\top]^\top$ and $\boldsymbol{y}_s = [\boldsymbol{y}_{s_1}^\top, \boldsymbol{y}_{s_2}^\top, \ldots, \boldsymbol{y}_{s_T}^\top]^\top$ as $\boldsymbol{x} = \boldsymbol{M}_x \boldsymbol{x}_s$ and $\boldsymbol{y} = \boldsymbol{M}_y \boldsymbol{y}_s$, where $\boldsymbol{M}_x$ and $\boldsymbol{M}_y$ are determined by the velocity and acceleration calculation functions [2]. Then, a joint distribution GMM (JD-GMM) λ with a set of parameters $\{\alpha_m, \boldsymbol{\mu}_m^{(z)}, \Sigma_m^{(z)}\}_{m=1}^M$ is estimated to model a joint PDF between the source and target acoustic features, where M denotes the total number of mixture components in the JD-GMM, and α_m, $\boldsymbol{\mu}_m^{(z)}$, and $\Sigma_m^{(z)}$ correspond to the mixture weight, mean vector, and covariance matrix associated with the mth Gaussian component. The mean vector and covariance matrix are structured as

$$\boldsymbol{\mu}_m^{(z)} = \begin{bmatrix} \boldsymbol{\mu}_m^{(x)} \\ \boldsymbol{\mu}_m^{(y)} \end{bmatrix}, \Sigma_m^{(z)} = \begin{bmatrix} \Sigma_m^{(xx)} & \Sigma_m^{(xy)} \\ \Sigma_m^{(yx)} & \Sigma_m^{(yy)} \end{bmatrix}. \tag{10}$$

To reduce the number of model parameters and computational cost, $\Sigma_m^{(xx)}$, $\Sigma_m^{(yy)}$, $\Sigma_m^{(xy)}$, and $\Sigma_m^{(yx)}$ are commonly set to be diagonal [2]. These model parameters are typically estimated by the ML criterion as

$$\lambda^* = \arg\max_\lambda p(\boldsymbol{x}, \boldsymbol{y} \mid \lambda), \tag{11}$$

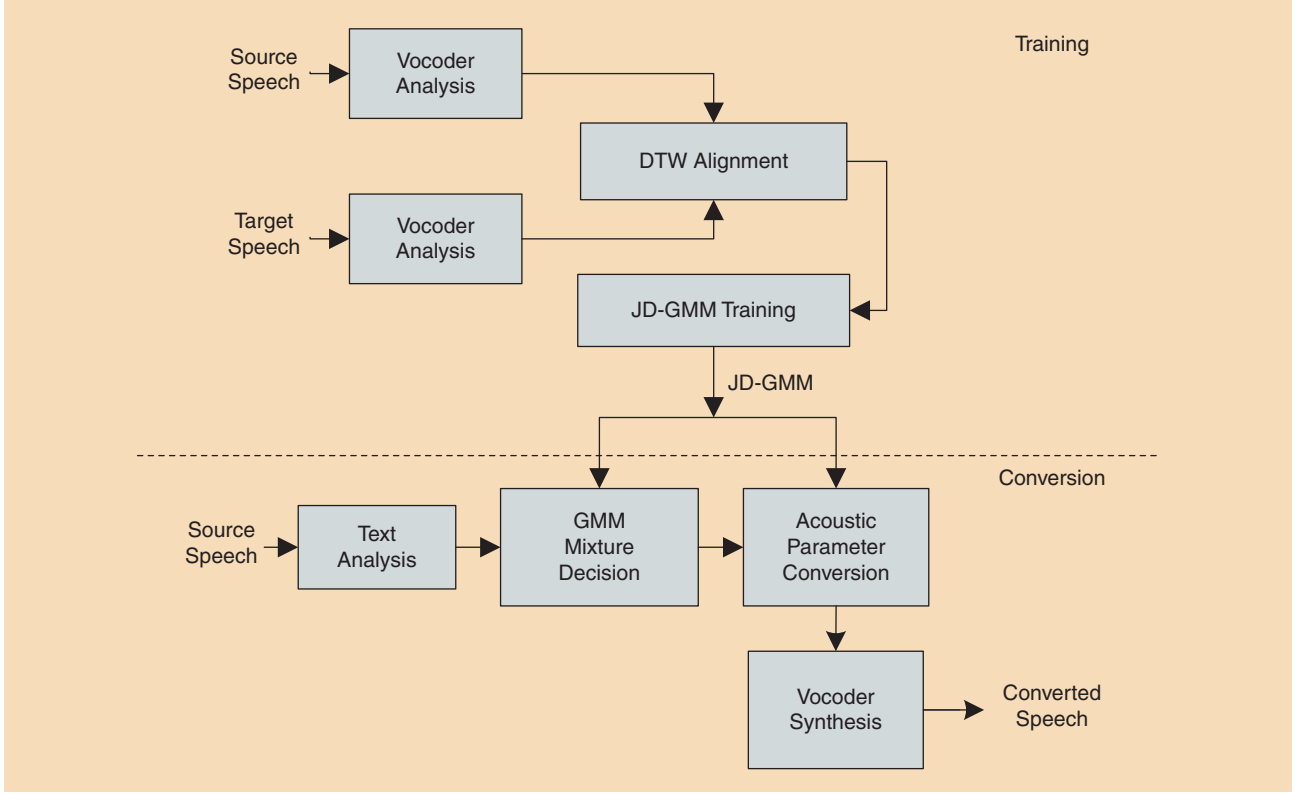
$$= \arg\max_\lambda \prod_{t=1}^T p(\boldsymbol{z}_t \mid \lambda). \tag{12}$$

The conditional PDF given an input source acoustic feature $\tilde{\boldsymbol{x}}$ can be further derived from the trained JD-GMM λ^* as

$$p(\boldsymbol{y} \mid \tilde{\boldsymbol{x}}, \lambda^*) = \sum_{\forall \boldsymbol{m}} p(\boldsymbol{y}, \boldsymbol{m} \mid \tilde{\boldsymbol{x}}, \lambda^*), \tag{13}$$

$$= \sum_{\forall \boldsymbol{m}} P(\boldsymbol{m} \mid \tilde{\boldsymbol{x}}, \lambda^*) \prod_{t=1}^T p(\boldsymbol{y}_t \mid \tilde{\boldsymbol{x}}_t, m_t, \lambda^*), \tag{14}$$

where $\boldsymbol{m} = \{m_1, \ldots, m_T\}$ denotes the sequence of mixture components. $P(\boldsymbol{m} \mid \tilde{\boldsymbol{x}}, \lambda^*) = \prod_{t=1}^T P(m_t \mid \tilde{\boldsymbol{x}}_t, \lambda^*)$ and $P(m_t \mid \tilde{\boldsymbol{x}}_t, \lambda^*)$



[FIG4] A block diagram of a typical GMM-based voice conversion system.

can be determined from the marginal PDF of $\boldsymbol{x}_t$, which is a GMM of M mixture components with the set of model parameters $\{\alpha_m, \boldsymbol{\mu}_m^{(x)}, \Sigma_m^{(x)}\}$. The conditional PDF $p(\boldsymbol{y}_t \mid \boldsymbol{x}_t, m_t, \lambda)$ is a Gaussian distribution with a mean vector

$$\boldsymbol{\mu}_{m,t}^{\boldsymbol{y}|\boldsymbol{x}} = \boldsymbol{\mu}_m^{(y)} + \Sigma_m^{(yx)} \Sigma_m^{(xx)-1} (\boldsymbol{x}_t - \boldsymbol{\mu}_m^{(x)}) \tag{15}$$

and a covariance matrix

$$\Sigma_m^{\boldsymbol{y}|\boldsymbol{x}} = \Sigma_m^{(yy)} - \Sigma_m^{(yx)} \Sigma_m^{(xx)-1} \Sigma_m^{(xy)}. \tag{16}$$

Figure 5(a) shows the PDF of an example JD-GMM with two mixtures, where the source and target acoustic features are simply represented by scalars. Two examples of the conditional distributions derived from the JD-GMM are illustrated in Figure 5(b), which are also two-mixture GMMs.

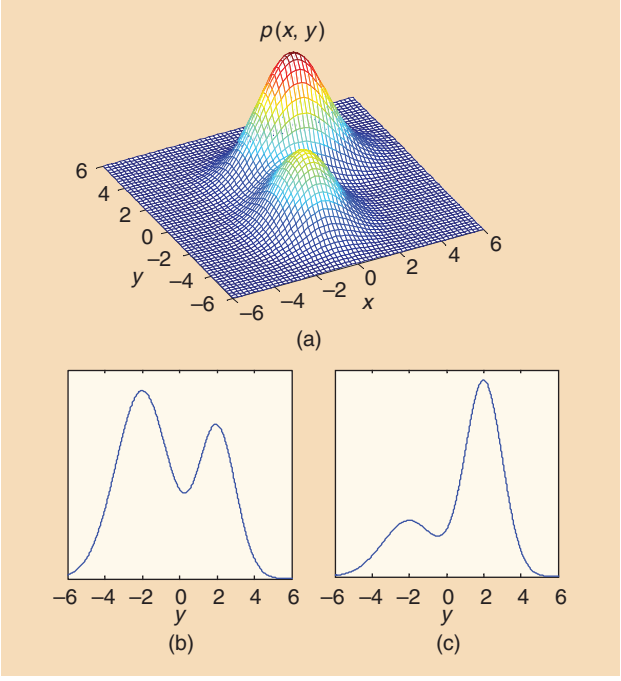
At conversion time, the converted acoustic features can be predicted using either the minimum mean-square error [56] or the maximum a posteriori criterion [2], given the source acoustic feature sequence $\tilde{\boldsymbol{x}}$. If the maximum a posteriori criterion is adopted, the static acoustic features of the target voice are predicted as

$$\boldsymbol{y}_s^* = \arg\max_{\boldsymbol{y}_s} p(\boldsymbol{y} \mid \tilde{\boldsymbol{x}}, \lambda^*)\Big|_{\boldsymbol{y}=M_y \boldsymbol{y}_s}. \tag{17}$$

Similar to HMM-based speech synthesis, the solution to (17) is simplified by only considering the mixture components with the highest posterior probability at each frame in (14). Thus, we have

$$m_t^* = \arg\max_{m_t} P(m_t \mid \tilde{\boldsymbol{x}}_t, \lambda^*), \tag{18}$$

$$\boldsymbol{y}_s^* = \arg\max_{\boldsymbol{y}_s} \prod_{t=1}^{T} p(\boldsymbol{y}_t \mid \tilde{\boldsymbol{x}}_t, m_t^*, \lambda^*)\Big|_{\boldsymbol{y}=M_y \boldsymbol{y}_s}. \tag{19}$$



[FIG5] PDFs of (a) a joint distribution GMM $p(x, y)$ with two mixtures and (b) and (c) the conditional distributions $p(y \mid x)$ derived from it. (b) $p(y \mid x = -1)$. (c) $p(y \mid x = 1)$.

Then, a closed-form solution to (19) can be achieved in a similar way to solve (9) [2]. Finally, the converted acoustic features are sent to a vocoder to reconstruct the corresponding speech waveform.

This GMM-based voice conversion framework has also been successfully applied to other frame-by-frame-mapping speech generation tasks, such as bandwidth extension [61], speech enhancement [62], [63], and articulatory-acoustic mapping [64].

THE COMMON STRUCTURE: TWO-STEP MAPPING

As shown in (8), (9), (18), and (19), both HMM- and GMM-based SPSG share the common structure of two-step mapping to represent the conditional PDF of the acoustic features $\boldsymbol{y}$, given the input features $\boldsymbol{x}$.

1) *Input-to-cluster mapping using hidden discrete variable*: In this step, each input feature vector is mapped to hidden discrete clusters of the acoustic features to be generated, i.e., the HMM state q_t^* in (8) or the GMM mixture component m_t^* in (18). In HMM-based speech synthesis, $\boldsymbol{q}^*$ is determined using the decision trees for state-output PDFs and the state-duration PDFs. In GMM-based voice conversion, this is achieved by the posterior probabilities $P(m_t \mid \tilde{\boldsymbol{x}}_t, \lambda^*)$.

2) *Cluster-to-feature mapping using Gaussian distributions*: Given the input features, once the cluster sequence is determined, the conditional PDF for generating the acoustic features can be determined by combining the PDFs describing each cluster in the sequence, i.e., $p(\boldsymbol{y}_t \mid q_t^*, \lambda^*)$ in (9) and $p(\boldsymbol{y}_t \mid \tilde{\boldsymbol{x}}_t, m_t^*, \lambda^*)$ in (19). In the current SPSG approaches, the PDF associated with each cluster is typically an ML-estimated single Gaussian distribution with a diagonal covariance matrix [2], [6].

Although the acoustic modeling approach described earlier works reasonably well in SPSG, it has well known limitations. First, decision-tree-based input-to-cluster mapping in HMM-based speech synthesis is inefficient for expressing complex context dependencies, such as the exclusive OR (XOR) problem. This may lead to overfitting to the training data because of the data partitioning issue [65]. Second, the cluster-to-feature mapping using single Gaussian distributions with diagonal covariance matrices is established based on two independence assumptions: 1) conditional independence between frames given the state or the Gaussian component and 2) independence of acoustic features within a frame. As discussed earlier, this leads to reconstructed spectral envelopes being oversmoothed and the quality of synthetic speech is degraded.

Compared with the statistical models used in the conventional acoustic modeling of SPSG (such as decision trees, HMMs, and GMMs), deep learning techniques are better at representing the intrinsic correlations among the units of input vectors (e.g., the input context features for speech synthesis), among the units of output vectors (e.g., the output spectral features for speech synthesis), and between the input and output vectors (e.g., the aligned spectral features of the source and target speakers for voice conversion) using a joint (e.g., RBM and DBN) or conditional (e.g., DNN) modeling framework. Therefore, it is promising that the deep learning techniques can help the acoustic modeling of speech generation to

overcome the limitations of the current approach mentioned earlier, so as to achieve better input-to-cluster or/and cluster-to-feature mapping. Furthermore, human speech production mechanisms involve clearly layered hierarchical structures in transforming the information from the linguistic level to the acoustic level via intermediate levels of motor control and articulation [66]–[69], also suggesting the need for deep model structures for SPSG applications.

This article reviews a number of recent approaches, based on the deep learning techniques, for overcoming these limitations and improving acoustic modeling for SPSG. A few basic models for deep learning are first reviewed in the section "Basic Models for Deep Learning," including some mathematical details that are uncommon in the literature but essential for using these models in SPSG.

BASIC MODELS FOR DEEP LEARNING

Since 2010, deep learning techniques have been successfully applied to the modeling of speech signals, such as speech recognition [70]–[74], spectrogram coding [20], voice activity detection [75], and acoustic-articulatory inversion mapping [76]. One significant advantage of deep learning techniques is their strong ability to represent the intrinsic correlation or mapping relationship among the units of a high-dimensional stochastic vector using a joint (e.g., RBM and DBN) or conditional (e.g., CRBM and DNN) modeling framework. Considering that speech generation is a regression task and the aim of its acoustic modeling is to describe the joint or conditional distribution of continuous acoustic features, we will review these basic models from the viewpoint of density models in this section.

RBMs

An RBM is an undirected graphical model (i.e., a Markov random field) that can model the dependency among a set of random variables using a two-layered architecture [19]. In an RBM, visible stochastic units $\boldsymbol{v} = [v_1, \ldots, v_V]^\top$ are connected to hidden stochastic units $\boldsymbol{h} = [h_1, \ldots, h_H]^\top$, as shown in Figure 6, where V and H are the numbers of units at the visible and hidden layers, respectively. When $\boldsymbol{v} \in \{0,1\}^V$ and $\boldsymbol{h} \in \{0,1\}^H$ are both binary stochastic variables, the energy function of the state $\{\boldsymbol{v}, \boldsymbol{h}\}$ is defined as

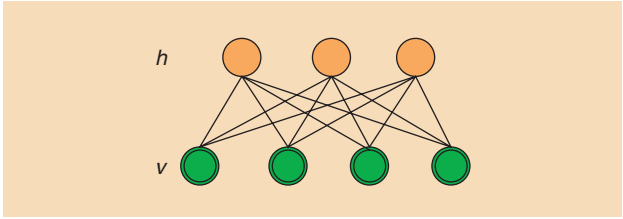
$$E(\boldsymbol{v}, \boldsymbol{h}; \lambda) = -\sum_{i=1}^{V} a_i v_i - \sum_{j=1}^{H} b_j h_j - \sum_{i=1}^{V}\sum_{j=1}^{H} w_{\text{ij}} v_i h_j, \tag{20}$$

where w_{ij} represents the symmetric interaction between v_i and h_j, a_i and b_j are bias terms, and λ denotes the set of model parameters consisting of $\boldsymbol{a} = [a_1, \ldots, a_V]^\top$, $\boldsymbol{b} = [b_1, \ldots, b_H]^\top$, and $\boldsymbol{W} = \{w_{\text{ij}}\} \in \mathcal{R}^{V \times H}$. The joint PDF over the visible and hidden units is given by a Boltzmann distribution as

$$P(\boldsymbol{v}, \boldsymbol{h} \mid \lambda) = \frac{1}{\mathcal{Z}_\lambda} \exp\{-E(\boldsymbol{v}, \boldsymbol{h}; \lambda)/C_T\}, \tag{21}$$

where C_T is a temperature parameter, which is assumed to be 1 in the rest of this article, and

$$\mathcal{Z}_\lambda = \sum_{\forall \boldsymbol{v}} \sum_{\forall \boldsymbol{h}} \exp\{-E(\boldsymbol{v}, \boldsymbol{h}; \lambda)\} \tag{22}$$



[FIG6] A graphical model representation for an RBM.

is the partition function, which can be estimated using the annealed importance sampling (AIS) technique [18]. The marginal PDF over the visible vector $\boldsymbol{v}$ can be calculated as

$$P(\boldsymbol{v} \mid \lambda) = \frac{1}{\mathcal{Z}_\lambda} \sum_{\forall \boldsymbol{h}} \exp\{-E(\boldsymbol{v}, \boldsymbol{h}; \lambda)\}. \tag{23}$$

Given a training set, λ can be estimated based on the ML criterion by stochastic gradient descent. The derivative of $\log P(\boldsymbol{v} \mid \lambda)$ with respect to the model parameters, e.g., w_{ij}, can be derived using (20)–(23) as

$$\frac{\partial \log P(\boldsymbol{v} \mid \lambda)}{\partial w_{\text{ij}}} = \mathrm{E}_{P_{\text{Data}}}[v_i h_j] - \mathrm{E}_{P_{\text{Model}}}[v_i h_j], \tag{24}$$

where $\mathrm{E}_{P_{\text{Data}}}[\cdot]$ denotes an expectation with respect to the distribution of the training data and $\mathrm{E}_{P_{\text{Model}}}[\cdot]$ denotes an expectation with respect to the distribution of the model $P(\boldsymbol{v} \mid \lambda)$. Because computation of $\mathrm{E}_{P_{\text{Model}}}[\cdot]$ is intractable, the contrastive divergence (CD) algorithm has been proposed to approximate $\mathrm{E}_{P_{\text{Model}}}[\cdot]$ by Gibbs sampling [77].

RBMs can also be applied to model the distribution of real-valued data (e.g., mel-frequency MCCs in ASR), categorical data (e.g., some linguistic context features in TTS), or a mixed vector of binary, real-valued, and categorical data by defining different forms of energy functions [25]. For a Gaussian-Bernoulli RBM, which means $\boldsymbol{v} \in \mathcal{R}^V$ are real-valued and $\boldsymbol{h} \in \{0,1\}^H$ are binary, the energy is defined as

$$E(\boldsymbol{v}, \boldsymbol{h}; \lambda) = \sum_{i=1}^{V} \frac{(v_i - a_i)^2}{2\sigma_i^2} - \sum_{j=1}^{H} b_j h_j - \sum_{i=1}^{V}\sum_{j=1}^{H} w_{\text{ij}} h_j \frac{v_i}{\sigma_i}, \tag{25}$$

where the variance parameters σ_i^2 are commonly fixed to a predetermined value instead of learning them from training data [17]. While training a Gaussian–Bernoulli RBM using the CD algorithm, the two conditional PDFs for Gibbs sampling are derived as

$$P(h_j = 1 \mid \boldsymbol{v}, \lambda) = g\left(b_j + \boldsymbol{v}^\top \Sigma^{-\frac{1}{2}} \boldsymbol{w}_{\cdot j}\right), \tag{26}$$

$$p(\boldsymbol{v} \mid \boldsymbol{h}, \lambda) = \mathcal{N}(\boldsymbol{v}; \boldsymbol{W}\boldsymbol{h} + \boldsymbol{a}, \Sigma), \tag{27}$$

where $g(x) = 1/(1 + \exp(-x))$ is a sigmoid function, $\boldsymbol{w}_{\cdot j}$ denotes the jth column of a matrix $\boldsymbol{W}$, $\mathcal{N}(\boldsymbol{v}; \boldsymbol{\mu}, \Sigma)$ denotes a Gaussian distribution of $\boldsymbol{v}$ with a mean vector $\boldsymbol{\mu}$ and a covariance matrix Σ, and $\Sigma = \mathrm{diag}\{\sigma_1^2, \ldots, \sigma_V^2\}$ is diagonal. If $\{\sigma_i^2\}_{i=1}^V$ are fixed to 1, Σ turns into an identity matrix.

RBMs have been successfully used in unsupervised pretraining of DNN-based acoustic models in ASR [22]. RBMs have also been used as density models to represent the distributions of acoustic

features for SPSG [23], [24], [27]. The marginal PDF of a Gaussian–Bernoulli RBM can be derived from (23) and (25) as [the variance parameters σ_i^2 in (25) are fixed to 1 for notational simplicity]

$$\begin{aligned} p(\boldsymbol{v} \mid \lambda) &= \frac{1}{\mathcal{Z}_\lambda} \sum_{\forall \boldsymbol{h}} \exp\{-E(\boldsymbol{v}, \boldsymbol{h}; \lambda)\} \\ &= \frac{1}{\mathcal{Z}_\lambda} \sum_{\forall \boldsymbol{h}} \exp\left\{-\sum_{i=1}^{V} \frac{(v_i - a_i)^2}{2} + \boldsymbol{b}^\top \boldsymbol{h} + \boldsymbol{v}^\top \boldsymbol{W} \boldsymbol{h}\right\} \\ &= \frac{1}{\mathcal{Z}_\lambda} \exp\left\{-\sum_{i=1}^{V} \frac{(v_i - a_i)^2}{2}\right\} \\ &\quad \cdot \prod_{j=1}^{H} \sum_{h_j \in \{0,1\}} \exp(b_j h_j + \boldsymbol{v}^\top \boldsymbol{w}_{\cdot j} h_j) \\ &= \frac{1}{\mathcal{Z}_\lambda} \prod_{i=1}^{V} \exp\left\{-\frac{(v_i - a_i)^2}{2}\right\} \\ &\quad \cdot \prod_{j=1}^{H} \{1 + \exp(b_j + \boldsymbol{v}^\top \boldsymbol{w}_{\cdot j})\}, \end{aligned} \tag{28}$$

which shows that a Gaussian–Bernoulli RBM can be considered either a product of experts (PoEs) or a GMM.

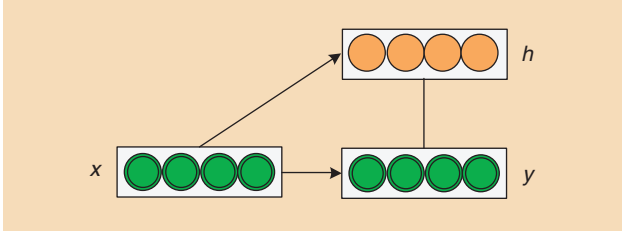
- *PoE* [78]: A PoE represents a probability distribution by multiplying several simpler distributions, followed by normalization. PoEs can produce much sharper distributions than their individual experts and perform more efficiently than mixture models in high-dimensional space [77]. As shown in (28), elements in the first product represent single-variable experts without cross-dimensional correlations. The elements in the second product represent constraints between input variable using the model parameters corresponding to each hidden unit.
- *GMM*: An RBM can also be considered as a GMM with 2^H mixture components with structured mean vectors and identity covariance matrices. For example, if $H = 0$,

$$p(\boldsymbol{v} \mid \lambda) = \frac{1}{\mathcal{Z}_\lambda} \exp\left\{-\sum_{i=1}^{V} \frac{(v_i - a_i)^2}{2}\right\} \tag{29}$$

is a single Gaussian distribution with a mean vector $\boldsymbol{a}$. If H is increased to 1, $p(\boldsymbol{v} \mid \lambda)$ in (28) can be rewritten as

$$\begin{aligned} p(\boldsymbol{v} \mid \lambda) &= \frac{1}{\mathcal{Z}_\lambda} \exp\left\{-\sum_{i=1}^{V} \frac{(v_i - a_i)^2}{2}\right\} \\ &\quad + \frac{\kappa}{\mathcal{Z}_\lambda} \exp\left\{-\sum_{i=1}^{V} \frac{(v_i - a_i - w_{i1})^2}{2}\right\}, \end{aligned} \tag{30}$$

where κ is a constant value determined by the model parameters. We can see that $p(\boldsymbol{v} \mid \lambda)$ becomes a GMM with two mixture components, where their mean vectors become $\boldsymbol{a}$ and $\boldsymbol{a} + \boldsymbol{w}_{\cdot 1}$, respectively. Generally speaking, as the number of hidden units is incremented, the number of mixture components is doubled by copying and shifting the mean vectors. These structured mean vectors and the tied covariance matrices provide better generalization. Thus, they are robust toward data sparsity.



[FIG7] The graphical model representation for a CRBM.

RBMs can also be used to model conditional PDFs between two groups of visible units using their variation form, i.e., the conditional RBM (CRBM). The CRBM was originally proposed to model the temporal dependency of human motion features [79]. The model structure of a CRBM representing the conditional PDF $p(\boldsymbol{y} \mid \boldsymbol{x}, \lambda)$ is illustrated in Figure 7. In this model, the links between the visible units $\boldsymbol{y}$ and the hidden units $\boldsymbol{h}$ are undirected. If $\boldsymbol{x}$ is known, $\boldsymbol{y}$ and $\boldsymbol{h}$ form an RBM and its model parameters depend on $\boldsymbol{x}$ through the two directed links from $\boldsymbol{x}$ to $\boldsymbol{y}$ and $\boldsymbol{h}$. If $\boldsymbol{h} \in \{0,1\}^H$ are binary and $\boldsymbol{x} \in \mathcal{R}^{D_X}$ and $\boldsymbol{y} \in \mathcal{R}^{D_Y}$ are real-valued, the energy function of a CRBM can be written as

$$\begin{aligned} E(\boldsymbol{y}, \boldsymbol{h}, \boldsymbol{x}; \lambda) &= \sum_{i=1}^{D_Y} \frac{\left(y_i - a_i - \sum_k A_{ki} x_k\right)^2}{2\sigma_i^2} \\ &\quad - \sum_{j=1}^{H} \left(b_j + \sum_k B_{kj} x_k\right) h_j - \sum_{i=1}^{D_Y} \sum_{j=1}^{H} w_{ij} h_j \frac{y_i}{\sigma_i}, \end{aligned} \tag{31}$$

where $\lambda = \{\boldsymbol{A}, \boldsymbol{B}\}$ is the set of parameters in the CRBM, $\boldsymbol{A} = \{A_{ki}\} \in \mathcal{R}^{D_X \times V}$ and $\boldsymbol{B} = \{B_{kj}\} \in \mathcal{R}^{D_Y \times H}$ are matrices corresponding to the directed links in Figure 7. The conditional PDF of $\boldsymbol{y}$ given $\boldsymbol{x}$ can be written as

$$p(\boldsymbol{y} \mid \boldsymbol{x}, \lambda) = \sum_{\forall \boldsymbol{h}} p(\boldsymbol{y}, \boldsymbol{h} \mid \boldsymbol{x}, \lambda) \tag{32}$$

$$= \frac{1}{\mathcal{Z}_\lambda} \sum_{\forall \boldsymbol{h}} \exp\{-E(\boldsymbol{y}, \boldsymbol{h}, \boldsymbol{x}; \lambda)\}, \tag{33}$$

where

$$p(\boldsymbol{y}, \boldsymbol{h} \mid \boldsymbol{x}, \lambda) = \frac{1}{\mathcal{Z}_\lambda} \exp\{-E(\boldsymbol{y}, \boldsymbol{h}, \boldsymbol{x}; \lambda)\}, \tag{34}$$

$$\mathcal{Z}_\lambda = \int \sum_{\forall \boldsymbol{h}} \exp\{-E(\boldsymbol{y}, \boldsymbol{h}, \boldsymbol{x}; \lambda)\} d\boldsymbol{y}. \tag{35}$$

Similar to RBMs, λ can be trained based on the ML criterion using the CD algorithm [79].

DBNs

A DBN is a probabilistic generative model that is composed of many layers of hidden units [16]. The graphical model representation for a three-hidden-layer DBN is shown in Figure 8. In this model, each layer captures the correlations among the activities of hidden features in the layer below. The top two layers of the DBN form an undirected graph. The lower layers form a directed graph with a top–down direction to generate the visible units. Assuming that $\boldsymbol{v}$ is real-valued and $\{\boldsymbol{h}^{(l)}\}_{l=1}^{L}$ are binary, the joint PDF of a DBN over the visible and hidden units can be written as

$$\begin{aligned} p(\boldsymbol{v}, \boldsymbol{h}^{(1)}, \ldots, \boldsymbol{h}^{(L)} \mid \lambda) &= p(\boldsymbol{v} \mid \boldsymbol{h}^{(1)}, \lambda) \prod_{l=2}^{L-1} P(\boldsymbol{h}^{(l-1)} \mid \boldsymbol{h}^{(l)}, \lambda) \\ &\quad \cdot P(\boldsymbol{h}^{(L-1)}, \boldsymbol{h}^{(L)} \mid \lambda), \end{aligned} \tag{36}$$

where $\boldsymbol{h}^{(l)} = [h_1^{(l)}, \ldots, h_{H_l}^{(l)}]^\top$ is the hidden stochastic vector at the lth hidden layer, H_l is the dimensionality of $\boldsymbol{h}^{(l)}$, and L is the number of hidden layers. $P(\boldsymbol{h}^{(L-1)}, \boldsymbol{h}^{(L)} \mid \lambda)$ is represented by an RBM as (21) with the weight matrix $\boldsymbol{W}^{(L)}$ and the bias vectors $\boldsymbol{a}^{(L)}$ and $\boldsymbol{b}^{(L)}$. $p(\boldsymbol{v} \mid \boldsymbol{h}^{(1)}, \lambda)$ and $\{P(\boldsymbol{h}^{(l-1)} \mid \boldsymbol{h}^{(l)}, \lambda)\}_{l=2}^{L-1}$ are represented by sigmoid belief networks [80]. Each sigmoid belief network is described by a weight matrix $\boldsymbol{W}^{(l)}$ and a bias vector $\boldsymbol{a}^{(l)}$. Assuming that $\boldsymbol{v}$ is real-valued and $\{\boldsymbol{h}^{(l)}\}_{l=2}^{L}$ are binary, the conditional PDF $p(\boldsymbol{v} \mid \boldsymbol{h}^{(1)}, \lambda)$ of a sigmoid belief network is described by (27). For $l \in \{2, 3, \ldots, L-1\}$, the dependency between two adjacent hidden layers is represented by

$$P(h_i^{(l-1)} = 1 \mid \boldsymbol{h}^{(l)}, \lambda) = g\left(a_i^{(l)} + \sum_j w_{ij}^{(l)} h_j^{(l)}\right). \tag{37}$$

For an L-hidden-layer DBN, its model parameters are composed of $\{\boldsymbol{a}^{(1)}, \boldsymbol{W}^{(1)}, \ldots, \boldsymbol{a}^{(L-1)}, \boldsymbol{W}^{(L-1)}, \boldsymbol{a}^{(L)}, \boldsymbol{b}^{(L)}, \boldsymbol{W}^{(L)}\}$. Furthermore, the marginal PDF of the visible variables for a DBN can be written as

$$p(\boldsymbol{v} \mid \lambda) = \sum_{\forall \boldsymbol{h}^{(l)}} \cdots \sum_{\forall \boldsymbol{h}^{(L)}} p(\boldsymbol{v}, \boldsymbol{h}^{(1)}, \ldots, \boldsymbol{h}^{(L)} \mid \lambda). \tag{38}$$

Given the training samples of the visible units, it is difficult to estimate the model parameters of a DBN directly based on the ML criterion due to the complex model structure with multiple hidden layers. Therefore, a greedy learning algorithm has been proposed and popularly applied to train DBNs in a layer-by-layer manner [16]. A stack of RBMs are used in this algorithm. First, it estimates the parameters $\{\boldsymbol{a}^{(1)}, \boldsymbol{b}^{(1)}, \boldsymbol{W}^{(1)}\}$ of the first-layer RBM to model the visible training data. Then, it freezes the parameters $\{\boldsymbol{a}^{(1)}, \boldsymbol{W}^{(1)}\}$ of the first layer and draws samples from $P(\boldsymbol{h}^{(1)} = 1 \mid \boldsymbol{v}, \lambda)$ using (26) to train the next-layer RBM $\{\boldsymbol{a}^{(2)}, \boldsymbol{b}^{(2)}, \boldsymbol{W}^{(2)}\}$. This training procedure is conducted recursively until it reaches the top layer and gets $\{\boldsymbol{a}^{(L)}, \boldsymbol{b}^{(L)}, \boldsymbol{W}^{(L)}\}$. It has been shown that this greedy learning algorithm can improve the lower bound on the log-likelihood of the model, given training samples by adding each new hidden layer [16], [18]. Once the model parameters are estimated, the calculation of the log probability that a DBN assigns to training or test data by applying (38) directly becomes computationally intractable. A lower bound on the log probability can be estimated by combining the AIS-based partition function estimation with the approximate inference [18].

DNNs

A DNN is a feed-forward, artificial neural network that has more than one layer of hidden units between its input and output layers [22]. The model representation for a two-hidden-layer DNN is shown in Figure 9. At each hidden layer, each hidden unit typically maps the weighted sum of its inputs from the layer below to a deterministic value using a nonlinear activation function and passes it to the layer above. If a sigmoid function $g(\cdot)$ is used as an activation function, its output is given as

$$h_j^{(l)} = g\left(b_j^{(l)} + \sum_i h_i^{(l-1)} w_{ij}^{(l)}\right), \tag{39}$$

where $h_j^{(l)}$ is the jth hidden unit at the lth layer ($h_i^{(0)} = x_i$ is the ith dimension of input feature), $b_j^{(l)}$ is the bias of the jth unit at the lth layer, and $w_{ij}^{(l)}$ is the weight associated with the link from $h_i^{(l-1)}$ to $h_j^{(l)}$. The form of activation functions at the output layer depends on the task. For multiclass classification tasks, a softmax function is typically used

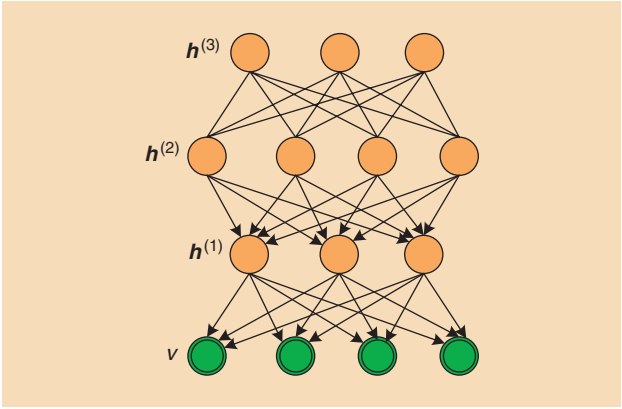
$$\tilde{y}_j = \frac{\exp\left\{b_j^{(L+1)} + \sum_i h_i^{(L)} w_{ij}^{(L+1)}\right\}}{\sum_k \exp\left\{b_k^{(L+1)} + \sum_i h_i^{(L)} w_{ik}^{(L+1)}\right\}}, \tag{40}$$

where $\tilde{y}_j = h_j^{(L+1)}$ gives the posterior probability of the jth class and L is the number of hidden layers. For regression tasks, a linear activation function is often used

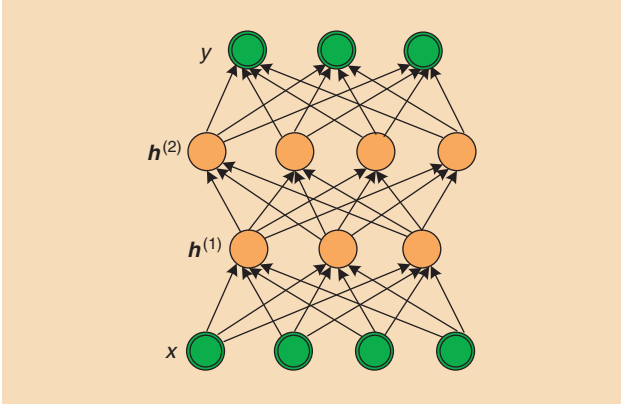
$$\tilde{y}_j = b_j^{(L+1)} + \sum_i h_i^{(L)} w_{ij}^{(L+1)}. \tag{41}$$

The set of parameters of an L-hidden-layer DNN consists of $\lambda = \{\boldsymbol{b}^{(1)}, \boldsymbol{W}^{(1)}, \ldots, \boldsymbol{b}^{(L+1)}, \boldsymbol{W}^{(L+1)}\}$. They can be optimized in a supervised way by minimizing a loss function that measures the difference between data and predicted outputs using the backpropagation algorithm [81]. For classification tasks, the cross entropy between correct and predicted class posterior probabilities is often used as the loss function

$$\mathcal{L}(\boldsymbol{y}, \tilde{\boldsymbol{y}}; \lambda) = -\sum_j y_j \log(\tilde{y}_j), \tag{42}$$



[FIG8] The graphical model representation for a three-hidden-layer DBN.



[FIG9] The model representation for a two-hidden-layer DNN.

where y_j denotes the correct class posterior probability given input, which is typically a binary value. For regression tasks, the mean square error is commonly adopted as the loss function

$$\mathcal{L}(\boldsymbol{y}, \tilde{\boldsymbol{y}}; \lambda) = \sum_j (y_j - \tilde{y}_j)^2, \tag{43}$$

where y_j and $\tilde{y}_j$ are the jth dimension of the correct and predicted outputs, respectively. A DNN for regression can be considered a probabilistic model representing a conditional PDF of $\boldsymbol{y}$ given $\boldsymbol{x}$ using a Gaussian distribution, i.e.,

$$p(\boldsymbol{y} \mid \boldsymbol{x}, \lambda) = \mathcal{N}(\boldsymbol{y}; \tilde{\boldsymbol{y}}, \boldsymbol{I}), \tag{44}$$

where $\boldsymbol{I}$ is an identity matrix and $\tilde{\boldsymbol{y}}$ depends on $\boldsymbol{x}$ and λ. Thus, minimizing the mean square error between $\tilde{\boldsymbol{y}}$ and $\boldsymbol{y}$ with respect to λ is equivalent to the ML estimation of λ.

DNNs can be powerful models of the highly complex and nonlinear relationship between inputs and outputs. However, it is difficult to train a DNN with many hidden layers. The error signal in back-propagation training decays as it is back-propagated along many hidden layers, which leads to the vanishing gradient problem [82], i.e., the lower layers cannot get much information about how to update their model parameters. Supervised training of DNNs can also result in overfitting to training data because of the power of DNNs to represent training samples. To avoid this problem, unsupervised pretraining techniques, which use DBN (stacked RBMs) weights to initialize a DNN, were proposed [17]. To build an L-hidden-layer DNN, an L-hidden-layer DBN is first trained. Then, weights of the DBN are used to initialize the weights of the DNN. After initializing the DNN weights, supervised fine-tuning is conducted using back-propagation to adjust the weights estimated in pretraining. This unsupervised pretraining strategy can provide a better starting point for supervised fine-tuning than random initialization and reduce overfitting significantly.

Besides RBMs, autoencoders (AEs) are another form of model that can be used for pretraining DNNs in a layerwise manner. An AE is a particular type of one-hidden-layer neural network [83]. It first maps an input vector $\boldsymbol{x}$ to a hidden representation $\boldsymbol{h}$ using a weight matrix $\boldsymbol{W}$ and then maps $\boldsymbol{h}$ back into a reconstruction $\tilde{\boldsymbol{x}}$ of the same shape as $\boldsymbol{x}$ using a weight matrix $\boldsymbol{W}'$. The two weight matrices may optionally be constrained: $\boldsymbol{W}' = \boldsymbol{W}^\top$. The parameters are optimized such that the average reconstruction error from $\boldsymbol{x}$ to $\tilde{\boldsymbol{x}}$ is minimized. The reconstruction error can be measured using either the mean square error or the cross-entropy criterion depending on the assumed distribution on the input features.

To prevent the hidden layer from simply learning the identity transform, a common modification of the AE is the DAE [21], which is trained to reconstruct the original input from a corrupted copy. Compared with RBMs, one of the advantages of using AEs and DAEs is that many traditional optimization algorithms for neural networks can be used in training. The DAE can also be stacked to form a particular type of DNN, called a deep DAE, through unsupervised pretraining and supervised fine-tuning. While pretraining each layer, the hidden representations given by the DAE of the layer below are used as the input to the current layer. For supervised fine-tuning, an output layer is added on top of the network and the weights of the entire network are adjusted to minimize the cost function [83].

ACOUSTIC MODELING USING DEEP LEARNING TECHNIQUES FOR SPSG

Given the success of applying deep learning to a variety of speech tasks, we believe that the approach can also be applied to acoustic speech modeling in speech generation to overcome the limitations mentioned earlier and to achieve better input-to-cluster and/or cluster-to-feature mapping. Applications of the deep learning techniques to SPSG had not been investigated until very recently. During the last year, several articles on the topic for speech synthesis [23]–[26], [33], [34], voice conversion [27]–[29], and speech enhancement [30]–[32] have been published. They reported positive results that the deep learning techniques improved the naturalness, similarity, and/or quality of generated speech. These deep learning approaches can be classified into three categories according to the modeling steps, as well as the relationship between the input and output features represented in the model.

CLUSTER-TO-FEATURE MAPPING USING DEEP GENERATIVE MODELS

In this approach, the deep learning techniques are applied to the cluster-to-feature mapping step of acoustic modeling for SPSG, i.e., to describe the distribution of acoustic features at each cluster. The input-to-cluster mapping, which determines the clusters from the input features, still uses conventional approaches, such as decision trees and state-duration PDFs in HMM-based speech synthesis and posterior probabilities of mixture components in GMM-based voice conversion. One example of this approach is HMM-based speech synthesis using RBMs and DBNs for spectral modeling [24]. This work improves the conventional spectral modeling approach in HMM-based parametric speech synthesis. Improvement was achieved in two aspects: First, raw spectral envelopes extracted by speech transformation and representation based on adaptive interpolation of weighted spectrum (STRAIGHT) analysis [84] rather than the low-dimensional representations, such as MCCs or line spectral pairs (LSPs) derived from these spectral envelopes, were modeled. Second, RBMs and DBNs were adopted to replace single Gaussian distributions at the leaf nodes of decision trees. The model structure of this approach is shown in Figure 10. To simplify model training with high-dimensional spectral features, decision trees and state alignments were assumed to be given.

At the acoustic feature extraction stage using STRAIGHT analysis, original spectral envelopes were stored in addition to spectral parameters. The context-dependent HMMs for low-dimensional spectral parameters and F_0 features were estimated according to the approach introduced in the section "HMM-Based Speech Synthesis." A single Gaussian distribution was used to model the spectral parameters at each leaf node of the decision trees. Then, a state-level forced alignment was carried out with the trained HMMs. The state boundaries obtained were used to gather the

spectral envelopes for each decision tree's leaf node. Then, an RBM or a DBN was trained at each leaf node according to the ML criterion. In this approach, the spectral envelope features at each frame consisting of static, velocity, and acceleration components correspond to the visible vector v in (23) for RBMs and (38) for DBNs.

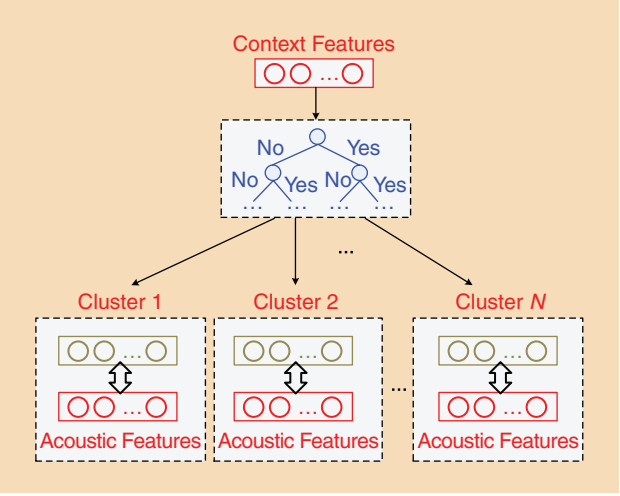
To simplify model estimation, each dimension of the spectral envelope features was normalized to zero mean and unit variance before training RBMs or DBNs, and the variance parameters σ_i^2 in (25) were fixed to 1 for each leaf node. As a result, a set of context-dependent RBM-HMMs or DBN-HMMs is trained for modeling the spectral envelopes.

At synthesis time, the speech parameter-generation algorithm was used to generate the spectral envelopes. The optimal sequences of spectral envelopes were determined so as to maximize their output probability given the RBM-HMM or the DBN-HMM. If a single Gaussian distribution is adopted as the state-output PDFs of HMMs, and the state sequence is given, there is a closed-form solution to determine the optimal acoustic feature trajectories [7]. However, the marginal PDFs of RBMs and DBNs are much more complicated than a single Gaussian distribution. Thus, there is no closed-form solution to find the optimal acoustic feature trajectories. To avoid this problem, a Gaussian approximation was applied before the parameter-generation stage as a simplification. At each decision tree leaf node of decision trees, a Gaussian distribution $\mathcal{N}(v; \boldsymbol{\mu}, \Sigma)$ was constructed, where

$$\boldsymbol{\mu} = \arg\max_{v} p(v \mid \lambda) \tag{45}$$

was the mode vector estimated [24] from $p(v \mid \lambda)$ for each RBM or DBN and Σ was a diagonal covariance matrix computed from the training samples associated with the leaf node. Because each dimension of the training samples of v was normalized to zero mean and unit variance, a denormalization processing was conducted before parameter-generation to derive the distributions of the original spectral envelope features from the estimated $\boldsymbol{\mu}$ and Σ. The RBMs/DBNs at the leaf nodes were replaced by these Gaussian distributions at the synthesis stage. Therefore, the speech parameter-generation algorithm can be followed to predict the spectral envelopes. For details about the mode estimation algorithm, refer to [24].

A group of subjective evaluations has been conducted to prove the effectiveness of this approach [24]. Some evaluation results are summarized and shown in Table 1. In this table, each line presents the preference percentages given by a preference listening test conducted between two systems. For example, the first row means that 48% of the stimuli generated by the GMM system was judged by the listeners to be better than those of the baseline system. The percentage of converse preference was 18.67%. The baseline system was constructed using Mel-cepstra and single Gaussian distributions for cluster-to-feature mapping. At training time, Mel-cepstra were derived from the spectral envelopes extracted by STRAIGHT. At synthesis time, the spectral envelopes recovered from the generated mel-cepstra were sent into STRAIGHT to reconstruct speech waveforms. A system using spectral envelopes and single Gaussian distributions for cluster-to-feature mapping



[FIG10] A model structure of cluster-to-feature mapping using RBMs for HMM-based speech synthesis [24].

was also constructed. However, it was found that this system had very similar synthetic results to the baseline system. Some detailed explanation can be found in [24], which means that simply replacing mel-cepstra with spectral envelopes is not helpful if the model structures are not modified accordingly. Therefore, the baseline system was adopted as a representative for these two systems in the subjective evaluation to simplify the test design. The GMM and RBM systems adopted GMMs of eight mixtures and RBMs of 50 hidden units to model the distribution of spectral envelopes at each leaf node of the decision trees. No postfiltering techniques, such as GV-based parameter-generation [13], were applied to any of these systems. It can be seen from the table that the use of RBMs to model the spectral envelopes at each leaf node achieved significantly better naturalness than the use of single Gaussian distributions and GMMs. A comparison between the spectral envelopes generated by the baseline system and the RBM system is shown in Figure 11. From this figure, we can observe the enhanced formant structures after modeling the spectral envelopes using RBMs.

In addition to speech synthesis, this approach was also applied to other speech generation tasks, such as voice conversion [27]. Similar to conventional GMM-based voice conversion, the input-to-cluster mapping in [27] was determined by the posterior probabilities of mixture components of a trained GMM, given the input acoustic features. Then, RBMs were adopted to model the joint PDFs between the source and target acoustic features for each cluster. The subjective evaluation results also demonstrated the effectiveness of

[TABLE 1] THE SUBJECTIVE PREFERENCE SCORES (%) AMONG SPEECH SYNTHESIZED USING THE BASELINE, GMM, AND RBM SYSTEMS.

BASELINE	GMM	RBM	N/P*	p
18.67	**48**	–	33.33	0.0014
5.33	–	**70.67**	24	0
–	16	**69.33**	14.67	0

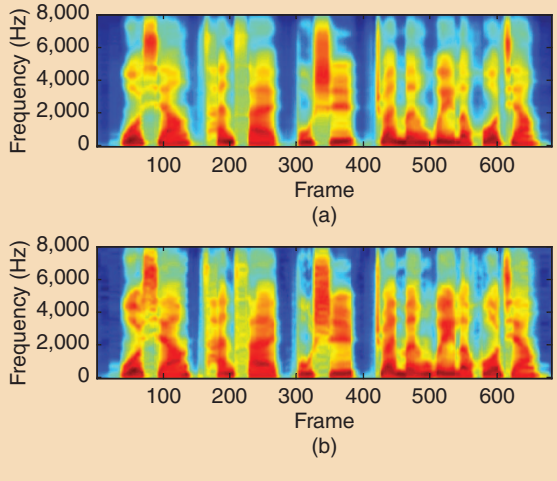
* N/P denotes "no preference" [24].
The systems that achieved significantly better preference at the $p < 0.05$ level are in bold font.

this approach when either MCCs or spectral envelopes were used as spectral features. The mean opinion score (MOS) of similarity of the converted speech improved from 2.83 to 3.13, and the MOS of naturalness increased from 2.90 to 3.45, respectively [27].

INPUT-TO-FEATURE MAPPING USING DEEP JOINT MODELS

This approach uses a single deep generative model to achieve the integrated input-to-feature mapping by modeling the joint PDF between the input and output features. For example, a synthesis method using a multidistribution DBN (MD-DBN) has been proposed in [25] with input features capturing linguistic contexts and output features being acoustic features. More specifically, the input contextual features for speech synthesis were the tonal syllables in Mandarin Chinese, which were encoded within a 1-of-k code following the categorical distribution (i.e., the generalized Bernoulli distribution). The output acoustic features to be generated consisted of syllable-level spectrum and excitation features. Each syllable was represented by an acoustic feature supervector, which consisted of multiple frames of Mel-generalized cepstral coefficients (MGCs), log-energy, $\log F_0$, and voiced/unvoiced (U/V) flags. These frames were uniformly spaced within the boundary of a syllable. Different types of acoustic features including spectrum and excitation parameters are modeled by a single network so that the correlation between them can be modeled. Syllable duration was modeled and predicted separately in this framework.

To model the different distributions of the binary data (i.e., the U/V flags) and the continuous data (i.e., the MGCs and $\log F_0$), the approach used an MD-DBN, as shown in Figure 12. This consisted of the building blocks of RBMs, with different types of distribution units in the visible layer. Gaussian distributions were used for the spectral data and $\log F_0$, and Bernoulli distributions for the U/V flags, to form the Gaussian–Bernoulli RBM (GB-RBM) for the bottom layer. Training of the MD-DBN began with unsupervised learning, where an MD-DBN with $L-1$ hidden layers was first trained using the acoustic features as observations as shown in the right part of Figure 12. The MD-DBN was built by stacking up multiple Bernoulli RBMs (B-RBMs) on top of the bottom GB-RBM layer; thus, the depth of the model could be easily controlled. This was followed by supervised learning where the $(L-1)$th layer was extended with a 1-of-k vector $\boldsymbol{x}$ that encoded Mandarin syllable IDs and then learned one more layer on top. This additional layer modeled the joint distribution between the syllable IDs and the hidden activations of the supervector using the Categorical-Bernoulli RBM (CB-RBM).



[FIG11] The spectrograms of a segment of synthetic speech using (a) the baseline system and (b) the RBM system [24].

This training paradigm has three advantages over HMM-based synthesis: 1) It models all training data in a centralized network and avoids data partitioning. Instead of using thousands of Gaussian distributions to piece the acoustic space together as in the HMM-based approach, this approach uses only one MD-DBN to portray the whole acoustic space, which potentially reduces the requirements of training data and increases the efficiency of model parameters. 2) The supervector consists of multiple acoustic frames from a syllable with temporal dynamics intact, which can be captured by the MD-DBN. This differs from the HMM-based synthesis, which assumes that acoustic observations are dependent only on the current hidden state. Since the correlations in the temporal domain can be captured directly by the MD-DBN, the use of dynamic features can be eliminated. 3) In the frequency domain, the correlations between spectral coefficients within a single frame can also be modeled by the MD-DBN, which does not adopt any independence assumptions such as those introduced by the use of a GMM with a diagonal covariance matrix. As a result, the decoupling process in the speech feature extraction can be eliminated to preserve more information.

At synthesis time, the contextual features $\boldsymbol{x}$ were first determined for each syllable by text analysis. Then, alternative Gibbs sampling using $P(h_i^{(L)} = 1 \mid \boldsymbol{x}, \boldsymbol{h}^{(L-1)}, \lambda)$ and $P(h_j^{(L-1)} = 1 \mid \boldsymbol{h}^{(L)}, \lambda)$ were conducted with the $\boldsymbol{x}$ clamped to update $\boldsymbol{h}^{(L-1)}$ until convergence or a maximum number of iterations was reached. Then, the acoustic feature supervector was predicted as the mean vector of $p(\boldsymbol{y} \mid \boldsymbol{h}^{(1)}, \lambda)$, which was determined by recursively generating hidden variables from $\boldsymbol{h}^{(L-1)}$ to $\boldsymbol{h}^{(1)}$. Finally, the generated acoustic features were interpolated according to the predicted syllable durations and were sent into the Mel log spectrum approximation filter [85] to reconstruct the speech waveforms. No postfiltering or global-variance-based voice enhancement techniques were incorporated.

It is worth noting that this acoustic modeling method discarded HMMs and modeled the joint PDF between the input contextual features and the output acoustic features using one single MD-DBN without the conventional two-step mapping. Table 2 shows the five-point Likert scale MOSs of the HMM baseline (HMM), the system predicting MGCs using the proposed MD-DBN approach [DBN (MGCs)], and the system predicting both MGCs and $\log F_0$ using the MD-DBN approach [DBN (MGCs + $\log F_0$)] [25]. Comparing DBN (MGCs) with HMM, we can see that the proposed MD-DBN approach outperforms the conventional HMM baseline for modeling and

predicting spectral features. The quality degradation from DBN(MGCs) to DBN(MGCs + $\log F_0$) suggests that the low-dimensional F_0 features are not well modeled when combined with high-dimensional spectrum features.

INPUT-TO-FEATURE MAPPING USING DEEP CONDITIONAL MODELS

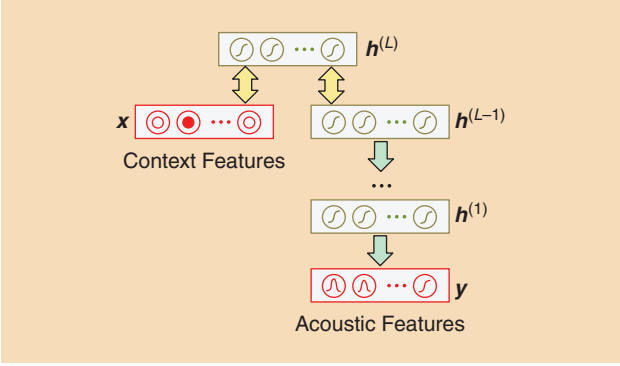
Similar to the previous approach, this one predicts acoustic features from inputs using an integrated deep generative model. The difference is that this approach models a conditional PDF of output acoustic features, given input features instead of their joint PDF.

A DNN-based speech synthesis approach was proposed in [26]. In this approach, context and acoustic features were treated as inputs and targets of a DNN, respectively, as shown in Figure 13. As introduced in the DNNs section, a DNN describes a conditional PDF of outputs given inputs using a Gaussian distribution. A text to be synthesized was first converted to a sequence of frame-level linguistic context features. The linguistic context features at each frame included binary answers to questions about contexts, numeric context descriptors, position of the current frame within a segment, and segment durations. The acoustic features at each frame were composed of MCCs, $\log F_0$, excitation aperiodicities, their derived dynamic components [3], and binary U/V decisions. The weights of the DNN were trained from pairs of inputs and targets extracted from training data. Like the DBN-based approach discussed in the section "Input-to-Feature Mapping Using Deep Joint Models," as acoustic features include both spectral and excitation parameters and a single DNN is trained, correlations between them can be modeled. At synthesis time, phoneme durations were first determined by a duration prediction module; then, frame-level linguistic context features were composed. By feeding the composed linguistic context features to the trained DNN, output acoustic features were predicted. By using these predicted output acoustic features as means along with the frame-independent variances of output acoustic features computed from all training data, the speech parameter-generation algorithm [7] generated the smooth acoustic feature trajectories. The generated acoustic feature parameters were post-processed by a postfilter (in the experiment reported in [26], postfiltering in the mel-cepstral domain [86] was applied to emphasize formant structure) and then sent to a vocoder to reconstruct a speech waveform.

A subjective preference listening test was conducted to compare the performance of the DNN-based systems with HMM-based systems [26]. The experimental results are shown in Table 3. In this experiment, HMM- and DNN-based systems with similar numbers of parameters were compared. The α in the first column of Table 3 is the scaling factor for the penalty term in the minimum description length (MDL) criterion, which is often used to control the number of parameters in HMM-based systems. It can be seen in the table that, for all three model sizes, the DNN-based system achieved better naturalness than the HMM-based system according to the p values given by hypothesis tests.

Other approaches of DNN-based TTS can be found [33], [34]. These include a hybrid approach between DNN and Gaussian process (GP)-based regression [33] to predict $\log F_0$; a DNN that maps linguistic context features to $\log F_0$ was first trained, and then the activations at the last hidden layer were used as inputs for GP-based nonparametric regression. This approach combined the parametric and nonparametric regression models. An alternative approach [34] used a vector-space representation of input texts as inputs of DNN-based TTS. This vector-space representation was derived without using any linguistic resources; only orthographic information (graphemes) was used; thus, it did not require any language knowledge to build a model.

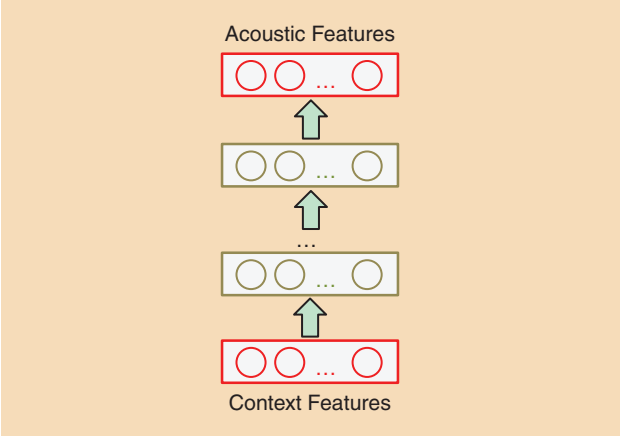
The acoustic modeling approach using deep conditional models has also been applied to other speech generation tasks, such as voice conversion [28], [29] and speech enhancement [30]–[32]. A DNN-based voice conversion approach has been proposed in [28]. In this approach, acoustic features of a source



[FIG12] The model structure of input-to-feature mapping using DBN for speech synthesis [25].

[TABLE 2] THE SUBJECTIVE EVALUATION RESULTS FOR THE DBN-BASED SPEECH SYNTHESIS [25].

SYSTEM	MOS
HMM	2.86
DBN (MGCs)	3.09
DBN (MGCs + $\log F_0$)	2.88



[FIG13] A model structure of input-to-feature mapping using DNN for speech synthesis [26].

[TABLE 3] THE SUBJECTIVE PREFERENCE SCORES (%) BETWEEN SPEECH SAMPLES FROM THE HMM- AND DNN-BASED SYSTEMS [26].

HMM (α)	DNN (# LAYERS × # UNITS)	N/P	p
15.8 (16)	**38.5** (4 × 256)	45.7	$< 10^{-6}$
16.1 (4)	**27.2** (4 × 512)	56.8	$< 10^{-6}$
12.7 (1)	**36.6** (4 × 1,024)	50.7	$< 10^{-6}$

The systems that achieved significantly better preference at the $p < 0.01$ level are shown in bold font.

voice were mapped to those of a target voice using a DNN that was initialized by concatenating two DBNs. CRBMs have also been used to construct conditional models for voice conversion. In [29], a CRBM was estimated to model a conditional PDF of acoustic features of a target voice given acoustic features from a source voice. For speech enhancement, conditional generative model-based approaches have been proposed for mapping acoustic features extracted from noisy speech to those of clean speech using DNNs [32] or DAEs [30], [31].

COMPARISONS AMONG THESE THREE APPROACHES

The cluster-to-feature mapping approach using deep generative models has the model structure most similar to conventional HMM- or GMM-based approaches. The input-to-cluster mapping step is preserved, and few modifications to the existing speech generation engines are necessary after off-line model training [24]. The input-to-feature mapping approaches using deep joint models or deep conditional models integrate the two-step mapping of acoustic modeling into a single step [25], [26], which can express complicated mapping functions more efficiently and provide better generalization than the approaches using conventional input-to-cluster mapping, such as decision trees and GMM posterior probabilities. Compared with the sampling-based parameter-generation from a DBN [25], generating acoustic features from a DNN is more straightforward [26]. However, the conditional PDF represented by a DNN is relatively simple because it is a Gaussian distribution with an identity covariance matrix as described in the "DNNs" section. Table 4 summarizes the recently proposed acoustic modeling approaches using deep learning techniques for SPSG. Some discussions on these approaches will be given in the "Discussion" section.

DISCUSSION

PERFORMANCE OF RBMs AS DENSITY MODELS

RBMs are the basis of many deep models such as DBNs and DNNs. As introduced in the "RBMs" section, RBMs have some good properties in describing the distribution of high-dimensional observations with cross-dimension correlations. The performance of GMMs and RBMs in modeling the distribution of mel-cepstra and spectral envelopes for a specific context-dependent HMM state was investigated in [23]. Spectral envelopes were extracted by STRAIGHT analysis [84], and MCCs were derived from the spectral envelopes at each frame. In the experiment, a leaf node with 720 frames was used; 520 frames were used for training and the remaining 200 frames were used as a test set. The number of mixture components in a GMM varied from 1 to 32, and the number of hidden units in an RBM varied from 1 to 1,000. The average log probabilities on the training and test sets for different model structures are shown in Table 5 for MCCs and the spectral envelopes, respectively. It can be seen from the tables that the GMMs overfit more to the training data as the model complexity increased. On the other hand, the RBMs consistently gave good generalization ability even with a large number of hidden units. It can be seen

[TABLE 4] A SUMMARY OF THE PROPOSED ACOUSTIC MODELING APPROACHES USING DEEP LEARNING TECHNIQUES FOR SPSG.

	TASK	MODEL STRUCTURE	INPUT FEATURES	GENERATED ACOUSTIC FEATURES
LING ET AL. 2013 [24]*	SPEECH SYNTHESIS	RBM/DBN-HMM	RICH CONTEXT FEATURES	SPECTRAL ENVELOPES
KANG ET AL. 2013 [25]@	SPEECH SYNTHESIS	DBN	SIMPLE LINGUISTIC FEATURES	MCCs, $\log F_0$, AND U/V
ZEN ET AL. 2013 [26]§	SPEECH SYNTHESIS	DNN	RICH LINGUISTIC CONTEXT FEATURES	MCCs, $\log F_0$, APERIODICITIES, AND U/V
LU ET AL. 2013 [34]§	SPEECH SYNTHESIS	DNN	VECTOR SPACE REPRESENTATION OF TEXTS	LSPs, $\log F_0$, AND APERIODICITIES
FERNANDEZ ET AL. 2013 [33]§	SPEECH SYNTHESIS	DNN-GP	RICH LINGUISTIC CONTEXT FEATURES	$\log F_0$
CHEN ET AL. 2013 [27]*	VOICE CONVERSION	MIXTURE OF RBMs	SPECTRAL ENVELOPES OF SOURCE VOICE	SPECTRAL ENVELOPES
NAKASHIKA ET AL. 2013 [28]§	VOICE CONVERSION	DNN	MCCs OF SOURCE VOICE	MCCs OF TARGET VOICE
WU ET AL. 2013 [29]§	VOICE CONVERSION	CRBM	MCCs OF SOURCE VOICE	MCCs OF TARGET VOICE
LU ET AL. 2013 [30]§	SPEECH ENHANCEMENT	DEEP DAE	POWER SPECTRA OF NOISY SPEECH	POWER SPECTRA OF CLEAN SPEECH
XIA ET AL. 2013 [31]§	SPEECH ENHANCEMENT	DAE	POWER SPECTRA OF NOISY SPEECH	POWER SPECTRA OF CLEAN SPEECH
XU ET AL. 2014 [32]§	SPEECH ENHANCEMENT	DNN	POWER SPECTRA OF NOISY SPEECH	POWER SPECTRA OF CLEAN SPEECH

*, @, and § denote the three categories descibed in the section "Acoustic Modeling Using Deep Learning Techniques for SPSG."
* denotes cluster-to-feature mapping using deep generative models.
@ denotes input-to-feature mapping using deep joint models.
§ denotes input-to-feature mapping using deep conditional models.

from Table 5 that the best GMM and the best RBM had very close test-set log probabilities while modeling the MCCs. However, the RBMs gave much higher test-set log probabilities than the GMMs as shown in Table 5(b). These results can be attributed to the fact that mel-cepstral analysis decorrelates spectral parameters, whereas the advantage of RBMs is to analyze the latent patterns embedded in the high-dimensional raw data with strong interdimensional correlations, such as raw spectral envelopes.

INPUT AND TARGET FEATURES

In acoustic modeling for SPSG, the forms of input features are task dependent. The same is true for the acoustic modeling using deep learning techniques. As shown in Table 4, simple to rich linguistic context features are typically used as input features for speech synthesis [24]–[26], [33], whereas vector-space representation of input texts has also been used [34]. Input features for voice conversion are typically spectral features extracted from a source voice [27]–[29]. Likewise, input features for speech enhancement are typically power spectra extracted from noisy speech [30]–[32].

Various output acoustic features for speech generation have been used, as listed in Table 4. The discussion in the section "Performance of RBMs as Density Models" shows that RBMs and other deep generative models are good at modeling the distribution of high-dimensional acoustic features with cross-dimensional correlations. Thus, some approaches took this into account when selecting their output acoustic features. The cross-dimensional correlations represented by the deep generative models exist in both the frequency domain, e.g., by using raw power spectra or spectral envelopes at each frame [24], [30]–[32], and the temporal domain, e.g., by concatenating the acoustic features of multiple frames [25]. In some speech generation tasks, such as speech synthesis, F_0 is another important acoustic feature to be predicted in addition to spectral parameters. F_0 together with other excitation-related acoustic features, including U/V decisions and aperiodicity ratios, has also been used as a part of target features in some deep-learning-based acoustic modeling approaches [25], [26]. However, the prediction performance of $\log F_0$ was not as good as that of spectral features as shown in the experimental results in [25] and [26].

MODEL STRUCTURES AND MODEL TRAINING

As shown in Table 4, different model structures have been adopted in these approaches. RBMs and DBNs were used to represent joint PDFs and to achieve cluster-to-feature [24], [27] or input-to-feature mapping [25]. On the other hand, DNNs, CRBMs, and DAEs were adopted to represent conditional PDFs and to achieve direct input-to-feature mapping [26], [30]–[32]. The depth of architecture, i.e., the number of hidden layers, is an important characteristic of a deep model. In DBN–HMM-based speech synthesis [24], the experimental results in Table 1 show that increasing the number of layers did not improve the naturalness of synthetic speech because of the difficulty of estimating the mode of a DBN.

In other works [25], [26], [30], [32], the number of hidden layers was tuned to minimize the mean squared error between targets (data) and outputs (predicted acoustic features) on development sets. The results show that multiple hidden layers could achieve better prediction accuracy than a single hidden layer. However, the optimal depth is commonly not as deep as that used in DNN-HMM-based ASR. It is reasonable considering that the amount of training data for speech generation tasks is limited compared with ASR. In the DNN-based approaches, different initialization strategies have been employed, e.g., random initialization for speech synthesis [26], structured pretraining using DBNs and NNs for voice conversion [28], and pretraining using stacked AEs or RBMs for speech enhancement [30], [32]. Considering the heavy computational cost of training RBMs, DNNs, and other deep models, graphics processing unit-based acceleration was applied to reduce the training time [25], [26].

A COMPARISON BETWEEN SPEECH SYNTHESIS AND RECOGNITION BOTH USING DNN-HMMs

The DNN-HMM is the dominant form of acoustic modeling with deep structures for ASR [22]. In this approach, a DNN is trained to map input acoustic features (e.g., mel-frequency cepstral coefficients, log-filterbank features, etc.) to posterior probabilities of leaf nodes of decision trees at each frame. HMMs are used to connect the hidden states with the higher-level linguistic representations for decoding with language models at recognition time. While there seems to be a converging deep learning

[TABLE 5] THE AVERAGE LOG PROBABILITIES ON THE TRAINING AND TEST SETS WHEN MODELING THE MEL-CEPSTRA AND SPECTRAL ENVELOPES OF A SPECIFIC STATE USING DIFFERENT MODELS [23].

MEL-CEPSTRA COEFFICIENTS	AVERAGE LOG PROBABILITY		NUMBER OF PARAMETERS
	TRAIN	TEST	
GMM (1)-DIAG	−58.176	−56.380	82
GMM (4)-DIAG	−51.188	−53.097	328
GMM (16)-DIAG	−40.869	−59.492	1,312
GMM (32)-DIAG	−29.973	−72.056	2,624
GMM (1)-FULL	−30.883	−54.648	902
RBM (1)	−56.464	−55.244	83
RBM (10)	−52.416	−52.660	461
RBM (50)	−51.840	−53.636	2,141
RBM (200)	−53.554	−55.020	8,441
RBM (1,000)	−55.797	−56.940	42,041
SPECTRAL ENVELOPES	**AVERAGE LOG PROBABILITY**		**NUMBER OF PARAMETERS**
	TRAIN	TEST	
GMM (1)-DIAG	−727.915	−728.647	1,026
GMM (4)-DIAG	−599.642	−648.818	4,104
GMM (16)-DIAG	−485.072	−665.609	16,416
GMM (32)-DIAG	−379.980	−717.523	32,832
GMM (1)-FULL	2,207.177	−89,202.438	132,354
RBM (1)	−685.799	−700.938	1,027
RBM (10)	−629.906	−649.823	5,653
RBM (50)	−587.146	−628.222	30,317
RBM (200)	−576.461	−617.480	103,313
RBM (1,000)	−562.439	−583.169	514,513

The numbers in the brackets indicate the Gaussian mixture numbers for the GMMs and the hidden unit numbers or the RBMS. "DIAG" and "FULL" denote using diagonal and full covariance matrices, respectively.

architecture based on the DNN-HMM for the dominant use in ASR, there has been a greater variety of model structures proposed for SPSG using deep learning techniques, where the variety can be seen in Table 4. Among them, the DNN-based conditional modeling approach [26], [32] adopts a model structure quite similar to the DNN-HMM for acoustic modeling in ASR. One main difference is in the activation functions used at the DNN's output layers: the softmax layer for multiclass classification in ASR versus the linear layer for regression in SPSG.

In DNN–HMM-based ASR, acoustic features are the input to a DNN for classification, while DNN-based SPSG predicts acoustic features for speech generation. Therefore, the acoustic features used in DNN-based SPSG should take into account the requirement of reconstructing speech waveforms. Some acoustic features that are not adopted in DNN–HMM-based ASR, such as excitation-related features [26] and power spectra [32], have been used in DNN-based SPSG.

CONCLUSIONS

This article provides an overview of the emerging speech generation approaches using deep learning techniques. Compared with the conventional acoustic modeling methods in SPSG based on the use of HMMs and GMMs, deep joint models (e.g., RBMs and DBNs) and deep conditional models (e.g., CRBMs and DNNs), which we reviewed in this article, are better able to describe the complex and nonlinear relationship between the inputs and targets of the SPSG system and, therefore, improve the naturalness, similarity to the target speaker, and quality of the generated speech. Various implementations of building acoustic models using deep learning for SPSG in the current literature have been reviewed and compared. To facilitate a review of the area and to offer insights into the different approaches reported in the literature, we categorize them into three classes, describe and analyze each, and make connections in a systematic manner.

Despite the empirical successes of a range of deep learning methods in SPSG as reviewed in this article, there remain important issues that need further investigation to make full use of the intrinsic strength of deep learning models and methods in SPSG. For example, current attempts have not achieved positive results in modeling and prediction of F_0 using deep generative models [25], [26]. Considering the different physiological mechanisms between the production of F_0 and of spectral features, deep model structures designed specifically for F_0 modeling and prediction may be necessary. Furthermore, few considerations have been made thus far in deep learning approaches to model the temporal dependencies among the sequence of acoustic features. We believe that a promising direction to pursue in the near future is to apply the deep generative models with better temporal modeling abilities, such as recurrent neural networks, to the SPSG tasks in the future.

AUTHORS

Zhen-Hua Ling (zhling@ustc.edu.cn) received his B.E. degree in electronic information engineering and his M.S. and Ph.D. degrees in signal and information processing from the University of Science and Technology of China, Hefei, in 2002, 2005, and 2008, respectively. From 2007 to 2008, he was a Marie Curie Fellow at the Centre for Speech Technology Research, University of Edinburgh, United Kingdom. From 2008 to 2011, he was a joint postdoctoral researcher at the University of Science and Technology of China and iFLYTEK Co., Ltd., China. He is currently an associate professor at the University of Science and Technology of China. He was also with the University of Washington, United States, as a visiting scholar from 2012 to 2013. His research interests include speech processing, speech synthesis, voice conversion, speech analysis, and speech coding. He received the 2010 IEEE Signal Processing Society Young Author Best Paper Award. He is a Member of the IEEE.

Shi-Yin Kang (sykang@se.cuhk.edu.hk) received his B.S. degree in automation and his M.S. degree in computer science and technology from Tsinghua University, Beijing, China, in 2007 and 2010, respectively. He is currently pursuing his Ph.D. degree in the Department of Systems Engineering and Engineering Management of The Chinese University of Hong Kong, with a research focus in speech synthesis. His current research interests include statistical parametric speech synthesis and applications in machine learning.

Heiga Zen (heigazen@google.com) received his Ph.D. degree from the Nagoya Institute of Technology, Japan, in 2006. Before joining Google in 2011, he was an intern/cooperative researcher at the IBM T.J. Watson Research Center, Yorktown Heights, New York, from 2004 to 2005, and a research engineer at Toshiba Research Europe Ltd. Cambridge Research Laboratory, Cambridge, United Kingdom, from 2008 to 2011. His research interests include speech synthesis and recognition. He was one of the original authors and the first maintainer of the hidden Markov model-based speech synthesis system.

Andrew Senior (andrewsenior@google.com) received his Ph.D. degree from the University of Cambridge, United Kingdom, for his work on handwriting recognition with recurrent neural networks. He worked at the IBM T.J. Watson Research Center, Yorktown Heights, New York, in the areas of fingerprint, face, and audio-visual speech recognition, as well as visual tracking and privacy protection. He is currently a research scientist at Google, where he works on deep learning for speech recognition and synthesis. He coauthored *A Guide to Biometrics* and recently coorganized an International Conference on Machine Learning Workshop on deep learning.

Mike Schuster (schuster@google.com) received his Diplom Ingenieur degree in electrical engineering from Gerhard-Mercator University in Duisburg, Germany, and his Ph.D. degree from the Nara Institute of Science and Technology, Japan. He was with Advanced Telecommunications Research Laboratories in Kyoto, Japan; Nuance in the United States; and Nippon Telegraph and Telephone in Japan, where he mostly worked on various machine-learning techniques applied to sequential modeling and, in particular, speech recognition. He is currently a research scientist at Google working on machine-learning techniques for speech recognition, speech synthesis, translation, recommendation systems, and related areas.

Xiao-Jun Qian (xjqian@se.cuhk.edu.hk) received his B.E. degree in electrical engineering from Fudan University, Shanghai,

China, in 2007. From 2007 to 2010, was with the speech group of Microsoft Research Asia. He joined the Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, as a Ph.D. candidate in 2009. His research interests include discriminative training, subspace acoustic modeling, and deep learning. He was the recipient of the 2010 Microsoft Research Asia Fellowship.

Helen Meng (hmmeng@se.cuhk.edu.hk) received her S.B., S.M., and Ph.D. degrees, all in electrical engineering, from the Massachusetts Institute of Technology. She joined The Chinese University of Hong Kong in 1998, where she is currently a professor and chair in the Department of Systems Engineering and Engineering Management. She was also the associate dean of research of the Faculty of Engineering between 2005 and 2010. Her research interest is in the area of human–computer interaction via multimodal and multilingual spoken language systems, speech retrieval technologies, and computer-aided pronunciation training. She served as the editor-in-chief of *IEEE Transactions on Audio, Speech, and Language Processing* from 2009 to 2011. She is an elected board member of the International Speech Communication Association as well as an elected member of the IEEE Signal Processing Society Board of Governors. She is a Fellow of the IEEE.

Li Deng (deng@microsoft.com) received his Ph.D. degree from the University of Wisconsin–Madison. He was a tenured professor from 1989 to 1999 at the University of Waterloo, Ontario, Canada, and then joined Microsoft Research, Redmond, Washington, where he is currently a principal research manager of its Deep Learning Technology Center. Since 2000, he has also been an affiliate full professor at the University of Washington, Seattle, teaching computer speech processing. He has been granted more than 60 U.S. or international patents, and has received numerous awards and honors bestowed by the IEEE, the International Speech Communication Association (ISCA), the Acoustical Society of America (ASA), and Microsoft, including the 2013 IEEE Signal Processing Society Best Paper Award on deep neural networks for speech recognition. He has authored or coauthored four books. He is a Fellow of the ASA, IEEE, and ISCA. He was the editor-in-chief of *IEEE Signal Processing Magazine* from 2009 to 2011 and the editor-in-chief of *IEEE Transactions on Audio, Speech, and Language Processing* from 2012 to 2014. His recent research interests and activities have been focused on deep learning and machine intelligence applied to large-scale text analysis and to speech/language/image multimodal processing.

REFERENCES

[1] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, H. Yamagishi, and K. Oura, "Speech synthesis based on hidden Markov models," *Proc. IEEE*, vol. 101, no. 5, pp. 1234–1252, 2013.

[2] 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.

[3] H. Zen, T. Toda, M. Nakamura, and K. Tokuda, "Details of Nitech HMM-based speech synthesis system for the Blizzard Challenge 2005," *IEICE Trans. Inf. Syst.*, vol. E90-D, no. 1, pp. 325–333, 2007.

[4] Z.-H. Ling, Y.-J. Wu, Y.-P. Wang, L. Qin, and R.-H. Wang, "USTC system for Blizzard Challenge 2006: An improved HMM-based speech synthesis method," in *Proc. Blizzard Challenge Workshop*, 2006.

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

[6] 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.

[7] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter-generation algorithms for HMM-based speech synthesis," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2000, vol. 3, pp. 1315–1318.

[8] H. Zen, K. Tokuda, and T. Kitamura, "Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic feature vector sequences," *Comput. Speech Lang.*, vol. 21, no. 1, pp. 153–173, 2006.

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

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

[11] Y.-J. Wu and R.-H. Wang, "Minimum generation error training for HMM-based speech synthesis," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP),* 2006, pp. 89–92.

[12] Y.-J. Wu and K. Tokuda, "Minimum generation error training with direct log spectral distortion on LSPs for HMM-based speech synthesis," in *Proc. Interspeech*, 2008, pp. 577–580.

[13] 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.

[14] T. Tiomkin, D. Malah, and S. Shechtman, "Statistical text-to-speech synthesis based on segment-wise representation with a norm constraint," *IEEE Trans. Audio Speech Lang. Processing*, vol. 18, no. 5, pp. 1077–1082, 2010.

[15] Z.-H. Ling and L.-R. Dai, "Minimum Kullback-Leibler divergence parameter-generation for HMM-based speech synthesis," *IEEE Trans. Audio Speech Lang. Processing*, vol. 20, no. 5, pp. 1492–1502, 2012.

[16] G. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computat.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[17] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[18] R. Salakhutdinov, "Learning deep generative models," Ph.D. thesis, Univ. of Toronto, 2009.

[19] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," in *Parallel Distributed Processing*, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA: MIT Press, 1986, vol. 1, ch. 6, pp. 194–281.

[20] L. Deng, M. Seltzer, D. Yu, A. Acero, A. Mohamed, and G. Hinton, "Binary coding of speech spectrograms using a deep auto-encoder," in *Proc. Interspeech*, 2010, pp. 1692–1695.

[21] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Dec. 2010.

[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 Processing Mag.*, vol. 29, no. 6, pp. 82–97, 2012.

[23] Z.-H. Ling, L. Deng, and D. Yu, "Modeling spectral envelopes using restricted Boltzmann machines for statistical parametric speech synthesis," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP),* 2013, pp. 7825–7829.

[24] 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. Audio Speech Lang. Processing*, vol. 21, no. 10, pp. 2129–2139, 2013.

[25] S.-Y. Kang, X.-J. Qian, and H. Meng, "Multi-distribution deep belief network for speech synthesis," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 8012–8016.

[26] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 7962–7966.

[27] L.-H. Chen, Z.-H. Ling, Y. Song, and L.-R. Dai, "Joint spectral distribution modeling using restricted Boltzmann machines for voice conversion," in *Proc. Interspeech*, 2013, pp. 3052–3056.

[28] T. Nakashika, R. Takashima, T. Takiguchi, and Y. Ariki, "Voice conversion in high-order eigen space using deep belief nets," in *Proc. Interspeech*, 2013, pp. 369–372.

[29] Z.-Z Wu, E.S. Chng, and H.-Z. Li, "Conditional restricted Boltzmann machine for voice conversion," in *Proc. ChinaSIP*, 2013, pp. 104–108.

[30] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder," in *Proc. Interspeech*, 2013, pp. 436–440.

[31] B.-Y. Xia and C.-C. Bao, "Speech enhancement with weighted denoising auto-encoder," in *Proc. Interspeech*, 2013, pp. 3444–3448.

[32] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal Processing Lett.*, vol. 21, no. 1, pp. 65–68, 2014.

[33] R. Fernandez, A. Rendel, B. Ramabhadran, and R. Hoory, "f_0 contour prediction with a deep belief network-Gaussian process hybrid model," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 6885–6889.

[34] 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. 261–265.

[35] S.-Y. Kang and H. Meng, "Statistical parametric speech synthesis using weighted multi-distribution deep belief network," in *Proc. Interspeech,* 2014, pp. 1959–1963.

[36] Y. Qian, Y.-C. Fan, W.-P. Hu, and F. K. Soong, "On the training aspects of deep neural networks (DNN) for parametric TTS synthesis," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 3857–3861.

[37] H. Zen and A. Senior, "Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 3872–3876.

[38] T. Nakashika, T. Takiguchi, and Y. Ariki, "Voice conversion in time-invariant speaker-independent space," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 7939–7943.

[39] L.-H. Chen, Z.-H. Ling, L.-J. Liu, and L.-R. Dai, "Voice conversion using deep neural networks with layer-wise generative training," in *Proc. IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, 2014, pp. 1859–1872.

[40] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "Dynamic noise aware training for speech enhancement based on deep neural networks," in *Proc. Interspeech* (to be published).

[41] K. Tokuda, T. Kobayashi, and S. Imai, "Speech parameter-generation from HMM using dynamic features," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 1995, pp. 660–663.

[42] 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. Processing*, vol. 17, no. 6, pp. 1208–1230, 2009.

[43] J. Yamagishi and T. Kobayashi, "Average-voice-based speech synthesis using HSMM-based speaker adaptation and adaptive training," *IEICE Trans. Inf. Syst.*, vol. E90-D, no. 2, pp. 533–543, 2007.

[44] J. Yamagishi, K. Onishi, T. Masuko, and T. Kobayashi, "Acoustic modeling of speaking styles and emotional expressions in HMM-based speech synthesis," *IEICE Trans. Inf. Syst.*, vol. E88-D, no. 3, pp. 503–509, 2005.

[45] M. Tachibana, J. Yamagishi, T. Masuko, and T. Kobayashi, "Speech synthesis with various emotional expressions and speaking styles by style interpolation and morphing," *IEICE Trans. Inf. Syst.*, vol. E88-D, no. 11, pp. 2484–2491, 2005.

[46] T. Nose, J. Yamagishi, 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.

[47] L. Saheer, J. Dines, and P. N. Garner, "Vocal tract length normalization for statistical parametric speech synthesis," *IEEE Trans. Audio Speech Lang. Processing*, vol. 20, no. 7, pp. 2134–2148, 2012.

[48] H. Zen, N. Braunschweiler, S. Buchholz, M.J.F. Gales, K. Knill, S. Krstulovic, and J. Latorre, "Statistical parametric speech synthesis based on speaker and language factorization," *IEEE Trans. Audio Speech Lang. Processing*, vol. 20, no. 6, pp. 1713–1724, 2012.

[49] Z.-H. Ling, K. Richmond, and J. Yamagishi, "Articulatory control of HMM-based parametric speech synthesis using feature-space-switched multiple regression," *IEEE Trans. Audio Speech Lang. Processing*, vol. 21, no. 1, pp. 207–219, 2013.

[50] K. Tokuda, H. Zen, and A. Black, "An HMM-based speech synthesis system applied to English," in *Proc. IEEE Speech Synthetic Workshop, 2002, CD-ROM Proc.*

[51] 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.

[52] J. Odell, "The use of context in large vocabulary speech recognition," Ph.D. thesis, Cambridge Univ., 1995.

[53] 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.

[54] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Duration modeling in HMM-based speech synthesis system," in *Proc. ICSLP*, 1998, vol. 2, pp. 29–32.

[55] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, "Voice conversion through vector quantization," *J. Acoust. Soc. Jpn. (E)*, vol. 11, no. 2, pp. 71–76, 1990.

[56] Y. Stylianou, O. Cappe, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Trans. Audio Speech Lang. Processing*, vol. 6, no. 2, pp. 131–142, 1998.

[57] D. Erro, A. Moreno, and A. Bonafonte, "Voice conversion based on weighted frequency warping," *IEEE Trans. Audio Speech Lang. Processing*, vol. 18, no. 5, pp. 922–931, 2010.

[58] S. Desai, A. W. Black, B. Yegnanarayana, and K. Prahallad, "Spectral mapping using artificial neural networks for voice conversion," *IEEE Trans. Audio Speech Lang. Processing*, vol. 18, no. 5, pp. 954–964, 2010.

[59] E. Helander, H. Silen, T. Virtanen, and M. Gabbouj, "Voice conversion using dynamic kernel partial least squares regression," *IEEE Trans. Audio Speech Lang. Processing*, vol. 20, no. 3, pp. 806C817, 2012.

[60] D. Saito, S. Watanabe, A. Nakamura, and N. Minematsu, "Statistical voice conversion based on noisy channel model," *IEEE Trans. Audio Speech Lang. Processing*, vol. 20, no. 6, pp. 1784–1794, 2012.

[61] K. Park and H. Kim, "Narrowband to wideband conversion of speech using GMM based transformation," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2000, pp. 1843–1846.

[62] A. Mouchtaris, J. Van der Spiegel, P. Mueller, and P. Tsakalides, "A spectral conversion approach to single-channel speech enhancement," *IEEE Trans. Audio Speech Lang. Processing*, vol. 15, no. 4, pp. 1180–1193, 2007.

[63] T. Toda, M. Nakagiri, and K. Shikano, "Statistical voice conversion techniques for body-conducted unvoiced speech enhancement," *IEEE Trans. Audio Speech Lang. Processing*, vol. 20, no. 9, pp. 2505–2517, 2012.

[64] T. Toda, A. Black, and K. Tokuda, "Statistical mapping between articulatory movements and acoustic spectrum using a Gaussian mixture model," *Speech Commun.*, vol. 50, pp. 215–227, 2008.

[65] H. Zen. (2013). Deep learning in speech synthesis. *Keynote speech given at ISCA SSW8*. [Online]. Available: http://research.google.com/pubs/archive/41539.pdf

[66] L. Deng and D. O'Shaughnessy, *Speech Processing: A Dynamic and Optimization-Oriented Approach*. New York: Marcel Dekker, 2003.

[67] J. Sun and L. Deng, "An overlapping-feature based phonological model incorporating linguistic constraints: Applications to speech recognition," *J. Acoust. Soc. Am.*, vol. 111, pp. 1086–1101, 2002.

[68] L. Deng, G. Ramsay, and D. Sun, "Production models as a structural basis for automatic speech recognition," *Speech Commun.*, vol. 33, nos. 2–3, pp. 93–111, Aug. 1997.

[69] L. Deng, "Switching dynamic system models for speech articulation and acoustics," in *Mathematical Foundations of Speech and Language Processing*. New York: Springer-Verlag, 2003, pp. 115–134.

[70] D. Yu, L. Deng, and G. Dahl, "Roles of pre-training and fine-tuning in context-dependent DBN-HMMs for real-world speech recognition," in *Proc. NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2010.

[71] G. Dahl, D. Yu, L. Deng, and A. Acero, "Large vocabulary continuous speech recognition with context-dependent DBN-HMMs," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 4688–4691.

[72] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[73] A. Mohamed, G. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. Audio Speech Lang. Processing*, vol. 20, no. 1, pp. 14–22, 2012.

[74] T.N. Sainath, B. Kingsbury, H. Soltau, and B. Ramabhadran, "Optimization techniques to improve training speed of deep neural networks for large speech tasks," *IEEE Trans. Audio Speech Lang. Processing*, vol. 21, no. 11, pp. 2267–2276, 2013.

[75] X.-L. Zhang and Ji Wu, "Deep belief networks based voice activity detection," *IEEE Trans. Audio Speech Lang. Processing*, vol. 21, no. 4, pp. 697–710, 2013.

[76] B. Uria, S. Renals, and K. Richmond, "A deep neural network for acoustic-articulatory speech inversion," in *Proc. NIPS 2011 Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[77] G. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computat.*, vol. 14, no. 8, pp. 1711–1800, 2002.

[78] G. Hinton, "Products of experts," in *Proc. 9th Int. Conf. Artificial Neural Networks*, 1999, pp. 1–6.

[79] G. Taylor, G. Hinton, and S. Roweis, "Modeling human motion using binary latent variables," in *Proc. Advances in Neural Information Processing Systems,* 2007, pp. 1345–1352.

[80] R. Neal, "Connectionist learning of belief networks," *Artificial Intell.*, vol. 56, no. 1, pp. 71–113, 1992.

[81] D. Rumelhart, G. Hinton, and R. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[82] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient flow in recurrent nets: The difficulty of learning long-term dependencies," in *A Field Guide to Dynamical Recurrent Neural Networks*, S. Kremer and J. Kolen, Eds. Piscataway, NJ: IEEE Press, 2001, pp. 237–244.

[83] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proc. Advances in Neural Information Processing Systems*, 2007, pp. 153–160.

[84] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigne, "Restructuring speech representations using pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," *Speech Commun.*, vol. 27, nos. 3–4, pp. 187–207, 1999.

[85] T. Fukada, K. Tokuda, T. Kobayashi, and S. Imai, "An adaptive algorithm for mel-cepstral analysis of speech," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 1992, vol. 1, pp. 137–140.

[86] T. Yoshimura, "Simultaneous modeling of phonetic and prosodic parameters, and characteristic conversion for HMM-based text-to-speech systems," Ph.D. thesis, Nagoya Inst. of Tech., 2002.

[SP]