MODELING SPECTRAL ENVELOPES USING DEEP CONDITIONAL RESTRICTED BOLTZMANN MACHINES FOR STATISTICAL PARAMETRIC SPEECH SYNTHESIS

Xiang Yin, Zhen-Hua Ling, Ya-Jun Hu, Li-Rong Dai

National Engineering Laboratory of Speech and Language Information Processing,
University of Science and Technology of China, Hefei, P.R. China

byx1030@mail.ustc.edu.cn, zhling@ustc.edu.cn, hyj15475@mail.ustc.edu.cn, lrdai@ustc.edu.cn

ABSTRACT

This paper proposes a spectral modeling method using a deep conditional restricted Boltzmann machine (DCRBM) for statistical parametric speech synthesis. In this method, a DCRBM, which combines a deep neural network (DNN) with a conditional restricted Boltzmann machine (CRBM), is utilized to describe the conditional distribution of spectral envelopes given linguistic features. Compared with DNN and deep mixture density network (DMDN), DCRBM is better at describing the multimodal distribution of high-dimensional acoustic features with cross-dimension correlations. At training stage, the DNN part and the CRBM part of the DCRBM are pre-trained successively and then a unified fine-tuning of all model parameters is conducted. At synthesis time, spectral envelopes are generated from the estimated DCRBM model by iterative sampling and dynamic-feature-constrained parameter generation given linguistic features of input text. Experimental results show that our proposed method can produce more natural speech sounds than the hidden Markov model (HMM)-based, DNN-based, and DMDN-based synthesis methods. This method also outperforms previous work which adopts restricted Boltzmann machines (RBM) to model the distributions of spectral envelopes at HMM states.

Index Terms— Speech synthesis, hidden Markov model, deep neural network, restricted Boltzmann machine, spectral envelope

1. INTRODUCTION

Recently, hidden Markov model (HMM) based statistical parametric speech synthesis (SPSS) has become a mainstream speech synthesis method [1]. At training stage, the spectral, F0 and duration features are modeled simultaneously within a unified framework of HMMs [2]. At the stage of synthesis, acoustic features are predicted from corresponding HMMs through the maximum likelihood parameter generation (MLPG) algorithm under the conjugate constraint between static and dynamic features [3]. Finally, the speech waveforms are reconstructed by high quality vocoder from the predicted features. This method can synthesize highly intelligible and relatively smooth speech sounds [4].

However, the quality of its synthetic speech degrades and the inadequacy of acoustic modeling is one of the main reasons [5]. Recently, some methods to improve the acoustic modeling of SPSS using deep learning techniques have been proposed. One of them applies restricted Boltzmann machines (RBM) to model the distribution of spectral envelopes at the leaf nodes of decision trees [6, 7], considering the advantage of RBMs in describing the distribution of high-dimensional observations with cross-dimension correlations. At synthesis stage of this method, the mode vectors of trained RBMs were estimated and used to replace the Gaussian mean vectors in the parameter generation process. Another approach is to replace decision trees used in the conventional HMM-based speech synthesis with deep neural networks (DNN) in order to better model the effects of linguistic features on the distribution of acoustic features [8]. Compared with decision trees, DNN model can describe complex context dependencies and avoid the data fragmentation problem in building large decision trees. However, the conditional probability density function (PDF) of acoustic features given specific linguistic features can be multimodal since humans can speak the same text in different ways [9]. Current DNN-based acoustic modeling method fails to embody such multimodal property. Thus, a deep mixture density network (DMDN) based speech synthesis method has been proposed [9], which adopted a Gaussian mixture model (GMM) to describe the distribution of acoustic features given linguistic features. The parameters of the GMM were mapped from linguistic features using a DNN structure. Although this method can predict acoustic features more accurately than the DNN-based method and improve the naturalness of synthesized speech, its ability to model cross-dimension correlations of acoustic observations is still limited.

This paper proposes a new spectral modeling method to combine the advantages of RBM-based distribution representation in [7] and DNN-based dependency modeling in [8]. In this method, an RBM conditioned on the output of a DNN is utilized to model the distribution of spectral envelopes given linguistic features. This model structure is named deep conditional restricted Boltzmann machine (DCRBM) in this paper. DCRBM can be considered as an extension of conditional restricted Boltzmann machine (CRBM) [10], which employs a deep generative model to map input features into conditional vectors. The difference between DCRBM and DMDN is that the GMM distribution determined by the output layer of a DMDN is replaced by an RBM, which is better at describing the multimodal distribution of high-dimensional acoustic features with cross-dimension correlations as discussed in [6].

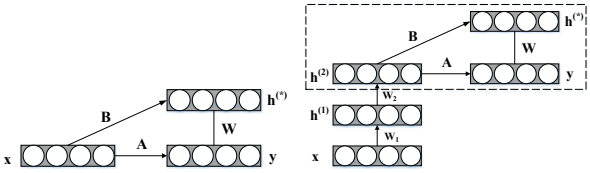
The paper is organized as follows. Section 2 describes the details of our proposed method after a brief review of CRBM. The experimental results are shown in Section 3, then the conclusion is given in Section 4.

2. METHODS

2.1. Conditional restricted Boltzmann machine

The conditional restricted Boltzmann machine (CRBM) was originally proposed to model the temporal dependency of human motion features [10] and was later applied to learn the relationship between

This work was partially funded by the National Nature Science Foundation of China (Grant No.61273032) and the Electronic Industry Development Fund of Ministry of Industry and Information Technology (Grant No. [2014]425).



(a) (b)

Fig. 1. Model structures of (a) a CRBM and (b) a two-hidden-layer DCRBM.

source and target speech in voice conversion [11]. The model structure of a CRBM is illustrated in Fig. 1(a). The links between the visible units $\mathbf{y}$ and the hidden units $\mathbf{h}^{(*)}$ are undirected. If the conditional vector $\mathbf{x}$ is given, $\mathbf{y}$ and $\mathbf{h}^{(*)}$ compose an RBM and its parameters depend on weights $\mathbf{A}$ and $\mathbf{B}$ through the two directed links. When $\mathbf{h}^{(*)} \in \{0,1\}^H$ are binary and $\mathbf{x} \in \mathbb{R}^{D_X}$ and $\mathbf{y} \in \mathbb{R}^{D_Y}$ are real-valued, the energy function of a CRBM is written as

$$E(\mathbf{y},\mathbf{h}^{(*)},\mathbf{x};\theta_C) = \sum_{i=1}^{D_Y} \frac{(y_i - a_i - \sum_k A_{ki}x_k)^2}{2} - \sum_{j=1}^{H}(b_j + \sum_k B_{kj}x_k)h_j^* - \sum_{i=1}^{D_Y}\sum_{j=1}^{H} w_{ij}h_j^* y_i, \quad (1)$$

where $\theta_C = \{\mathbf{W},\mathbf{A},\mathbf{B},\mathbf{a},\mathbf{b}\}$ is the set of parameters in the CRBM, $\mathbf{W} = \{w_{ij}\} \in \mathbb{R}^{D_Y \times H}$ are the symmetric weights between visible and hidden units, $\mathbf{A} = \{A_{ki}\} \in \mathbb{R}^{D_X \times D_Y}$ and $\mathbf{B} = \{B_{kj}\} \in \mathbb{R}^{D_X \times H}$ are matrices corresponding to the directed links in Fig. 1(a), $\mathbf{a}$ and $\mathbf{b}$ are the bias vectors of visible and hidden layers. The conditional PDF of $\mathbf{y}$ given $\mathbf{x}$ can be written as

$$p(\mathbf{y}|\mathbf{x},\theta_C) = \frac{1}{Z_{\theta_C}} \sum_{\forall h^{(*)}} \exp\{-E(\mathbf{y},\mathbf{h}^{(*)},\mathbf{x};\theta_C)\}, \quad (2)$$

where $Z_{\theta_C} = \int \sum_{\forall \mathbf{h}^{(*)}} \exp\{-E(\mathbf{y},\mathbf{h}^{(*)},\mathbf{x};\theta_C)\}d\mathbf{y}$ is the partition function.

The conditional probability of output feature y_i given the hidden layer is

$$p(y_i|\mathbf{h}^{(*)},\mathbf{x}) = \mathcal{N}(a_i + \sum_{k=1}^{D_X} A_{ki}x_k + \sum_{j=1}^{H} w_{ij}h_j^*, 1), \quad (3)$$

where $\mathcal{N}(\cdot)$ denotes a Gaussian distribution. Similarly, the conditional probability of hidden unit given y is

$$p(h_j^{(*)}|\mathbf{y},\mathbf{x}) = g(b_j + \sum_{k=1}^{D_X} B_{kj}x_k + \sum_{i=1}^{D_Y} w_{ij}y_i, 1), \quad (4)$$

where $g(\cdot)$ is a sigmoid function.

A CRBM is usually trained under maximum likelihood criterion using gradient descent method. The gradients of model parameters can be derived from the negative log-likelihood function $\mathcal{L}(\theta_C) = -\log p(\mathbf{y}|\mathbf{x},\theta_C)$ using contrastive divergence (CD) algorithm [10].

2.2. Deep conditional restricted Boltzmann machine

The structure of a DCRBM is presented in Fig. 1(b). In this figure, the dot-lined box contains the structure of a traditional CRBM with parameter set $\theta_C = \{\mathbf{W},\mathbf{A},\mathbf{B},\mathbf{a},\mathbf{b}\}$. The DNN part of this DCRBM has 2 hidden layers. We name this model as a *2-hidden-layer DCRBM* for concise expression. For a L-hidden-layer DCRBM, the activities of the top hidden layer $\mathbf{h}^{(L)}$ can be calculated as $\mathbf{h}^{(L)} = \Phi(\mathbf{x},\theta_D)$, where the mapping function $\Phi(\cdot)$ is composed of L sigmoid functions, $\theta_D = \{\mathbf{W}_1,\mathbf{c}_1,...,\mathbf{W}_L,\mathbf{c}_L\}$ are the model parameters of the DNN part, $\mathbf{W}_l$ and $\mathbf{c}_l$ are the weight matrix and the bias vector at the l-th hidden layer. The complete model parameter set of a DCRBM $\theta = \{\theta_C,\theta_D\}$ contains the CRBM part θ_C and the DNN part θ_D. Given an input feature vector $\mathbf{x}$, the DCRBM first maps it into a hidden representation $\mathbf{h}^{(L)}$. Then $\mathbf{h}^{(L)}$ acts as the conditional vector of a CRBM to determine the distribution of output feature $\mathbf{y}$.

Similar to CRBM, the criterion of training a DCRBM is to minimize the negative log-likelihood function $\mathcal{L}(\theta) = -\log p(\mathbf{y}|\mathbf{x},\theta)$ on training set, which can be written as

$$\mathcal{L}(\theta) = -\log \sum_{\forall \mathbf{h}^{(*)}} \exp\{-E(\mathbf{y},\mathbf{h}^{(*)},\mathbf{h}^{(L)};\theta_C)\} + \log(Z_{\theta_C}), \quad (5)$$

where the energy function $E(\mathbf{y},\mathbf{h}^{(*)},\mathbf{h}^{(L)};\theta_C)$ and partition function Z_{θ_C} are the same as the ones introduced in Section 2.1. After initialization, the parameter set θ is iteratively updated by gradient descent. This process is named *fine-tuning* in this paper. The gradients $\partial\mathcal{L}(\theta)/\partial\theta_C$ are calculated in the same way as CRBM training using CD approximation. The gradients $\partial\mathcal{L}\theta/\partial\theta_D$ are converted into $\partial\mathcal{L}(\theta)/\partial\mathbf{h}^{(L)} \cdot \partial\mathbf{h}^{(L)}/\partial\theta_D$, where $\partial\mathcal{L}(\theta)/\partial\mathbf{h}^{(L)}$ are given by CD approximation and $\partial\mathbf{h}^{(L)}/\partial\theta_D$ are derived from the mapping function $\Phi(\cdot)$ similar to DNN training.

In addition to initializing θ randomly, two *pre-training* methods for DCRBM estimation are implemented and compared in this paper.

(1) *DBN pre-training*

In this method, a deep belief network (DBN) [12] with L-hidden layers is built using input feature vectors $\mathbf{x}$ in the training set. A DBN is a probabilistic generative model, whose parameters can be learnt in a layer-by-layer manner using a stack of RBMs [12]. The first RBM is built using all feature vectors $\mathbf{x}$ in the training set. Once the RBM of the l-th layer has been trained, the RBM of the $(l+1)$-th layer can be estimated using the samples drawn from $P(\mathbf{h}^{(l+1)}|\mathbf{h}^{(l)})$. The mean-field approximation [13] is adopted here for sampling. Then, the parameters of the estimated DBN are utilized to initialize θ_D of the DCRBM. After that, a CRBM is estimated using the $\{\mathbf{h}^{(L)},\mathbf{y}\}$ pairs in training set to initialize θ_C of the DCRBM.

(2) *DNN pre-training*

In this method, a regression DNN with L-hidden layers is first constructed to map input feature $\mathbf{x}$ toward output feature $\mathbf{y}$. The weights of the regression DNN are initialized randomly, and then optimized under minimum mean square error (MMSE) criterion using training data and back-propagation algorithm [14]. The model parameters corresponding to the L hidden layers of the regression DNN are utilized to initialize θ_D of the DCRBM. Then θ_C of the DCRBM can be pre-trained in the same way as DBN pre-training.

2.3. DCRBM-based speech synthesis

When applying the DCRBM model introduced in Section 2.2 to the spectral modeling of SPSS, the input feature $\mathbf{x}$ and output feature $\mathbf{y}$ correspond to the linguistic features and spectral features at each frame respectively. Here each vector $\mathbf{y}$ is composed of a K-dimension spectral envelope extracted by STRAIGHT [15]

together with its delta and acceleration components. At training time, a conventional HMM-based speech synthesis system using mel-cepstra as spectral features is built at first. Then, the acoustic feature sequence of each utterance in the training set is aligned towards context-dependent HMM states to get the $\{\mathbf{x}, \mathbf{y}\}$ pairs for DCRBM training. Finally, a L-hidden-layer DCRBM model is trained following the method introduced in Section 2.2. Different pre-training and fine-tuning strategies will be compared in our experiments.

At synthesis stage, the text to be synthesized is firstly converted into a sequence of linguistic features $\mathbf{x}$. Given the trained DCRBM $\theta = \{\theta_C, \theta_D\}$, the conditional distribution $p(\mathbf{y}|\mathbf{x}, \theta)$ for each frame equals to an RBM with weight matrix $\mathbf{W}$, visible bias vector $\mathbf{a} + \mathbf{A}\mathbf{h}^{(L)}$, and hidden bias vector $\mathbf{b} + \mathbf{B}\mathbf{h}^{(L)}$ according to the definition of DCRBM in Section 2.2, where $\mathbf{h}^{(L)} = \Phi(\mathbf{x}, \theta_D)$. Then, a Gibbs sampling is conducted on this RBM to generate the spectral feature vector at current frame. In our implementation, the sampling of $\mathbf{h}^{(*)}$ is achieved by comparing the conditional probability in (4) with a fixed threshold of 0.5. By setting the spectral features predicted from DCRBMs as mean vectors and the global variances calculated from training data as covariance matrices, the dynamic-feature-constrained parameter generation algorithm [3] is applied to generate the static spectral envelope sequences of an utterance. These spectral envelopes together with the F0 features predicted by conventional HMM modeling are sent into STRAIGHT vocoder to synthesize speech waveforms.

3. EXPERIMENTS

3.1. Experimental conditions

In our experiments, we used a Chinese speech corpus read by a professional female speaker of Chinese. The corpus consisted of $1,000$ sentences together with the segmental and prosodic labels. 800 sentences were selected randomly for training, 100 sentences were selected randomly for validation and the remaining 100 sentences were used as a test set. The waveforms were recorded in 16 kHz/16 bit format.

3.2. System construction

At first, a conventional HMM-based system using mel-cepstra as spectral features was built. 41-order mel-cepstra (including 0-th coefficient for frame power) were derived from the spectral envelopes given by STRAIGHT analysis at 5 ms frame shift. The F0 and spectral features consisted of static, velocity, and acceleration components. A 5-state, left-to-right-with-no-skip structure was used to train HMMs for context-dependent phones. Each HMM state was modeled by a single Gaussian distribution with diagonal covariance.

When modeling spectral envelopes, the dimension of static spectral envelopes was $K = 513$ due to the FFT length of 1024 used by STRAIGHT analysis. The spectral amplitudes at each frequency point were logarithmized and normalized to zero mean and unit variance. Then, two HMM-based systems using spectral envelopes as spectral features were built. The first system adopted a single Gaussian distribution with diagonal covariance to model the distribution of spectral envelopes at each HMM state, which was named *HMM-Baseline* in our experiment[1]. The second system adopted an RBM with 50 hidden units to model the distribution of spectral envelopes at each HMM state, which was denoted as *RBM-HMM* in our experiment. These two systems were built following [7]. All the HMM-based systems shared the same decision trees for model clustering and the same state alignment results.

Table 1. Preference scores (%) among the different systems , where N/P denotes "no preference" and p means p-value of t-test between two systems. The definition of systems can be found in Section 3.2.

DBN-CRBM	*DBN-CRBM-FT*	*RND -FT*	N/P	p
21.25	33.75	–	45.00	0.13
–	71.88	15.62	12.50	0.00
68.13	–	15.00	16.87	0.00

Then, a DNN-based system and a DMDN-based system were built and named *DNN-Baseline* and *DMDN* respectively. The input feature vector was of 568 dimensions, 562 of which were binary features for linguistic contexts and the remainders were numerical features which included relative position of frames in state and phone, and durations of state and phone. The numerical part of input features were normalized to be within $[0, 1]$ based on their minimum and maximum values in the training data. Different from the conventional DNN-based modeling methods using spectral parameters, such as mel-cepstra or line spectral pairs (LSP), we took spectral envelopes with dynamic components as output feature vectors in both *DNN-Baseline* and *DMDN* systems[2]. A network structure of 2 hidden layers with $2,048$ nodes per layer was adopted by both systems. Sigmoid activation function and linear activation function were used in the hidden layers and the output layer of both systems respectively. In *DMDN* system, the mixture number was set to 2. Approximately 80% of the silence frames were removed from the training data so as to balance the training data and to reduce the computational cost. After random initialization, the model parameters of both systems were estimated by back-propagation with a mini-batch-based stochastic gradient descent (SGD) algorithm.[3]

When building the system using our proposed method, the structure of the DNN part in DCRBM was also chosen to have 2 hidden layers and $2,048$ nodes per layer. The number of hidden nodes in the CRBM part was set to $1,024$ heuristically. The input and output features were the same as the ones used in *DNN-Baseline* system. The sigmoid activation function was also used in hidden layers. At synthesis time, 15-step Gibbs sampling was conducted on the RBM of each frame to predict spectral feature vectors as introduced in Section 2.3. The spectral envelopes generated by *DNN-Baseline* were adopted to initialize the Gibbs sampling.

In order to determine the optimal strategy of DCRBM training, the two pre-training methods introduced in Section 2.2 were first compared by a listening test. Two systems named *DBN-CRBM* and *DNN-CRBM* were built using our proposed method. In these systems, the DCRBM models were estimated by DBN pre-training

[1]The HMM-based systems using mel-cepstra and using spectral envelopes had very similar synthetic results in informal listening tests. Here, the system using spectral envelopes was adopted as the baseline system to make the type of spectral features consistent among different systems.

[2]A preference listening test between the DNN-based systems using mel-cepstra and spectral envelopes was conducted. Twenty sentences in the validation set were synthesized and eight Chinese-native listeners took part in the test. The average preference percentages of these two systems were 17% and 19% which indicates the difference between them are insignificant.

[3]We have investigated different structures for DNN, different activation functions in hidden layers, different learning rates, whether using DBN weights for pre-training, and different mixture number for DMDN. The configurations of *DNN-Baseline* and *DMDN* systems introduced here were tuned to have the best objective performance on validation set.

Table 2. Preference scores (%) among speech synthesized using the *HMM-Baseline*, *DNN-Baseline*, *RBM-HMM*, *DMDN* and *DCRBM* systems, where N/P denotes "no preference". Results of t-test show that $p < 0.05$ for all evaluated system pairs in this table.

HMM-Baseline	*RBM-HMM*	*DNN-Baseline*	*DMDN*	*DCRBM*	N/P
15.00	–	–	–	74.38	10.62
–	30.62	–	–	58.75	10.63
–	–	18.75	–	69.38	11.87
–	–	–	21.88	63.75	9.38

and DNN pre-training respectively without fine-tuning[4]. During DBN pre-training and DNN pre-training, the CRBM parts of both systems were trained for 200 epochs. The naturalness of these two systems were compared by a preference test. Twenty sentences from the validation set were synthesized using both systems. Each pair of synthetic sentences were evaluated by eight Chinese-native listeners. The results showed that the preference percentage of *DBN-CRBM* (71%) was much higher than that of *DNN-CRBM* (11%). Furthermore, we conducted 50-epoch fine-tuning after DBN pre-training and 250-epoch fine-tuning after random initialization, which produced two new systems named *DBN-CRBM-FT* and *RND-FT*. A group of preference tests were conducted among *DBN-CRBM*, *DBN-CRBM-FT*, and *RND-FT* systems. The results are shown in Table 1. Examining the preference scores between *DBN-CRBM* and *DBN-CRBM-FT* systems, we can see the positive effects of fine-tuning although the difference between these two systems is not significant. Besides, both *DBN-CRBM* and *DBN-CRBM-FT* systems had better preference scores than *RND-FT*. This demonstrated the importance of pre-training when building a DCRBM. Finally, the *DBN-CRBM-FT* system was adopted to represent our proposed method and was compared with other spectral modeling methods in next subsection.

3.3. Evaluation results and analysis

Four preference tests on naturalness were conducted to compare *DCRBM* system (i.e., the *DBN-CRBM-FT* system in previous subsection) with *HMM-Baseline*, *DNN-Baseline*, *RBM-HMM*, and *D-MDN* systems[5]. Twenty sentences from the test set were synthesized by the five systems respectively using the same duration and F0 prediction results given by *HMM-Baseline*. Eight Chinese-native listeners joined the test and the results are shown in Table 2.

From this table, we can see that the *DCRBM* system achieved significantly better naturalness than all other four systems. Comparing *DCRBM* with *DNN-Baseline* and *DMDN*, we can see the advantages of RBMs over single Gaussians and GMMs in modeling the distribution of high-dimensional spectral envelopes. *DCRBM* also outperforms *RBM-HMM*, where an RBM was estimated for each HMM state as a density model and the context dependencies were expressed by decision-tree based clustering. Our proposed method only employed a global DCRBM to model the conditional PDF of acoustic features given linguistic features. This result indicates the advantage of utilizing DCRBM as a global conditional generative model. The spectral envelopes generated by the five systems for one certain frame are illustrated in Fig. 2. We can see that the spectral envelopes generated by *DCRBM* and *RBM-HMM* have much sharper formant structures and less over-smoothing effect

[4]During pre-training and fine-tuning, the learning rates were set as 10^{-4}.

[5]Some examples of the synthetic speech using the five systems can be found at `http://home.ustc.edu.cn/~byx1030/demo-2.html`.

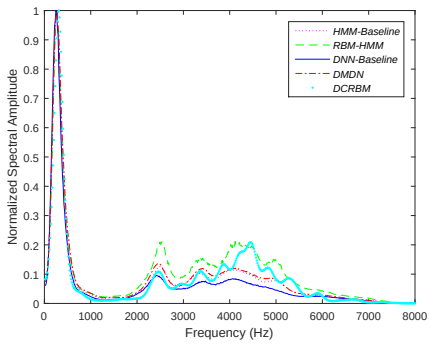


Fig. 2. Spectral envelopes generated by the five systems compared in Section 3.3 for one certain frame.

Table 3. Average spectral distortions (SD) on test set between the spectral envelopes generated by the five systems and the ones extracted from natural recordings.

system	**ave. SD (dB)**
HMM-Baseline	3.43
RBM-HMM	3.68
DNN-Baseline	3.41
DMDN	3.36
DCRBM	3.74

than the envelopes generated by the other three systems. This is consistent with the results of listening tests in Table 2.

In addition to the subjective evaluation, the spectral distortions between the spectral envelopes generated by the five systems and the ones extracted from natural recordings of test set were calculated following the method introduced in [16]. The results are shown in Table 3. We can see that the objective evaluation results are inconsistent with the subjective preference scores shown in Table 2. One possible reason is that the spectral distortion is calculated by treating each dimension of the logarithmized spectral envelopes independently and equally. However, the aim of our proposed method is to better model the multimodal property and cross-dimension correlations of the conditional distribution of acoustic features given linguistic features, which can not be expressed by this spectral distortion criterion explicitly. Similar inconsistency between subjective and objective evaluation results for speech synthesis has also been discussed in [7, 17].

4. CONCLUSIONS

This paper proposes to model the distribution of spectral envelopes given linguistic features using a DCRBM model for statistical parametric speech synthesis. A DCRBM is composed of a DNN part and a CRBM part. Its model parameters can be learnt by DBN-based or DNN-based pre-training and further fine-tuning under maximum likelihood criterion. Experimental results show that our proposed method can produce more naturally speech sounds than HMM-based, DNN-based, DMDN-based, and RBM-HMM-based synthesis methods. Future work will focus on extending the DNN and CRBM parts in DCRBMs to other forms of statistical models, such as recurrent neural networks (RNN) and neural autoregressive distribution estimators (NADE). Besides, unified modeling of spectra and F0s by DCRBM will also be a task of our future work.

5. 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. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," in *Proceedings of 6th European Conference on Speech Communication and Technology*, 1999, vol. 6, pp. 2347–2350.

[3] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. of ICASSP*, 2000, vol. 3, pp. 1315 –1318.

[4] Z.-H. Ling, L. Qin, H. Lu, Y. Gao, L.-R. Dai, R.-H. Wang, Y. Jiang, Z.-W. Zhao, J.-H. Yang, J. Chen, et al., "The USTC and iFlytek speech synthesis systems for Blizzard Challenge 2007," in *Blizzard Challenge Workshop*, 2007.

[5] H. Zen, K. Tokuda, and A. Black, "Statistical parametric speech synthesis," *Speech Commun.*, vol. 51, pp. 1039–1064, 2009.

[6] Z.-H. Ling, D. Li, and D. Yu, "Modeling spectral envelopes using restricted Boltzmann machines for statistical parametric speech synthesis," in *Proc. of ICASSP*, 2013, pp. 7825–7829.

[7] 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. Process.*, vol. 21, no. 10, pp. 2129–2139, 2013.

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

[9] H. Zen and A. Senior, "Deep mixture density network for acoustic modeling in statistical parametric speech synthesis," in *Proc. of ICASSP*, 2014, pp. 3872–3876.

[10] G.-W. Taylor, G.-E. Hinton, and S. Roweis, "Modeling human motion using binary latent variables," in *Proc. of NIPS*, 2007, pp. 1345–1352.

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

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

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

[14] D. Rumelhart, G.-E. Hinton, and R. Willams, "Learning representations by back-propagation erros," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[15] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," *Speech Communication*, vol. 27, no. 3, pp. 187–208, 1999.

[16] K.-K. Paliwal and B.-S. Atal, "Efficient vector quantization of LPC parameters at 24 bits/frame," *IEEE Trans. Speech Audio Process.*, vol. 1, no. 1, pp. 3–14, 1993.

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