DEEP BELIEF NETWORK-BASED POST-FILTERING FOR STATISTICAL PARAMETRIC SPEECH SYNTHESIS

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

National Engineering Laboratory of Speech and Language Information Processing
University of Science and Technology of China, Hefei, P.R.China
hyj15475@mail.ustc.edu.cn, {zhling,lrdai}@ustc.edu.cn

ABSTRACT

The speech synthesized by statistical parametric speech synthesis (SPSS) always sounds muffled. One important reason is that the generated spectral envelopes are over-smoothed and many detailed spectral structures in natural speech are lost. This paper presents a deep belief network (DBN)-based post-filtering method for hidden Markov model (HMM)-based SPSS to address this issue. At training time, a DBN is estimated using the spectral envelopes extracted from natural speech. This DBN serves as a generatively trained post-filter which processes the spectral envelopes recovered from the predicted spectral features at synthesis time. Experimental results show that the effectiveness of this method depends on the sampling strategy used to generate the training data of the restricted Boltzmann machines (RBM) which forms the higher layers of the DBN. When binary samples are adopted instead of mean-filed approximation, the DBN post-filter can alleviate the over-smoothing effect of parameter generation and improve the naturalness of synthetic speech significantly when either mel-cepstra or line spectral pairs (LSP) are used as spectral features. Its performance is comparative with the parameter generation method with global variance (GV) modeling for mel-cepstra and better than the LSP-based formant enhancement method used in previous work.

Index Terms— speech synthesis, hidden Markov model, post-filter, deep belief network, restricted Boltzmann machine

1. INTRODUCTION

Hidden Markov model (HMM) based statistical parametric speech synthesis (SPSS) [1] is one of the most popular methods for speech synthesis nowadays. This method is able to synthesize highly intelligible and smooth speech, and has various advantages such as compact footprint and the flexibility to control the characteristics of synthetic speech. However, this method has a tendency to over-smooth the spectral envelopes of synthetic speech because of the statistical averaging effect during HMM training and parameter generation, which makes the speech sound muffled [2].

The over-smoothing effect can be alleviated by using better acoustical models or better training criteria such as trajectory HMMs [3], deep neural networks (DNN) [4] and minimum generation error training [5]. Another other way is to compensate over-smoothing by post-filtering at synthesis time. The parameter generation algorithm considering global variance (GV) [6] is an effective one for mel-cepstral sequences. When line spectral pairs (LSPs) are used as spectral features, a method to enhance the formant structure by modifying the spaces between adjacent LSP orders has shown its effectiveness [7]. Recently, a deep learning based post-filtering method has been proposed [8]. In this method, a deep neural network (DNN) is generatively trained by concatenating two deep believe networks (DBN) and a bidirectional associative memory (BAM) to map the spectral envelopes of synthetic speech towards natural ones. One deficiency of this method is that the parameters of the post-filter depends on the parameter generation process, which means that the DNN needs to be re-trained if different spectral features or parameter generation algorithms are adopted.

This paper proposes to utilize a DBN as a post-filter to compensate the over-smoothing effect of HMM-based speech synthesis. At training time, the spectral envelopes derived from natural speech by STRAIGHT [9] are used to estimate a DBN, which is learnt in a layer-by-layer manner using a stack of RBMs [10]. Binary samples are used instead of mean-filed approximation to train the RBMs above the first layer, which aims at finding discrete patterns within training samples and proves to be essential in our experiments. At synthesis time, the DBN post-filter works like an auto-encoder [11], which first extracts high-level hidden representations from the spectral envelopes generated by HMMs and then recover the spectral envelopes through a top-down process [12]. These modified spectral envelopes are sent into a vocoder to reconstruct final speech waveforms. Because only the spectral envelopes of natural recordings are involved in training data, the DBN post-filter can work for the systems using different spectral features derived by STRAIGHT, such as mel-cepstra and LSPs.

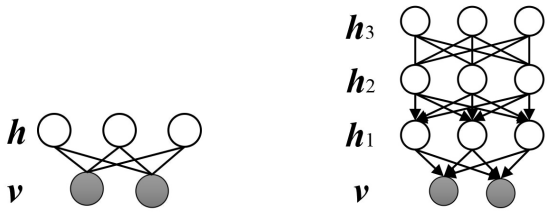
The rest of this paper is organized as follows. In Section 2, we will introduce our proposed post-filtering method after a brief review of DBNs and its building blocks RBMs. Section 3 presents the experimental results and discussion. Section 4 gives the conclusion.

2. METHODS

2.1. RBMs and DBNs

An RBM is an undirected bipartite graphical model with one layer of stochastic visible units connected to one layer of stochastic hidden units [13]. The hidden layer units of RBM is binary, while the visible layer could be either binary or Gaussian. The graphical representation of an RBM is shown in Fig.1(a). An RBM can be considered as a density model which describes the distribution of visible units. Given training samples of visible units, the model parameters of an RBM can be estimated under maximum likelihood criterion using contrastive divergence (CD) algorithm [14].

This work is 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) RBM (b) DBN

Fig. 1. The graphical model representations of an RBM and a DBN with three hidden layers. Gray circles denote visible units and white circles denote hidden units.

A DBN is a probabilistic generative model that contains many layers of hidden units [10]. The top two layers form an undirected bipartite graph with the lower layers forming a directed sigmoid belief network. Fig.1(b) shows the graphical representation of a DBN with three hidden layers.

Considering a DBN with Gaussian visible units and binary hidden units of K layers, the conditional distribution between two adjacent layers along top-down direction can be derived as

$$P(h_i^{k-1} = 1|\boldsymbol{h}^k) = g(a_i^k + \sum_j w_{ij}^k h_j^k), \tag{1}$$

where $k \in \{2, 3, ..., K\}$, $\boldsymbol{W}^k = \{w_{ij}^k\}$ and $\boldsymbol{a}^k = \{a_i^k\}$ are the weight matrix and visible bias vector at the k-th layer, and $g(x) = 1/(1 + \exp(-x))$ is the sigmoid function. At the bottom layer, we have

$$P(\boldsymbol{v} \mid \boldsymbol{h}^1) = \mathcal{N}(\boldsymbol{v}; \boldsymbol{W}^1\boldsymbol{h}^1 + \boldsymbol{a}^1, \boldsymbol{\Sigma}), \tag{2}$$

where $\mathcal{N}(\cdot)$ denotes a Gaussian distribution, and the covariance matrix $\boldsymbol{\Sigma}$ is commonly simplified to an identity matrix.

It is difficult to train a DBN directly under maximum likelihood criterion due to its complex model structure. A fast, unsupervised learning algorithm for training DBNs was proposed by training a stack of RBMs in a layer-by-layer manner [10]. Given training samples of visible units, the model parameters $\{\boldsymbol{W}^1, \boldsymbol{a}^1, \boldsymbol{b}^1\}$ of the bottom RBM are first learnt, where $\boldsymbol{W}^1$ and $\boldsymbol{a}^1$ are freezed as the parameters of the DBN. After the RBM at the $(k-1)$-th layer is learnt, the parameters of the RBM at the k-th layer can estimated using the samples drawn from

$$P(h_j^k = 1|\boldsymbol{h}^{k-1}) = g(b_j^k + \sum_i w_{ij}^k h_i^{k-1}), \tag{3}$$

where $k \in \{1, 2, ..., K\}$, and $\boldsymbol{h}^0 = \boldsymbol{v}$. Mean-field approximation is commonly used in practical implementation [15] to draw samples following (3), which means that the j-th dimension of the sampled vector is calculated as

$$\hat{h}_j^k = E\left[h_j^k|\boldsymbol{h}^{k-1}\right] = P(h_j^k = 1|\boldsymbol{h}^{k-1}). \tag{4}$$

This sample generation and RBM training process is conducted iteratively until it reaches the top layer. Then, all parameters of the DBN can be estimated.

2.2. DBN-based post-filtering

A DBN is a generative model, which generates visible units along top-down direction. The conditional distribution $P(\boldsymbol{v} \mid \boldsymbol{h}^K)$ can be calculated by cascading (1) and (2). When a DBN is estimated by a stack of RBMs, it can also extract hidden representations from visible units in a bottom-up way following (3). In previous work [8], DBNs have been adopted to achieve the transformation between spectral envelopes and hidden representations in a DNN-based post-filter for SPSS. In this paper, we investigate the method of applying a DBN alone as a post-filter to alleviate the over-smoothing effect of generated spectral features.

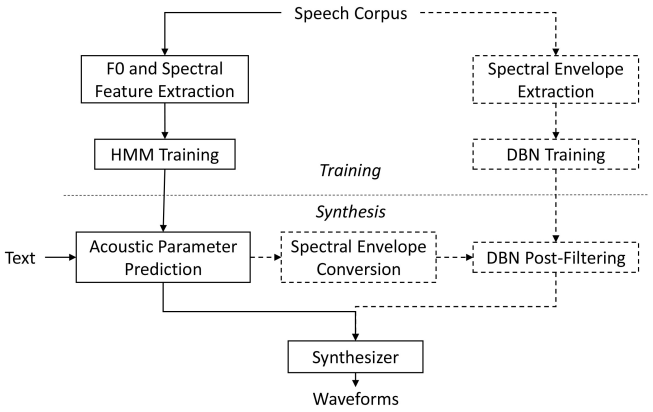


Fig. 2. Flowchart of our proposed DBN post-filtering method. The modules in solid lines represent the procedures of conventional HMM-based speech synthesis. The modules in dash lines describe the add-on procedures of our proposed method.

The flowchart of our proposed method is shown in Fig.2. STRAIGHT [9] is adopted as the vocoder for acoustic feature extraction and waveform reconstruction in our method. At the training stage, context-dependent HMMs are estimated following the conventional approach which uses mel-ceptrum or LSP as spectral feature. In the meantime, the raw spectral envelopes derived from the training corpus are used to estimate a global DBN model following the method introduced in Section 2.1. In this paper, binary samples of hidden units instead of the posterior probabilities in (4) are used to train the RBMs above the bottom layer. The samples are generated following maximum output probability criterion, i.e.,

$$\hat{h}_j^k = \arg\max_{h_j^k} P(h_j^k|\boldsymbol{h}^{k-1}). \tag{5}$$

The motivation of adopting this sampling strategy is that the estimated DBN is expected to act as a post-filter for the generated spectral features in this paper. Therefore, its aim is not to perfectly recover input features but to modify them according to the latent patterns of spectral vectors in the training set. Using binary samples may help to cluster similar vectors at lower layers for extracting representative patterns at higher layers.

At synthesis time, acoustic feature sequences are predicted from the estimated HMMs using conventional parameter generation algorithm [16]. The generated spectral features, i.e., mel-cesptra or LSPs, are first converted to spectral envelopes according to their definition. Then the spectral envelopes are sent into the trained DBN for post-filtering. Given a frame of generated spectral envelope, hidden representations are derived in a bottom-up way by applying (3) layer-by-layer. After reaching the top layer, the topmost hidden representations are used to reconstruct the visible feature in a top-down way by using (1) and (2). Because the post-filtering is conducted

Table 1. The systems using mel-cepstra.

System	Descriptions
HMM	the HMM-based baseline system using 41-dimensional mel-cepstra as spectral features
HMM-GV	the system using parameter generation with GV modeling [6]
DBN-B	the system using DBN-based post-filter, which is trained with spectral envelopes and binary samples at hidden layers
DBN-M	the system using DBN-based post-filter, which is trained with spectral envelopes and mean-field approximation at hidden layers
DBN-MCEP	the system using DBN-based post-filter, which is trained with mel-cepstra and binary samples at hidden layers

Table 2. The systems using LSPs.

System	Descriptions
HMM	the HMM-based baseline system using 41-dimensional LSPs as spectral features
HMM-FE	the system using LSP-based formant enhancement [7] for post-filtering
DBN-B	the system using DBN-based post-filter, which is trained with spectral envelopes and binary samples at hidden layers
DBN-M	the system using DBN-based post-filter, which is trained with spectral envelopes and mean-field approximation at hidden layers
DBN-LSP	the system using DBN-based post-filter, which is trained with LSPs and binary samples at hidden layers

frame-by-frame without further temporal smoothing, mean-field approximation at hidden layers is used during bottom-up and top-down mapping in order to avoid the discontinuity introduced by binarization. Finally, the post-filtered spectral envelopes are used to synthesize speech waveforms by STRAIGHT.

3. EXPERIMENTS

3.1. Experimental setup

The database of female US English speaker SLT in CMU ARCTIC databases (`http://festvox.org/cmu_arctic/`) was used in our experiments. The waveforms were recorded in 16kHz/16bit format. One thousand of the 1,132 utterances in the database were used for system training. The remaining 132 utterances were used for testing.

When constructing the baseline HMM-based speech synthesis system, 41-dimensional mel-cepstra (including a power dimension) or 41-dimensional LSPs (including a gain dimension) were derived from the spectral envelopes analyzed by STRAIGHT at 5ms frame shift. The F0 and spectral features consisted of static, velocity, and acceleration components. A 5-state left-to-right HMM structure with no skips was adopted to train the context-dependent phone models. The covariance matrix of the single Gaussian distribution at each HMM state was set to be diagonal.

The dimension of extracted spectral envelopes was 513 due to the FFT length of 1024 in STRAIGHT analysis. The spectral amplitudes at each frequency point were logarithmized. For DBN training, silence frames were removed to reduce computational cost and each dimension of training samples were normalized to zero mean and unit variance. The DBN was set to have three hidden layers and 1024 units per layer according to some preliminary and informal listening tests. The learning rate for RBM training was 0.0001. The batch size was set to 20 and 200 epochs were executed for estimating each RBM when building the DBN. CD learning with 1-step sampling (CD-1) was adopted for RBM training.

For comparison, we built another two DBNs using the mel-cepstral and LSP vectors of training data respectively. These DBNs were used as post-filters to process the generated mel-cepstra or LSPs directly. Their performance was compared with the DBN post-filter on spectral envelopes in our experiments. These two DBNs contained both three hidden layers and 82 hidden units per layer. The number of hidden units was chosen according to the proportion between the number of hidden and visible units in the DBN for spectral envelopes.

3.2. Results and discussion

Two groups of systems using mel-cepstra or LSPs as spectral features for HMM modeling were evaluated in our experiments. Details of the evaluated systems are shown in Table 1 and 2. Two MUSHRA (MUltiple Stimuli with Hidden Reference and Anchor) tests [17] were conducted to compare the naturalness scores of the systems in each group. Natural recordings were adopted as reference stimuli in each test. Ten sentences were randomly selected from the test set and were synthesized by all the systems listed in Table 1 and 2.[1] In each test, the utterances synthesized by five systems were evaluated by twenty English native listeners on the crowdsourcing platform of Amazon Mechanical Turk (`http://www.mturk.com`) with anti-cheating considerations [18]. The listeners were asked to give each utterance a score ranging from 0 to 100 by comparing the same sentence synthesized using other systems. The average naturalness scores and standard errors of all systems are shown in Fig. 3 and Fig. 4.

From the evaluation results, we can see that the GV-based parameter generation method for mel-cepstra and the LSP-based formant enhancement method improve the naturalness of synthetic speech a lot. *DBN-M* performs similar to *HMM* in both tables. Paired *t*-test shows that there is no significant difference between these two systems. The reason is that mean-field approximation was adopted by both DBN training and post-filtering in *DBN-M*. Thus, the DBN worked similar to an auto-encoder [11] considering that the RBMs were estimated by CD-1 algorithm.

In contrast, *DBN-B* performs better than *HMM* in both groups. This means that the same DBN post-filter can improve the naturalness of synthetic speech when either mel-cesptra or LSPs are used as spectral features. Comparing *DBN-B* with *DBN-M*, we can see the effectiveness of generating binary samples using (5) for training a DBN post-filter. Such binarization could be considered as a procedure of discretizing the generated hidden units and clustering their similar patterns. Patterns with high probabilities are emphasized, while patterns with low probabilities are suppressed. Therefore, the built DBN could be more sensitive to the principle patterns in natural spectral envelopes. The DBN-based post-filtering could be considered as a process which finds the latent spectral representations of

[1] Demos of synthetic speech can be found at `http://home.ustc.edu.cn/~hyj15475/DBNPost-Filtering/demo.html`.

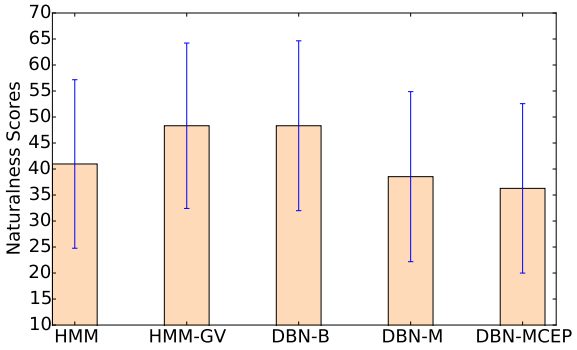


Fig. 3. Naturalness scores of systems using mel-cepstra.

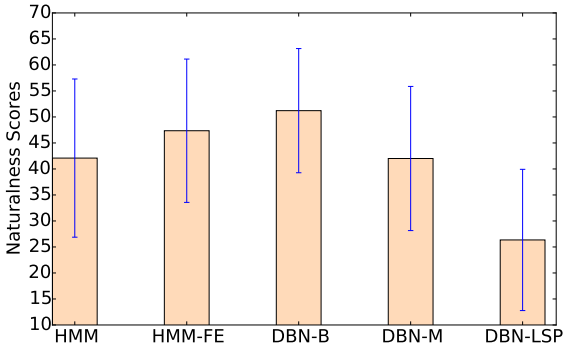


Fig. 4. Naturalness scores of systems using LSPs.

generated spectral envelopes and then recovers them using the principle patterns extracted from natural spectral envelopes.

Paired *t*-tests show that there is no significant difference between *DBN-B* and *HMM-GV* in Fig. 3. The average GV vectors of the mel-cepstra generated by the *HMM*, *HMM-GV*, and *DBN-B* systems on test set were compared with natural ones and are shown in Fig. 5. We can see that our proposed DBN-based post-filtering method can compensate the gap between the GVs of generated mel-cepstra and natural ones effectively even if GV parameters are not explicitly utilized by this method.

In Fig. 4, *DBN-B* achieves higher naturalness score than *HMM-FE*. Paired *t*-tests show that the difference between these two systems is significant at the significance level of 0.05. Fig.6 shows one frame of spectral envelopes generated by four systems using LSPs. We can see that both *HMM-FE* and *DBN-B* systems can increase the sharpness of formants effectively and the effect of DBN post-filter is more significant. This is consistent with the results of listening test shown in Fig. 4.

Finally, we can see that the *DBN-MCEP* and *DBN-LSP* systems perform worse than baseline systems. One possible reason is that the method of training DBNs using binary hidden samples may be inappropriate for the low-dimensional spectral features extracted from spectral envelopes, such as mel-cepstra and LSPs. As discussed in [8], the hidden units extracted from mel-cepstra using an RBM may not distribute in a fairly binary way. Therefore, using binary samples may introduce large distortions to the hidden representations.

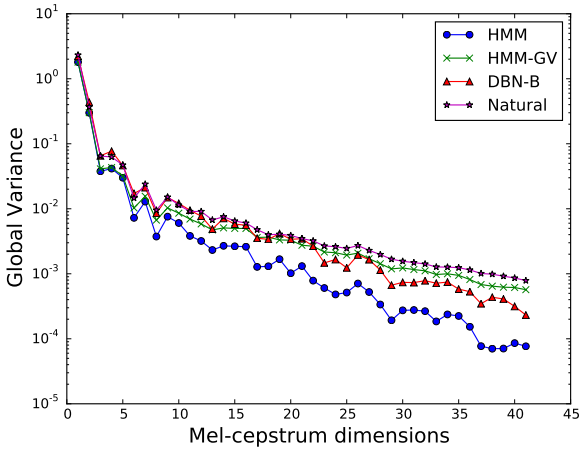


Fig. 5. Average GVs on test set of different systems in Table 1.

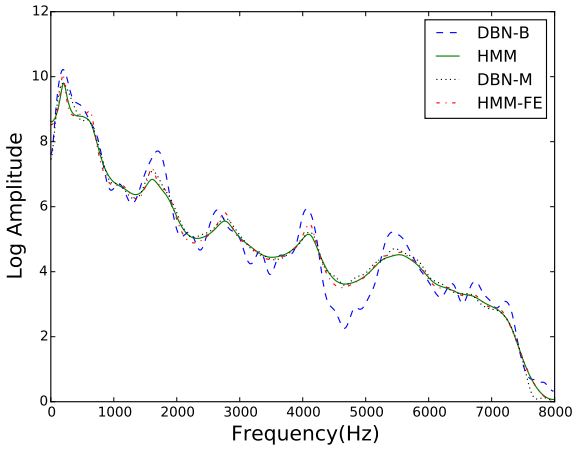


Fig. 6. Spectral envelopes generated by different systems in Table 2.

4. CONCLUSION

This paper presents a DBN-based post-filtering method to alleviating the over-smoothing effect of conventional HMM-based speech synthesis. The DBN is trained using the spectral envelopes of natural speech. A strategy of generating binary samples at hidden layers for DBN training is adopted to boost the performance of the DBN as a post-filter. At synthesis time, the spectral features predicted by HMMs are converted to spectral envelopes and then processed by the DBN through a bottom-up and top-down mapping process. Subjective results show that the built DBN post-filter can improve the naturalness of synthetic speech effectively when either mel-cepstra or LSPs are used as spectral features. It achieves equivalent performance to GV-based parameter generation for mel-cepstra, and outperforms the formant enhancement method for LSPs. Future work will focus on utilizing DBN as a feature extractor to improve the performance of HMM-based and DNN-based parametric speech synthesis.

5. REFERENCES

[1] Takayoshi Yoshimura, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," in *Proc. Eurospeech*, 1999, vol. 6, pp. 2347–2350.

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

[3] Keiichi Tokuda, Heiga Zen, and Tadashi Kitamura, "Trajectory modeling based on HMMs with the explicit relationship between static and dynamic features.," in *Proc. Eurospeech*, 2003, pp. 865–868.

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

[5] Yi-Jian Wu and Ren-Hua Wang, "Minimum generation error training for HMM-based speech synthesis," in *Proc. ICASSP*. IEEE, 2006, vol. 1, pp. 89–92.

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

[7] Zhen-Hua Ling, Yi-Jian Wu, Yu-Ping Wang, Long Qin, and Ren-Hua Wang, "USTC system for Blizzard Challenge 2006: an improved HMM-based speech synthesis method," in *Blizzard Challenge Workshop*, 2006.

[8] Ling-Hui Chen, Tuomo Raitio, Cassia Valentini-Botinhao, Zhen-Hua Ling, and Junichi Yamagishi, "A deep generative architecture for postfiltering in statistical parametric speech synthesis," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 23, no. 11, pp. 2003–2014, 2015.

[9] Hideki Kawahara, Ikuyo Masuda-Katsuse, and Alain 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. 187–207, 1999.

[10] Geoffrey Hinton, Simon Osindero, and Yee-Whye Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[11] Li Deng, Michael L Seltzer, Dong Yu, Alex Acero, Abdelrahman Mohamed, and Geoffrey E Hinton, "Binary coding of speech spectrograms using a deep auto-encoder," in *Proc. Interspeech*, 2010, pp. 1692–1695.

[12] Yoshua Bengio, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[13] Paul Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," 1986.

[14] Geoffrey Hinton, "A practical guide to training restricted Boltzmann machines," *Momentum*, vol. 9, no. 1, pp. 926, 2010.

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

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

[17] BS. 1534-1. Recommendation, ITUR, "Method for the subjective assessment of intermediate sound quality (MUSHRA)," *International Telecommunications Union, Geneva*, 2001.

[18] Sabine Buchholz and Javier Latorre, "Crowdsourcing preference tests, and how to detect cheating.," in *Proc. Interspeech*, 2011, pp. 3053–3056.