Multiple Feed-forward Deep Neural Networks for Statistical Parametric Speech Synthesis

Multiple Feed-forward Deep Neural Networks for Statistical Parametric Speech Synthesis

Shinji Takaki[1], *SangJin Kim*[2], *Junichi Yamagishi*[1,3], *JongJin Kim*[2]

[1]National Institute of Informatics (NII), Tokyo, 101-8430, Japan
[2]Naver Labs, Naver Corporation, Seongnam, 463-867, Korea
[3]The Centre for Speech Technology Research (CSTR),
University of Edinburgh, Edinburgh, EH8 9LW, United Kingdom
{takaki, jyamagis}@nii.ac.jp, {sangjin.kim, kimjj.geek}@navercorp.com

Abstract

In this paper, we investigate a combination of several feed-forward deep neural networks (DNNs) for a high-quality statistical parametric speech synthesis system. Recently, DNNs have significantly improved the performance of essential components in the statistical parametric speech synthesis, e.g. spectral feature extraction, acoustic modeling and spectral post-filter. In this paper our proposed technique combines these feed-forward DNNs so that the DNNs can perform all standard steps of the statistical speech synthesis from end to end, including the feature extraction from STRAIGHT spectral amplitudes, acoustic modeling, smooth trajectory generation and spectral post-filter. The proposed DNN-based speech synthesis system is then compared to the state-of-the-art speech synthesis systems, i.e. conventional HMM-based, DNN-based and unit selection ones.

Index Terms: Speech synthesis, DNN, Acoustic feature extraction, Acoustic modeling, Post-filtering

1. Introduction

Recently, statistical speech synthesis research has been significantly advanced thanks to deep neural networks (DNNs) with many hidden layers. For instance, DNNs have been applied for acoustic modeling. Zen et al. use a DNN to learn the relationship between input texts and extracted features instead of decision tree-based state tying[1]. Restricted Boltzmann machines or deep belief networks have been used to model output probabilities of hidden Markov model (HMM) states instead of GMMs[2]. Recurrent neural networks and long-short term memories have been used for prosody modelling[3] and acoustic trajectory modelling[4]. In addition, an auto-encoder neural network has also been used to extract low dimensional excitation parameters[5]. Furthermore a DNN-based probabilistic post-filter was also proposed[6] where a DNN is used to model the conditional probability of the spectral differences between natural and synthetic speech so that the fine spectral structure lost during modeling can be reconstructed at synthesis time.

In this paper we try to apply multiple feed-forward DNNs into several components of statistical speech synthesis systems rather than focusing on the improvement of a specific component and aim to better connect many components through neural network representations resulting in the construction of a high-quality statistical parametric speech synthesis system. More specifically we combine three types of feed-forward DNNs so that the DNNs can perform all standard steps of the statistical speech synthesis from end to end, including the feature extraction from STRAIGHT spectral amplitudes [7], acoustic modeling, smooth trajectory generation and spectral post-filter.

On the basis of this vision, we first construct a DNN that directly synthesizes high-dimensional spectral amplitudes from linguistic features without using spectral envelope parameters such as mel-cepstrum. However, it is well known that there are many problems for training a DNN such as the local optima, vanishing gradients and so on [8]. To train the DNN efficiently, we stack two DNNs, an auto-encoder neural network for data-driven non-linear feature extraction from the spectral amplitudes and another network for acoustic modeling and context clustering. We have confirmed that this training technique is effective and provides improvements in our previous experiment [9].

Although the above stacked DNN can predict spectral amplitudes from linguistic inputs frame-by-frame, we also need to consider sequential characteristics of speech for generating a smooth trajectory of spectral amplitudes. In the statistical parametric speech synthesis system, the parameter generation algorithm using time derivative features is a well known technique for synthesizing smooth trajectories [10]. There are also several new attempts to use recurrent neural networks or long short-term memories for explicitly modeling time-series data [3, 11]. There are two ways for synthesizing smooth trajectories; one is to consider time derivative features in a post processing, and the other is to have time dependency in the acoustic modeling.

In this paper we focus on a post-filter approach based on a feed-forward DNN that uses consecutive frames as inputs and outputs[6]. This DNN-based post-filter can perform the smoothing process in the time domain as well as spectral peak enhancement in frequency domain for generating natural-sounding smooth spectral amplitudes. As a result, three types of feed-forward DNNs; a feature extractor, an acoustic model and a post-filter, are used for constructing the proposed system.

The rest of this paper is organized as follows. Section 2 shows the DNN-based acoustic feature extractor. Section 3 describes the technique for constructing a DNN that directly synthesizes the spectral amplitudes. The DNN-based postfilter is shown in Section 4. The experimental conditions and results are shown in Section 5. Concluding remarks and future works are presented in Section 6.

2. Deep Auto-encoder based Acoustic Feature Extraction

An auto-encoder is an artificial neural network that is used generally for learning a compressed and distributed representation of a dataset. It consists of the encoder and the decoder. In the

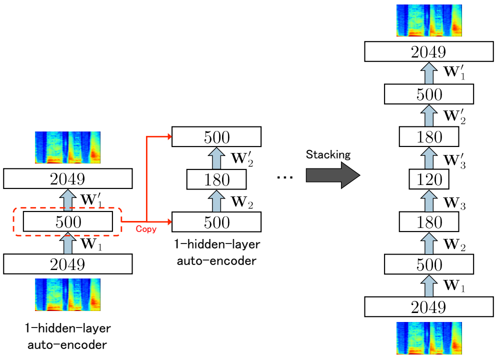

Figure 1: Greedy layer-wise pre-training for constructing a deep auto-encoder.

basic one-hidden-layer auto-encoder, the encoder maps an input vector $\mathbf{x}$ to a hidden representation $\mathbf{y}$ as follows:

$$\mathbf{y} = f_\theta(\mathbf{x}) = s(\mathbf{W}\mathbf{x} + \mathbf{b}), \tag{1}$$

where $\theta = \{\mathbf{W}, \mathbf{b}\}$. $\mathbf{W}$ and $\mathbf{b}$ represent an $m \times n$ weight matrix and a bias vector of dimensionality m, respectively, where n is the dimension of $\mathbf{x}$. The function s is a non-linear transformation on the linear mapping $\mathbf{W}\mathbf{x} + \mathbf{b}$. A sigmoid, a tanh, or a ReLU function is typically used for s. $\mathbf{y}$, the output of the encoder, is then mapped to $\mathbf{z}$, the output of the decoder. The mapping is performed by a linear mapping followed by an arbitrary function t that employs an $n \times m$ weight matrix $\mathbf{W}'$ and a bias vector of dimensionality n as follows:

$$\mathbf{z} = g_{\theta'}(\mathbf{y}) = t(\mathbf{W}'\mathbf{y} + \mathbf{b}'), \tag{2}$$

where $\theta' = \{\mathbf{W}', \mathbf{b}'\}$. An auto-encoder can be made deeper by stacking multiple layers of encoders and decoders to form a deep architecture.

Pre-training is widely used for constructing a deep auto-encoder. In pre-training, the number of layers in a deep auto-encoder increase twice compared to a deep neural network (DNN) when stacking each pre-trained unit. In this paper, we restrict the decoding weight as the transpose of the encoding weight following[12], that is, $\mathbf{W}' = \mathbf{W}^T$ where $\mathbf{W}^T$ denotes the transpose of $\mathbf{W}$. Each layer of a deep auto-encoder can be pre-trained greedily to minimize the reconstruction loss of the data locally. Figure 1 shows a procedure for constructing a deep auto-encoder using pre-training. In pre-training, a one-hidden-layer auto-encoder is trained and then the encoded output of the locally trained layer is used as the input and the output for the next layer. After all layers are pre-trained, they are stacked and fine-tuned to minimize the reconstruction error over the entire dataset using error backpropagation[13]. The mean square error (MSE) is used for the loss function of a deep auto-encoder.

3. DNN-based Acoustic Model

The DNN-based acoustic models representing the relationship between linguistic and speech features have been proposed for statistical parametric speech synthesis[1, 2, 3, 4]. One of the state-of-the-art DNN-based acoustic models[1] is briefly reviewed in this section.

Figure 2 illustrates a framework of the DNN-based acoustic model. In this framework, linguistic features obtained from a given text are mapped into speech parameters by a DNN. The input linguistic features are composed of binary answers to questions about linguistic contexts and numeric values such as the number of words in the current phrase, the position of the current syllable in the word, and durations of the current phoneme. In [1] the output speech parameters include spectral and excitation parameters and their time derivatives (dynamic features). By using pairs of input and output features obtained from training dataset, the parameters of the DNN can be trained with a stochastic gradient descent (SGD)[12]. Speech parameters can be predicted for an arbitrary text by a trained DNN using forward propagation.

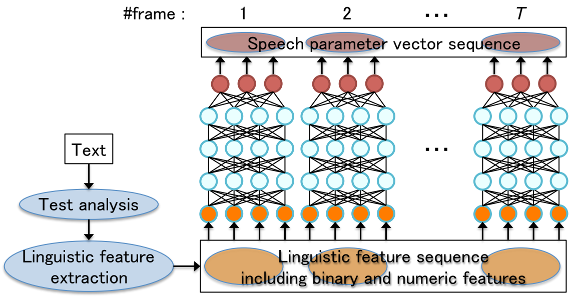
#frame :
1
2
···
T
Speech parameter vector sequence
···
Text
Test analysis
Linguistic feature extraction
Linguistic feature sequence including binary and numeric features

Figure 2: A framework of DNN-based acoustic model.

3.1. Spectral amplitude modeling using a DNN

The DNN-based acoustic model described above may be used for the direct spectral modeling by substituting the output of the network from mel-cepstrum to the spectrum. However, the dimension size of spectrum is much higher than that of mel-cepstrum. For a speech signal at 48 kHz, the mel-cepstral analysis order typically used is around 60-dim, whereas the dimension of FFT spectrum is 2049. Because of this high dimensional data, a more efficient training technique is needed to construct a DNN that directly represents the relationship between linguistic features and spectra. In this paper, we hence use a function-wise pre-training technique where we explicitly divide the general flow of the statistical parametric speech synthesis system into sub-processes, they are used to construct and optimize a DNN for each task individually, and to stack the individual networks for the final optimization.

Figure 3 shows the procedure for constructing the proposed DNN-based spectral model. Each step of the proposed technique is as follows:

Step 1. Train a deep auto-encoder using spectral amplitudes and extract bottleneck features. Layer-wise pre-training or other initialization may be used for the learning of the deep auto-encoder.

Step 2. Train a DNN-based acoustic model using the bottleneck features extracted in Step 1. Layer-wise pre-training or other initialization may be used for learning the DNN.

Step 3. Stack the trained DNN-based acoustic model for bottleneck features and the decoder part of the trained deep auto-encoder as shown in Figure 3 and optimize the whole network.

A DNN that represents the relationship between linguistic features and spectra is constructed based on a DNN-based spectral generator and a DNN-based acoustic model using the bottleneck features. After this proposed pre-training, we can fine-tune the DNN to minimize the error over the entire dataset using pairs of linguistic features and spectral amplitudes in training data with SGD.

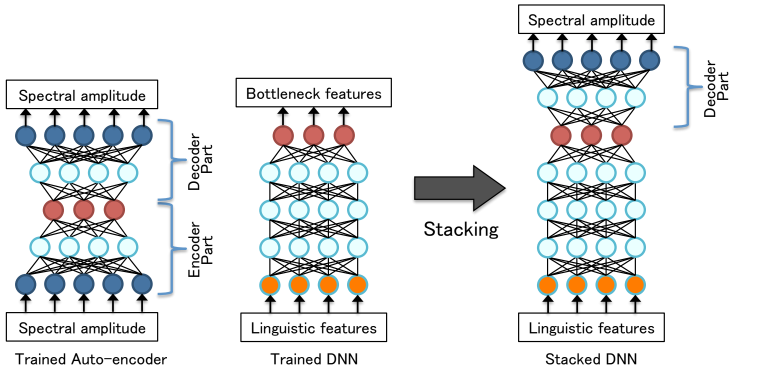

Figure 3: Procedure for constructing a DNN-based spectral model based on a deep autoencoder and a DNN-based acoustic model.

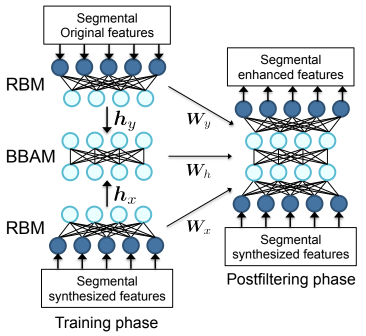
Segmental Original features
RBM
h_y
BBAM
h_x
RBM
Segmental synthesized features
Training phase
W_y
W_h
W_x
Segmental enhanced features
Segmental synthesized features
Postfiltering phase

Figure 4: A structure of a DNN for the postfilter. Training and postfiltering procedures are also illustrated.

Table 1: The detail of the English and Korean databases.

	English	Korean
Speaker	Professional female	
#Utterance (Train)	12,085	11,937
#Sentence (Test)	200	200
Sampling rate	48kHz	16kHz

4. DNN-based Post-filter for Parameter Generation in the Spectral Domain

The feed-forward DNN for probabilistic modeling of the differences between spectra of synthesized and natural speech have been proposed[6]. Figure 4 shows the structure of the DNN based post-filter. The DNN is trained layer-by-layer using two restricted Boltzmann machines (RBMs)[14] and a Bernoulli bidirectional associative memory (BBAM)[15] as shown in Figure 4. This model is directly applied to the high-dimensional spectral amplitudes.

One of the special properties of this technique is that consecutive synthesized and natural spectral amplitudes can be used as segmental inputs and outputs at each frame [6] and therefore this technique would enhance spectral amplitudes considering the differences between natural and synthetic speech in the time-frequency domain. In other words, the DNN performs spectral peak enhancements as well as spectral smoothing within the given segments in the time-frequency domain. In this paper we have further applied overlapp-add operation onto the outputs of the DNN-based post-filter instead of MLPG algorithm used in [6]. Using the overlap-added spectral amplitude sequences, we can drive the STRAIGHT vocoder to generate a synthetic speech waveform.

5. Experiments

We have evaluated the proposed technique in subjective experiments using English and Korean databases. Table 1 shows the detail of the English and Korean databases. The test sentences are not included in the training utterances.

The database provided for the Blizzard Challenge 2011[16], which contains approximately 17 hours of speech data, comprising 12K utterances, was used for the English experiment. The phoneme sequences, their boundaries and other linguistic information are automatically generated using Festival.

The Korean database is approximately 38 hours of speech data, which includes 11K utterances. The scripts are mostly news sentences including weather forecasts, traffic announcements, stock market conditions etc. The phoneme sequences, their boundaries and break strength are manually corrected.

Table 2: Spectral analysis conditions in the English and Korean experiments.

	English	Korean
FFT points (dim)	4096 (2049-dim)	2048 (1025-dim)
Cepstrum dims.	59	39

We have compared five techniques; *US* is the unit selection-based speech synthesis system, *HMM* is the HMM-based speech synthesis system with a GV technique[1][17, 18], *DNN* is the conventional single DNN-based speech synthesis system with a signal processing-based post-filter for cepstrum vectors[19], *MDNNs1* and *MDNNs2* are the proposed systems. The differences between *MDNNs1* and *MDNNs2* are F0 and aperiodicity measures. The proposed systems synthesize only spectrum features, and F0 and aperiodicity measures synthesized from *HMM* and *DNN* were used for *MDNNs1* and *MDNNs2* respectively.

Figure 5 shows network structures used in *DNN* and the proposed systems in the English experiment. We trained five-hidden-layer DNN-based acoustic models for *DNN*, *MDNNs1* and *MDNNs2*. The number of units in each of the hidden layers was set to 1024. Random initialization was used in a way similar to [1]. The symmetric five-hidden-layer auto-encoder was trained for the proposed systems. The numbers of units of the hidden layers were 2049-500-60-500-2049. As a result, we constructed and fine-tuned the eight-hidden-layer DNN for the acoustic model of the proposed systems. A two-hidden-layer DNN was trained for the spectral post-filtering in the proposed systems. Three consecutive spectral amplitudes were used as the segmental input and the output. The unit numbers of the hidden layers were 2048-2048 in the postfiltering DNN. During the overlap-add operation using the segmental outputs of the postfiltering DNN, weighting coefficients were 0.25, 0.5, 0.25 for previous, current and next frames respectively. Although in both the English and Korean experiments the settings of hidden layers were the same, the dimensions of input and output vectors were different due to differences of spectral analysis conditions. We have used a sigmoid function for all units in the hidden and output layers of all DNNs.

For each waveform, we first extract its frequency spectra using the STRAIGHT vocoder. Cepstrum vectors were extracted

[1] In this paper, the GV technique is applied to only spectral parameters.

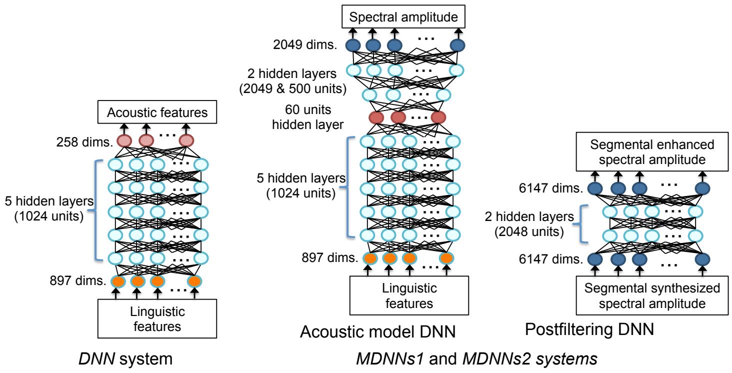

Figure 5: Structures of networks in *DNN* and the proposed systems.

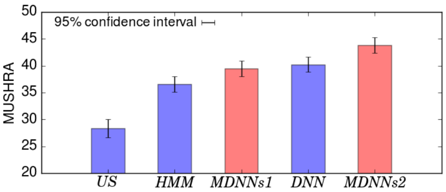

Figure 6: Subjective results (English).

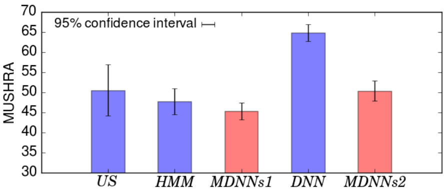

Figure 7: Subjective results (Korean).

from frequency spectra for constructing *HMM* and *DNN* systems. Spectrum and cepstrum were both frequency-warped using the Bark scale. Table 2 shows spectral analysis conditions for English and Korean datasets. Feature vectors for *HMM* and *DNN* systems were comprised of 258 and 198 dimensions for the English and Korean experiments: 59 (English) or 39 (Korean) bark-cepstral coefficients (plus the 0th coefficient), log f0, 25 dimensional band aperiodicity measures, and their dynamic and acceleration coefficients. Phoneme durations were also estimated by HMM-based speech synthesis in each language. The context-dependent labels were built using the pronunciation lexicon Combilex [20] for English and manual corrected phoneme sequences were used for Korean. The linguistic features for DNN acoustic models were comprised of 897 and 858 dimensions for English and Korean respectively. The linguistic features and spectral amplitudes in the training data were normalized for training DNNs. The input linguistic features were normalized to have zero-mean unit-variance, whereas the output spectral amplitudes were normalized to be within 0.0–1.0. We synthesized speech samples from spectrum amplitudes, F0 features and aperiodicity measures using the STRAIGHT vocoder in *HMM*, *DNN*, *MDNNs1* and *MDNNs2*. In *HMM* and *DNN*, synthesized mel-cepstral vectors were converted into spectrum amplitudes for the same STRAIGHT vocoder.

A Multisyn method[21] and NVOICE[22], which is the triphone-based in-house system of NAVER, were used to construct the English and Korean unit selection speech synthesis systems respectively.

For subjective evaluation, MUSHRA tests were conducted. Natural speech was used as a hidden top anchor reference. In the Korean experiment, a conventional HMM-based speech synthesis system without GV was used as a hidden bottom anchor. Thirty three and twenty three native subjects participated in the English and Korean experiments respectively. Fifteen sentences were randomly selected from the test set for each subject. The experiments were carried out using headphones in a quiet room.

5.1. Experimental results

Figures 6 and 7 show subjective results in the English and Korean experiments respectively. The results for reference and anchor speech were excluded from the figures to make comparison easier.

First we can see that the ratings of the unit selection system vary. As shown in figures 6 and 7, the Korean unit selection speech synthesis system is rated higher than the Korean HMM system. Whereas the English unit selection system is rated lower than the English HMM system. This would be because the Korean unit selection speech synthesis system used manually corrected phoneme alignments while the phoneme alignments were automatically estimated using HMMs in the English one.

More importantly, it can also be seen that *DNN* systems significantly outperformed the unit selection speech synthesis system in both English and Korean. In the case of Korean, although there were not many artifacts in each sample of unit concatenation as a result of the manual valuation, interestingly many test participants did not prefer *US* samples.

Comparing *DNN* and *MDNNs2* in the figure 6, we can observe that the proposed combination techniques produce more natural-sounding speech in the English experiment. This difference is statistically significant. In the Korean experiment however the proposed combination techniques produce almost the same quality of synthetic speech compared to HMM (*HMM* vs. *MDNNs1*) and less natural-sounding speech compared to DNN (*DNN* vs. *MDNNs2*) as shown in figure 7. This is the completely opposite outcome to the English findings.

We believe that this difference of the results between the English and Korean experiments may be caused by slightly different architectures due to different FFT points and sampling rate. In the proposed technique the best network structure of DNNs would strongly depend on FFT points and a high-frequency part only included in 48kHz sample rate speech, that is directly modeled and enhanced by the proposed technique. This would affect the quality of synthesized speech. Further investigation into 16kHz sample rate speech is required for the proposed technique.

6. Conclusions

In this paper, multiple feed-forward DNNs were combined to construct a high performance speech synthesis system that can deal with all of modeling of spectral amplitudes obtained from the STRAIGHT vocoder, enhancement and smoothing via neural networks. In the English experiment, the proposed combination technique was evaluated better than the conventional HMM, DNN and even unit selection systems. In future work, we will investigate the effect of the structures of the DNNs more thoroughly. Modeling F0 and aperiodicity measures in the proposed framework is also an interesting topic.

7. References

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

[2] Z.-H. Ling, L. Deng, and D. Yu, "Modeling spectral envelopes using restricted Boltzmann machines and deep belief networks for statistical parametric speech synthesis," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, pp. 2129–2139, 2013.

[3] Y. Fan, Y. Qian, F. Xie, and F. K. Soong, "TTS synthesis with bidirectional LSTM based recurrent neural networks," *Proceedings of Interspeech*, pp. 1964–1968, 2014.

[4] R. Fernandez, A. Rendel, B. Ramabhadran, and R. Hoory, "Prosody contour prediction with long short-term memory, bidirectional, deep recurrent neural networks," *Proceedings of Interspeech*, pp. 2268–2272, 2014.

[5] R. Vishnubhotla, S. Fernandez and B. Ramabhadran, "An autoencoder neural-network based low-dimensionality approach to excitation modeling for hmm-based text-to-speech," *Proceedings of ICASSP*, pp. 4614–4617, 2010.

[6] L.-H. Chen, T. Raitio, C. Valentini-Botinhao, J. Yamagishi, and Z.-H. Ling, "DNN-based stochastic postfilter for HMM-based speech synthesis," *Proceedings of Interspeech*, pp. 1954–1958, 2014.

[7] H. Kawahara, I. Masuda-Katsuse, and A. Cheveigne, "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, pp. 187–207, 1999.

[8] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," *Citeseer*, 2001.

[9] S. Takaki and J. Yamagishi, "Constructing a deep neural network based spectral model for statistical speech synthesis," *NOLISP*, 2015, (under review).

[10] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," *Proceedings of ICASSP 2000*, pp. 936–939, 2000.

[11] H. Zen, H. Sak, A. Graves, and A. Senior, "Statistical parametric speech synthesis based on recurrent neural networks," *in Poster presentation given at UKSpeech Conference*, 2014.

[12] G. E. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science 28*, vol. 313, no. 5786, pp. 504–507, 2006.

[13] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1," pp. 318–362, 1986.

[14] P. Smolensky, "Information processing in dynamical systems: foundations of harmony theory," *Parallel distributed processing: explorations in the microstructure of cognition, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA, USA: MIT Press*, vol. 1, pp. 194–281, 1986.

[15] B. Kosko, "Bidirectional associative memories," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 18, no. 1, pp. 49–60, 1988.

[16] S. King and V. Karaiskos, "The blizzard challenge 2011," 2011. [Online]. Available: http://festvox.org/blizzard/bc2011/summary_Blizzard2011.pdf

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

[18] T. Toda and K. Tokuda, "Speech parameter generation algorithm considering global variance for HMM-based speech synthesis," *Proceedings of Interspeech 2005*, pp. 2801–2804, 2005.

[19] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Incorporation of mixed excitation model and postfilter into HMM-based text-to-speech synthesis," *IEICE*, vol. J87-D-II, no. 8, pp. 1565–1571, 2004.

[20] K. Richmond, R. Clark, and S. Fitt, "On generating combilex pronunciations via morphological analysis," *Proceedings of Interspeech*, pp. 1974–1977, 2010.

[21] R. A. J. Clark, K. Richmond, and S. King, "Multisyn: Open-domain unit selection for the Festival speech synthesis system," *Speech Communication*, vol. 49, no. 4, pp. 317–330, 2007.

[22] J.-J. Kim, S.-J. Kim, S.-H. Kim, H.-J. Kim, R. Watanabe, and J.-P. Hong, "Introduction of the NAVER multi-lingual TTS system," in *Proceedings of the Acoustical Society of Korea Conference,*, vol. 32, no. 2, 2013, pp. 57–60, (in Korean).