J Sign Process Syst
DOI 10.1007/s11265-017-1293-z

Improving Deep Neural Network Based Speech Synthesis through Contextual Feature Parametrization and Multi-Task Learning

Zhengqi Wen[1] · **Kehuang Li**[2] · **Zhen Huang**[2] · **Chin-Hui Lee**[2] · **Jianhua Tao**[1,3,4]

Received: 4 February 2017 / Revised: 28 July 2017 / Accepted: 19 September 2017

Abstract We propose three techniques to improve speech synthesis based on deep neural network (DNN). First, at the DNN input we use real-valued contextual feature vector to represent phoneme identity, part of speech and pause information instead of the conventional binary vector. Second, at the DNN output layer, parameters for pitch-scaled spectrum and aperiodicity measures are estimated for constructing the excitation signal used in our baseline synthesis vocoder. Third, the bidirectional recurrent neural network architecture with long short term memory (BLSTM) units is adopted and trained with multi-task learning for DNN-based speech synthesis. Experimental results demonstrate that the quality of synthesized speech has been improved by adopting the new input vector and output parameters. The proposed BLSTM architecture for DNN is also beneficial to learning the mapping function from the input contextual feature to the speech parameters and to improve speech quality.

Keywords DNN-based speech synthesis · Vocoder · Speech parametrization · BLSTM · Phoneme embedded vector · Multi-task learning · Pitch-scaled spectrum

1 Introduction

Deep neural network (DNN) based technologies [1–6] have been used with promising results in a wide collection of research areas, such as speech processing [7–9], computer vision [10, 11] and natural language processing [12–14]. Its great ability in classification and regression has been explored not only in the research areas, but also in the real-life applications. For example, speech synthesis is a widely used technology in our real life and also catches a lot of researchers' attention. There are two typical synthesis methods in literature. One is synthesis by unit selection [15–17] and the generated waveform is concatenated from selected segments in a large speech corpus. The other is parametric speech synthesis [18–20] which estimate the related speech parameters directly from contextual features through some statistical models. In this paper, DNN is adopted as the model to predict speech parameters and called DNN-based speech synthesis [21].

There are three main components in a DNN-based speech synthesis system: input contextual features, network architecture and vocoder. New research is often focused on one or more of these three main components.

The initial work of this study was done while the first author was visiting Georgia Institute of Technology in 2014–2015.

✉ Zhengqi Wen
zqwen@nlpr.ia.ac.cn

Kehuang Li
kehle@gatech.edu

Zhen Huang
huangzhenee@gatech.edu

Chin-Hui Lee
chl@ece.gatech.edu

Jianhua Tao
jhtao@nlpr.ia.ac.cn

[1] National Laboratory of Pattern Recognition, Beijing, China

[2] School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250, USA

[3] CAS Center for Excellence in Brain Science and Intelligence Technology, Institute of Automation, Chinese Academy of Science, Beijing, China

[4] School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing, China

Published online: 02 October 2017

In the input layer of the DNN-based speech synthesis system, the contextual features include the main phoneme identity feature and other auxiliary information, such as part of speech, positional and prosodic features. Most of these contextual features are binary features suitable to construct the decision trees in the hidden Markov model (HMM)-based speech synthesis system [19], however they may be insufficient to represent the DNN's input. For the phoneme identity feature, the consequence is that the relationship between two similar phonemes might not be effectively conveyed if the phoneme identity feature is hardcoded as a binary-valued vector with a one-hot representation in the input to the neural networks. It is much more critical in natural language processing (NLP) where the dimension for one word is about tens of thousands. This problem has been alleviated with word embedding [22–26]. This model encodes a word as a real-valued low-dimensional vector based on the assumption that the semantic meanings of a word can be predicted from the external contexts with large-scale corpora. It also works for the phonemes with their pronunciation often influenced by the neighboring words and phonemes. In this paper we adopt a real-valued vector to parameterize the phoneme pronunciation. In addition to the phoneme identify feature, part of speech (POS) and pause information are also encoded as a real-valued vector from the bottleneck layer of a prediction network. After these two substitutions, all DNN inputs of DNN based speech synthesis will be real-valued vectors.

As for the network architecture, DNN here is used to construct a mapping function from the input contextual features to the output speech parameters. For example, Kang et al. used a deep belief network (DBN) to model a joint distribution of contextual and speech features in [27]. In [28], Ling et al. replaced the Gaussian mixture model (GMM) with DBN in HMM-based speech synthesis [19]. Nonetheless Zen et al. [19] proposed a deep neural network (DNN) based speech synthesis framework by mapping from the contextual features to the speech features directly and Fan et al. [29] further employed a recurrent neural network (RNN) with bi-directional long short term memory (BLSTM-RNN) [5, 6] units to model the direct mapping relationship. When compared with HMM-based speech synthesis, the DNNs are learned with little discriminative information in the output layer because the decision trees [20] used in HMM-based speech synthesis for categorizing different classes of the speech parameters have been removed from DNN-based speech synthesis. Leveraging upon recent successes in DNN-based automatic speech recognition (ASR) [7] and DNN-based automatic speech attribute transcription (ASAT) [30, 31], a key motivation in this study is to facilitate an incorporation of some categorical information in decision trees into training DNN-based speech synthesis systems. It is realized by an auxiliary categorization framework with an extra classification layer on top of the hidden layers of the regression DNNs. This classification layer is trained together with the affine-transform layer in multi-task learning (MTL) [32] which has already been used in speech synthesis in [33]. When compared with [33], this paper explored several secondary tasks in training the DNN to determine the tasks that are beneficial to improving speech quality and also applied the MTL in incorporating the vocoder.

In the vocoder, the final speech waveform is generated by the predicted speech parameters from the estimation models. In HMM-based and DNN-based speech synthesis with feed-forward network [19], the predicted speech parameters include the first and second derivatives which used in the maximum likelihood parameter generation (MLPG) algorithm [34]. While in [29], the MLPG algorithm can be removed in the BLSTM-RNN-based speech synthesis system. So in this paper we only predict the speech parameters directly. There are also other models recently proposed for vocoder. For example, Song et al. proposed an improved time-frequency trajectory excitation model in [35]. Fan et al. proposed a phase-embedded waveform representation in [36]. Hu et al. proposed to model the results of the frequency analysis in the complex domain directly in [37]. Here we propose to adopt our pitch scaled analysis (PSA) based vocoder [38] in BLSTM-RNN based speech synthesis and train an excitation model at the *phonemic* level. Because LF0 and the pitch scaled spectrum (PSS) [39] only exist in the voiced regions, two BLSTM-RNNs were trained in the proposed system. The first equipped with the multi-task learning in last paragraph is used to predict the line spectrum pair (LSP) [40] and the UV decision from the contextual features. The second is constructed to predict the log fundamental frequency (LF0), PSS and aperiodicity for the voiced phonemes with the input of the generated LSPs and contextual features. Speech is synthesized from the generated LSPs, LF0, PSS and aperiodicity parameters with the PSA-based vocoder.

The remainder of this paper is organized as follows. In Section 3, we introduce real-valued parameterization of the input contextual. In Section 4, the multi-task learning framework in DNN-based speech synthesis is described with four secondary classification tasks. In Section 5, we integrate the PSA-based vocoder into DNN-based speech synthesis with two BLSTM-RNNs. We describe our experiments in Section 6. Finally, we summarize our conclusions and propose some future work in Section 6.

2 Contextual Feature Parameterization

In conventional DNN-based speech synthesis systems, the phonemic feature is represented by a binary vector with a one-hot representation [19]. This is inefficient because the co-occurrence of phonemes is represented by a long vector with the neighboring phonemes. Vector space model (VSM) was proposed to parameterize the phonemic information as continuous values in [41, 42]. It is trained from the matrix of co-occurrence statistics and further decomposed by singular values. Our previous paper proposed to train phonemic embedded vectors (PEV) [43] in a neural network based language model (NNLM) and represent the phonemes together with word embedded vector (WEV). In this paper we enhance this representation by introducing the syllable embedded vector (SEV). The rest of this section includes two parts: one is how to train the embedded vector and the other is how to combine these embedded vectors to describe the phonemic features.

2.1 Joint Training with Embedded Vectors

There are a number of methods proposed to train the word embedded vector (WEV), such as Global C&W [44], continuous bag-of-words model (CBOW) [26] and Skip-Gram [26]. We will take CBOW in Fig. 1 as an example to describe the joint training structure.

Given a sentence with N training words, $S = \{x_1, x_2, \cdots, x_N\}$, an objective function of training CBOW is to maximize the average log probability in Eq. (1).

$$L(S) = \frac{1}{N{-}2K}\sum_{i-K+1}^{N-K} logP(x_i|x_{i-K}, \cdots, x_{i+K}) \tag{1}$$

where K is the size of the sliding window for the neighboring words. The probability $P(x_i| x_{i-K}, \cdots, x_{i+K})$ is a softmax function described in Eq. (2).

$$P(x_i|x_{i-K}, \cdots, x_{i+K}) = \frac{\exp\left(X_0^{\mathrm{T}} \cdot X_{\mathrm{i}}\right)}{\sum_{X_j \in W}\exp\left(\mathrm{X}_0^{\mathrm{T}} \cdot X_j\right)} \tag{2}$$

where W is the word vocabulary, X_{i} is the WEV of the target word x_{i}, and X_0 is the average of all neighboring context words in Eq. (3).

$$X_0 = \frac{1}{2K}\sum_{\mathrm{j=i}\text{-}K,\cdots,\mathrm{i}+K,\mathrm{j}\neq\mathrm{i}} \mathrm{X}_{\mathrm{j}}. \tag{3}$$

It is difficult to train the syllable embedded vector (SEV) or phonemic embedded vector (PEV) directly from the large corpus because the SEV or PEV takes a non-semantic meaning of a word. But it could be learned simultaneously with the WEV in a joint training structure described in [42, 45]. In this structure, the embedded vector for the context word x_{i} is changed from X_{i} to $X_{\mathrm{i}}^{\mathrm{new}}$ in Eq. (4).

$$X_{\mathrm{i}}^{\mathrm{new}} = X_{\mathrm{i}} + \frac{1}{N_{\mathrm{i}}}\sum_{\mathrm{m}=1}^{N_{\mathrm{i}}} P_{\mathrm{m}} \tag{4}$$

where $X_{\mathrm{i}}^{\mathrm{new}}$ is the composed embedded vector, X_{i} is the word embedded vector (WEV), P_{m} is the phoneme embedded vector (PEV) or syllable embedded vector (SEV), N_{i} is the number of syllable initials and finals or syllable for the ith word.

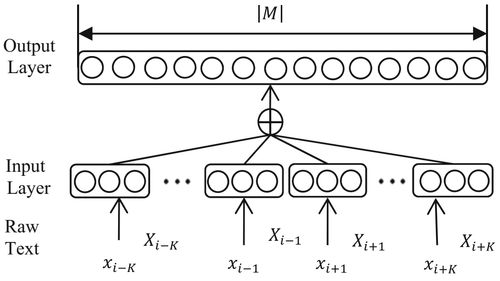

Figure 1 A block diagram of CBOW.

2.2 Combination of Embedded Vectors

The PEV and SEV are the byproducts of training the WEV and are generated in the word level. But in DNN-based speech synthesis systems, the synthesis unit is at the frame level. So the PEV, SEV and WEV should be converted into the frame level firstly. It is not easy to directly encode and a substitution is to encode these embedded vectors at the phonemic level and then to combine with positions' parameters at the frame level.

There are several ways to encode these embedded vectors into the phonemic level. This paper will adapt two ways: one is to calculate the mean of these vectors and the other is to concatenate them into one vector directly. In the first way described in Eq. (5), these three types of embedded vectors should be trained in the same dimension. In the second way described in Eq. (6), these three types of embedded vectors can be trained in the different dimension but should be in a low dimension to avoid the curse of dimensionality. Comparing experiments are conducted in Section 6 for evaluating these two combination methods.

$$X_{\mathrm{P\_new}} = \frac{1}{3}\left(X_{\mathrm{P}} + \mathrm{X}_{\mathrm{S}} + X_{\mathrm{W}}\right) \tag{5}$$

or

$$X_{\mathrm{P\_new}} = [X_{\mathrm{P}}, X_{\mathrm{S}}, X_{\mathrm{W}}] \tag{6}$$

where $X_{\mathrm{P\_new}}$ is the encoded PEV, X_{P} is the PEV, X_{S} is the SEV, X_{W} is the WEV.

3 Multi-Task Learning

3.1 Classical DNN-Based Speech Synthesis

A typical DNN based speech synthesis system shown in the left of Fig. 2 is constituted with a few hidden layers and an output layer. The hidden layers can be considered as a nonlinear feature extractor from the input contextual features. The

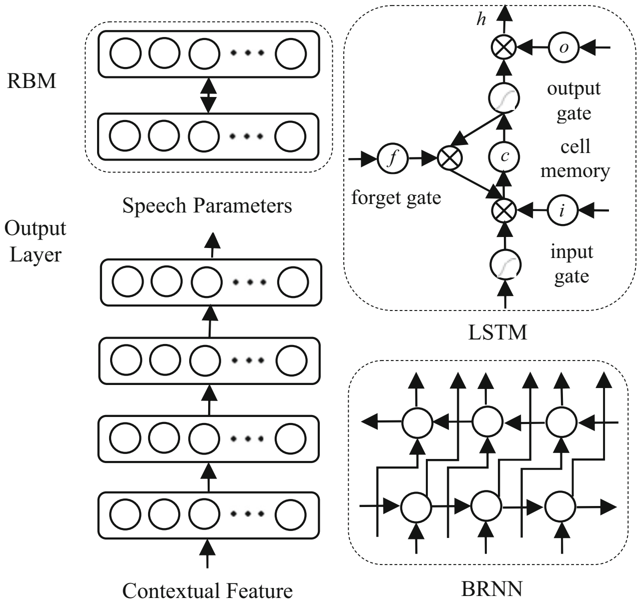

Figure 2 DNN based speech synthesis. Left: restricted Boltzmann Machine (RBM); right: long short term memory (LSTM) and bidirectional recurrent neural network (BRNN).

output layer stacked on the top of the hidden layers is an affine-transform layer for generating speech parameters from the nonlinearly transformed features. To train the DNN, the hidden layers are constituted by the pre-trained RBMs [1] with the contrastive convergence (CD) criterion [46]. The input of the first input layer is normalized as a Gaussian with zero mean and unity variance so the pre-trained hidden layers are stacked as the first Gaussian-Bernoulli RBM and the rest Bernoulli-Bernoulli RBMs.

This topology is also used in RNN based speech synthesis. In addition the hidden layers will be stacked by at least one recurrent layer, for example bidirectional recurrent neural network with long short term memory (BLSTM-RNN) units shown in the right of Fig. 2.

3.2 Proposed Classification Layer

DNN-based parameter learning for speech synthesis is often cast as a regression problem and DNN is used to construct a mapping function directly from the contextual features to the speech parameters. Thus this regression function is usually learned with little discriminative information in the output layer. To alleviate this problem, decision trees is adopted in HMM-based speech synthesis to classify the input contextual features and learn parameters. Besides this, the function will also introduce an over-smoothing problem because the generated speech parameters in the output layer are only decided by the input contextual features. To overcome this problem, Zen et al. adopted the maximum likelihood parametric generation (MLPG) algorithm [34] to get additional speech parameters, including the first and second order derivatives.

The two issues listed above could also be addressed by adding another output layer for categorization which is learned together with the affine-transform layer. The error signal of the categorization tasks will be back-propagated to update the hidden-layer parameters. Thus, the hidden layers will be learned with discriminative attributes. Moreover, this additional classification layer will also help overcoming the over-smoothing problem in discriminative learning. The proposed framework for the DNN-based speech synthesis is demonstrated in Fig. 3. A detailed description about how to learn the additional classification layer is given in the followings.

For regression, the mean square error (MSE) in Eq. (7) is minimized to fine-tune the DNN parameters:

$$D_{\mathrm{MSE}}(\hat{y},\mathrm{y}) = \frac{1}{T}\sum_{t=1}^{T}(\hat{\mathrm{y}}\text{-}\mathrm{y})^2 \tag{7}$$

where T is the total number of frames, y is the target speech feature vector and ŷ is the predicted speech feature vector as follow:

$$\hat{y} = \tilde{g}(W_{\mathrm{A}}, b_{\mathrm{A}}, h) \tag{8}$$

where $\tilde{g}$ is a linear function, W_{A} , b_{A} are the weight matrix and bias vector for the affine-transform layer, h is the output of the hidden layers.

As for classification, a soft-max layer is trained with the cross entropy (CE) criterion [47] in Eq. (9) as follow:

$$D_{CE}(\hat{\mathrm{s}},\mathrm{s}) = \sum_{\mathrm{n}=1}^{N}\sum_{\mathrm{t}=1}^{T}\mathrm{s}\log\hat{\mathrm{s}} \tag{9}$$

where N is the sentence number, T is the total number of frames, s is the target label for the categorization tasks, and ŝ is the generated label as follow:

$$\hat{s} = \frac{\exp\left(\tilde{g}(\mathrm{W_S},\mathrm{b_S},\mathrm{h})\right)}{\sum\exp\left(\tilde{g}(\mathrm{W_S},\mathrm{b_S},\mathrm{h})\right)} \tag{10}$$

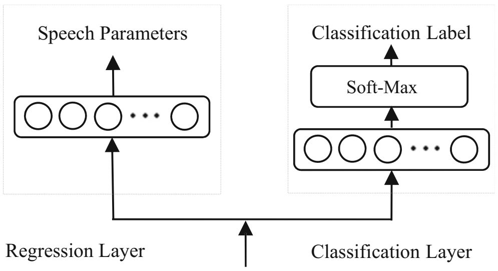

Figure 3 The framework of the output layers. Left: an affine-transform layer for generating speech parameters; right: a soft-max layer together with an affine-transform for classification.

A stochastic gradient descent (SGD) algorithm [48] is used in mini-batches to update the parameters in Eq. (11).

$$(W, b) \leftarrow (W, b) + \lambda \frac{\partial D}{\partial (W, b)} \tag{11}$$

where λ is the learning rate.

The outputs of back-propagation [49] in Eq. (7) and (8) are added together with an error ratio in Eq. (12) as the input for back-propagating to the hidden layers.

$$\mathrm{D}(\hat{y}, \mathrm{y}, \hat{s}, \mathrm{s}) = \mathrm{D}_{\mathrm{MSE}}(\hat{y}, \mathrm{y}) + \alpha \times \mathrm{D}_{\mathrm{CE}}(\hat{s}, \mathrm{s}) \tag{12}$$

where α is an error ratio.

3.3 Categorization Tasks

Decision trees facilitate a sharable structure for every state in the HMM-based speech synthesis system. It splits the set of categorical information into several nodes by asking a number of questions, such as phonemes identity, left or right contextual information and voiced/unvoiced labels. These questions help the decision trees to split the space of the speech parameters into small groups in order to learn more accurate parameters. Due to differences between the HMM and DNN, it is very hard to directly incorporate all the related questions into DNN training. Here we only consider four types of questions for constructing the classification layer.

The first is the voiced/unvoiced label. Due to the different vibrating state of glottis, the speech frames' spectra can be easily split into two groups: non-zero fundamental frequency with a harmonic structure and zero fundamental frequency with a noisy structure. This additional classification layer therefore enhances the hidden layers to describe the differences between voiced and unvoiced frames.

The second is the phone identity. In the HMM-based speech synthesis system, decision trees are constructed for every HMM state and the phone identity questions are asked in parallel with other contextual information. It means that the constructed decision trees are shared across all the phones. It is a cause of the over-smoothing problem existing in the HMM-based speech synthesis system. To alleviate this problem in DNN-based speech synthesis, we stack a phone identity classification layer on top of the hidden layers to re-enforce the phone identity's discrimination in the hidden layers.

The third is the phonation position. Every phone phonates in different positions of the vocal tract. So the phones can also be categorized into small groups. According to the knowledge in phonetics, the syllable initials and finals in Mandarin can be split into 15 groups as listed in Table 1. This layer will group the phones and learn the groups in a discriminative manner.

The fourth is the HMM state. In HMM-based speech synthesis, HMM states occupied by a number of speech frames represent a short-time stationary part of speech. So every speech frame can be categorized into a HMM state. This information can be obtained from the decision tree of the HMM-based speech synthesis system directly.

Table 1 Mandarin initials and finals based on phonation position.

labial	bilabial	p b m
	labiodental	f
coronal	dental	t d n l
	alveolar	z c s ii
velar		k g h
retroflex		zh ch sh r iii
alveolo-palatal		j q x
low	front	ai an
	central	a
	back	ang ao
middle	front	ei en
	central	eng er
	back	e o ong ou
high	front	i ia ian iang iao ie in ing iong iou v van ve vn
	back	u ua uai uan uang uei uen ueng uo

4 PSA-Based Vocoder in DNN-Based Speech Synthesis System

4.1 PSA-Based Vocoder

A pitch scaled analysis (PSA)-based vocoder [38] was used to model the residual signal as a pitch scaled spectrum (PSS) [39] and compensates the linear prediction (LP) spectrum by the detailed harmonic structure of the residual signal. It is realized by pitch scaled analysis [39] in the frequency domain.

In the analysis stage of PSA-based vocoder, the linear spectrum pairs (LSPs) [40] are first extracted for every speech frame. Then the inverse filter is constructed by LSPs to generate the residual signal. To reconstruct the residual signal in the frequency domain, PSS is defined by concatenating the peak points in the harmonic frequencies of the spectrum. An easy way to extract this envelope is by pitch-scaled analysis. Let $\mathrm{s}(k)$, $k = 1 \cdots N$ be a residual frame of two-pitch periods length and the corresponding discrete Fourier transform (DFT) of two-pitch periods length is $\mathrm{S}(n)$, $n = 1 \cdots N$. The even line of $\mathrm{S}(n)$, $n = 1 \cdots N$ in Eq. 14 which takes multiple fundamental frequencies can be indicated as PSS.

$$N = 2 \times f_s \div f_0 \tag{13}$$

$$\begin{aligned} f_k &= f_s \times k \div N = f_s \times k \div (2 \times f_s \div f_0) \\ &= f_0 \times k \div 2 \end{aligned} \tag{14}$$

where f_0, $f_s$and $f_k$are the fundamental frequency, the sampling frequency and the frequency of the kth sample.

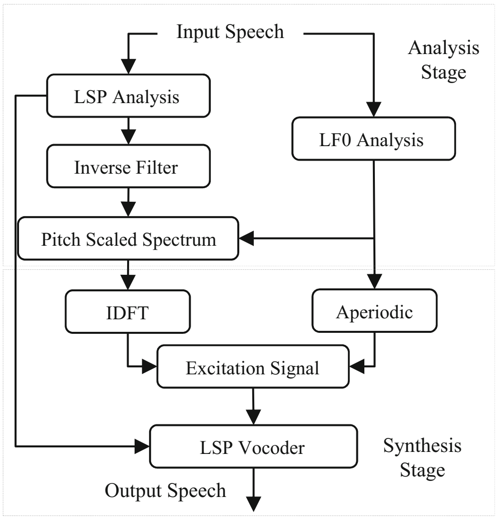

Figure 4 The workflow of a pitch scaled analysis (PSA)-based vocoder.

In the synthesis stage of the PSA-based vocoder technique, the inverse discrete Fourier transform (IDFT) with zero-phase criterion is adopted to synthesize the one pitch-cycle excitation signal. Then the periodic excitation is concatenated from these one pitch-cycle excitation signals by the overlap add (OLA) method. After mixing with the aperiodic excitation, the excitation signal is passed through a LSP vocoder to generate the speech.

A detailed workflow of PSA-based vocoder is shown in Fig. 4. The input speech is encoded by LSPs, LF0, PSS and aperiodicity for every frame. Output speech is decoded by these coefficients.

4.2 Integration into DNN-Based Speech Synthesis

The BLSTM-RNN based speech synthesis system predicts the speech parameters, unvoiced/voiced (UV) decision and LF0 directly from the input of contextual features [6]. The LF0 in the unvoiced regions is interpolated by the neighboring voiced regions. So the BLSTM-RNN is trained with the assumption that the unvoiced regions also take the continuous LF0 value. This assumption is in conflict with the input of the contextual features which takes unvoiced information, such as unvoiced phonemes. So this paper proposes to train two BLSTM-RNNs for the speech synthesis system and one of them is used to model the excitation at the phonemic level.

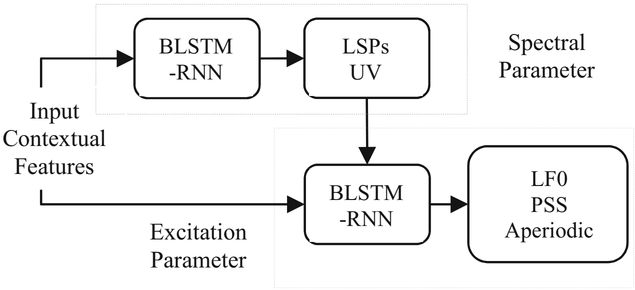

Figure 5 The framework of BLSTM-RNN based speech synthesis with the phonemic excitation.

Table 2 The LSD, LF0's RMSE and LSP's RMSE for different combination ways for PEV, SEV and WEV in DNN-based speech synthesis system.

	LSD	LF0	LSP
Meaning	3.707	0.195	1.132
Cancatenation	3.493	0.181	1.086

The framework of the proposed BLSTM-RNN based speech synthesis system is shown in Fig. 5. This system takes the contextual features as the input and generates the LSPs, LF0, PSS and aperiodicity where LSPs are generated by the first BLSTM-RNN and LF0, PSS and aperiodicity are generated by the second BLSTM-RNN. A detailed description is given below.

In the first BLSTM-RNN, the LSPs and UV decisions are predicted directly from the input contextual features. This is reasonable because the LSPs take the continuous value in the voiced and unvoiced regions and the UV decision can be made correctly based on the input phonemic information. After training the first BLSTM-RNN, the phonemes in the input can be classified into voiced or unvoiced. The phonemes with the voiced label are collected to train the second BLSTM-RNN at the phonemic level.

In the second BLSTM-RNN, the excitation parameters for the voiced phonemes are predicted which include the LF0, PSS and aperiodicity. They are all existed only in the voiced regions. The input for this network is a combination of the input contextual features and the generated LSPs. This is because the effectiveness of LSPs in predicting the LF0 has already been proved in our previous paper [50] and other researchers' work [51].

It can be concluded that the first BLSTM-RNN is used to predict the spectral parameters with continuous value and the second BLSTM-RNN is an extension of the first BLSTM-RNN and predicts the excitation parameters only in the voiced regions. It is much more reasonable than only one BLSTM-RNN predicting the spectral and excitation parameters together at any one time.

Table 3 The LSD, LF0's RMSE and LSP's RMSE for comparison between binary vector and real-valued vector of phonemes in BLSTM-RNN based speech synthesis.

	LSD	LF0	LSP
Binary-valued Vector	3.517	0.183	1.089
Real-valued Vector	3.493	0.181	1.086

Table 4 Comparing LSD, LF0's RMSE and LSP's RMSE for binary-valued and real-valued vectors in BLSTM-RNN based speech synthesis.

	LSD	LF0	LSP
Binary-valued Vector	3.517	0.183	1.089
Real-valued Vector	3.473	0.177	1.078

5 Experiments and Discussion

In this section, we first describe the experimental configuration in Section 5.1. Then the proposed three techniques are evaluated in Sections 5.2, 5.3 and 5.4, respectively.

5.1 Experiment Setup

There are two female Mandarin corpora used in the following experiments. One is used in Section 5.2 and 5.4 for about fifteen hours. The other is used in Section 5.3 for about seven hours. This is because the experiments constructed in two separately time with two different corpora. The contextual features include phoneme identity feature, part of speech, pause types, tone flags and other positional information. The speech parameters used in these experiments are line spectral pair (LSP) [40] extracted from the STRAIGHT spectrum [52], log fundamental frequency (LF0), and pitch-scaled spectrum for the PSA-based vocoder. The KALDI toolkit [53] was used for DNN training. The topology of the DNN used in the following experiments contains four hidden layers with 3072 units at each hidden layer and the RNN contains two BLSTM-RNN layers with 512 units.

The quality of the synthesized speech was verified in two ways. The first was through two objective measures, namely the root mean square error (RMSE) between the generated and the original speech parameters and log spectral distance (LSD) [54] between the generated and the original waveforms. The other was a subjective measure in terms of the ABX preference scores [55] in naturalness. In the preference tests, subjects were asked to listen to two versions of synthesized speech and choose one which sounds much better than the other. The better one will get a preference score of "1" or no preference (N/P) score of "1". The final scores were calculated by the mean value of the scores given by the 15 listeners who are working in some speech technology areas.

Table 5 Preference scores with a 0.005 confidence interval between the binary-valued and real-valued vectors in the BLSTM-RNN based speech synthesis systems.

Binary-Vector	Real-Valued Vector	N/P
0.194	0.543	0.263

Table 6 Preference scores with a 0.005 confidence interval between HMM-based and BLSTM-RNN based speech synthesis with binary-valued or real-valued vector.

HTS	Binary-Vector	Real-Valued Vector	N/P
0.263	0.501	–	0.236
0.183	–	0.602	0.215

5.2 Parameterization of Contextual Features

5.2.1 Replacing Binary Features of Phonemes

There are about 60 pronunciation initials and finals in the Mandarin language. Directly using the one-hot representation for five phonemes in the contextual features will cause the curse of dimensionality with low efficiency. But the embedded vector can be easily controlled in a low dimension. In our experiment, there are two combination ways described in Section 2.2 for these three types of embedded vector: PEV, SEV and WEV. When training the PEV, the dimension is kept as 60 as the number of pronunciation initials and finals. To simplify the representation, the SEV and WEV are also trained in the dimension of 60. So in the meaning method described in Eq. (5), the dimension for the phonemic vector is about 60; in the concatenating method described in Eq. (6), the dimension for the phonemic vector is about 180. Comparing experiment was carried out for these two combination method in DNN-based speech synthesis systems. The objective measures for the comparing results are shown in Table 2. The concatenating method gets a low objective measure than the meaning method. So in the following experiments, the concatenating method is adopted.

The difference between the binary and real-valued vector for phonemic features in DNN-based speech synthesis is compared in BLSTM-RNN based speech synthesis in Tables 3. The objective measures listed in Table 3 demonstrate that the real-valued vector is much more powerful than binary vector in describing the phonemic features.

5.2.2 Comparing with Binary Features in DNN-Based Speech Synthesis Systems

Besides phonemic feature, POS tag and pause label were used usually as binary vector in the input of the DNN-based speech

Table 7 The RMSE and LSD measures for DNN-based speech synthesis (DNN-SYN) and BLSTM-RNN-based speech synthesis (BLSTM-RNN-SYN) systems.

	LSD	LF0	LSP
DNN-SYN	4.932	0.145	1.118
BLSTM-RNN-SYN	4.896	0.138	1.112

Table 8 ABX pairwise preference scores with a 0.005 confidence interval for DNN-based speech synthesis (DNN-SYN) and BLSTM-RNN-based speech synthesis (BLSTM-RNN-SYN) systems.

DNN-SYN	BLSTM-RNN-SYN	N/P
0.1238	0.4667	0.4095

synthesis systems. This paper trained a prediction network and extracted the real vector from the bottleneck layer as the POS tag and pause label information. So the all DNN input is changed to the real-valued vector. The replacement was evaluated in BLSTM-RNN-based speech synthesis. The experimental results are listed in Tables 4 and 5.

The objective measures have been improved from the binary-valued to real-valued vector in BLSTM-RNN based speech synthesis listed in Table 4. The LSD has been reduced by about 1.25% from 3.517 to 3.473. The generated RMSEs of LF0 and LSP have been reduced by about 3.27% and 1.01%, respectively. These gains have also been confirmed in the listening test results in Table 5. Generated speech with the real-valued vector is much favored by about 35% (from 0.194 to 0.543) than the binary vector in the preference scores. These two sets of results demonstrate the effectiveness of the proposed parametrization method in DNN-based speech synthesis.

BLSTM-RNN based speech synthesis with binary-valued or real-valued vector is also compared with HMM-based speech synthesis (HMM-SSS) in subjective listening tests. The results listed in Table 6 show again the superiority of the proposed technique. Synthesized speech with the real-valued vector is much favored by about 42% than HMM-based speech synthesis (0.602 vs. 0.183) and the favored score is much larger than synthesized speech with the binary-valued vector.

5.3 Multi-Task Learning

We evaluate our proposed network architecture on two baseline systems, namely DNN-based speech synthesis (DNN-SYN) and BLSTM-RNN-based speech synthesis (BLSTM-RNN-SYN). They were trained with the same inputs to produce the desired outputs. The objective and subjective experimental results are shown in Tables 7 and 8, respectively.

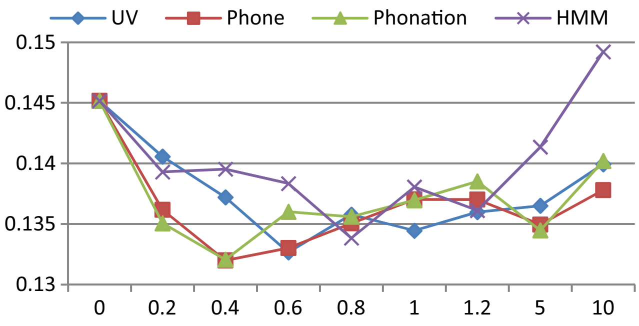

Figure 6 LF0's RMSE values as a function of error ratios for four different categorization tasks.

Table 9 A comparison of the RMSE, LSD, LSP and V/U (Voiced/Unvoiced) error for the method in [10] (UV-R-DNN) and our proposed method (UV-C-DNN) where R indicates "Regression" and C indicates "Classification".

	LSD	LF0	LSP	V/U Error
UV-R-DNN	4.983	0.153	1.120	5.449%
UV-C-DNN	4.899	0.133	1.113	3.888%

In Table 7, the objective measures were improved from DNN-based to BLSTM-RNN-based speech synthesis, especially for the LF0's RMSE which was reduced by about 4.8% (from 0.145 in the top row for DNN-SYN to 0.138 in the bottom row for BLSTM-RNN-SYN). In Table 8, the synthesized speech from BLSTM-RNN-SYN is also much preferred than from DNN-SYN.

5.3.1 Objective Measures

The error ratio α in the objective function in Eq. (12) is crucial for a proper incorporation of the classification layer part into DNN training. A series of preliminary experiments was carried out to decide which ratio is appropriate for different categorization tasks. In Fig. 6 we plot the LF0's RMSE changes with different error ratios for the four different categorization tasks. It reveals that different error ratios could be used for various tasks. The error ratios were set to be 0.6, 0.4, 0.4 and 0.8 for classifying voiced/unvoiced attribute, phone identity, phonation position, and the HMM state, respectively. These values will be used for the remaining experiments for both DNN and BLSTM-RNN based speech synthesis (Table 9).

In [30], the target voiced/unvoiced label is generated directly from the output of the regression layer. It is different from our proposed method that a classification layer is added only for the voiced/unvoiced label's classification. The objective measures are shown in Table 10 for DNN-based speech synthesis systems. The Voiced/Unvoiced error is decreased by about 1.56% from the regression-based (UV-R-DNN) to

Table 10 LSD and RMSE measures for LF0 and LSP for DNN based speech synthesis for four classification tasks.

Classification Tasks	LSD	LF0	LSP
Voiced/Unvoiced	4.899	0.133	1.112
Phoneme	4.889	0.132	1.110
Phonetic Feature	4.910	0.132	1.112
HMM State	4.917	0.134	1.113

Table 11 LSD and RMSE measures for LF0 and LSP for BLSTM-RNN based speech synthesis with four classification tasks.

Classification Tasks	LSD	LF0	LSP
Voiced/Unvoiced	4.891	0.131	1.110
Phoneme	4.885	0.132	1.108
Phonetic Feature	4.875	0.131	1.109
HMM State	4.881	0.132	1.112

classification-based (UV-C-DNN) method. When compared with Table 7, RMSE and LSD measures are also improved with the help of the classification layers, especially for RMSE of LF0 that was reduced by about 6%.

Again, three objective measures were compared with the additional classification layer for DNN and BLSTM-RNN based speech synthesis in Tables 10 and 11, respectively. It can be seen that these values vary little among themselves for the four different categorization tasks. Clearly the results with BLSTM-RNN in Table 11 are slightly better than those with DNN in Table 10, and they are all better than the baseline speech synthesis systems without the classification layers as shown in Table 7.

5.3.2 Subjective Preference Scores

Listening tests were also carried out to evaluate the proposed technique. Table 12 lists the preference scores for the voiced/unvoiced label used in DNN-based speech synthesis at a regression or a classification output layer. The score of 37.78% for the classification based method (UV-C-DNN) in the middle column of the bottom row in Table 12 is preferred to the score of at 8.89% in the regression based method (UV-R-DNN) in the left column.

Since the result differences between the four tasks in Tables 10 and 11 are very small, we only consider the voiced/unvoiced attribute in Table 13 with the best error ratio α set at 0.6. The preference scores are compared for DNN based speech synthesis with (UV-C-DNN and UV-C-RNN) and without (just plain DNN and BLSTM-RNN) the classification layer. The results again confirm that speech generated with the classification layer is much preferred to speech synthesis without the classification layer by about 24% (from 0.227 to 0.467) for DNN-based speech synthesis at the top row in Table 13, and by about 15% (from 0.187 to 0.34) for BLSTM-RNN based speech synthesis in the bottom row of Table 13.

Table 12 Preference scores with a 0.05 confidence interval for DNN based speech synthesis with regression (UV-R-DNN) or classification (UV-C-RNN) for voiced/unvoiced label.

UV-R-DNN	UV-C-RNN	N/P
0.0889	0.3778	0.5333

Table 13 Preference scores at a 0.05 confidence interval for DNN and RNN based synthesis with and without classification.

DNN	UV-C-DNN	RNN	UV-C-RNN	N/P
0.227	0.467	–	–	0.306
–	–	0.187	0.34	0.473

From these results, it could be concluded that by adding the classification layer on top of the hidden layers to the regression DNNs we could strengthen the DNN's modeling ability to generate better speech parameters from the contextual features. Among these four categorization tasks, there are no sharp differences between these tasks. But considering the convenience of extracting voiced/unvoiced attribute, the categorization task used in the multi-task learning structure in following experiments are voiced/unvoiced attribute.

5.4 PSA-Based Vocoder in DNN-Based Synthesis

To evaluate the proposed technique in the BLSTM-RNN based speech synthesis systems, a series of experiments was carried out. First, the proposed PSA-based vocoder was validated in DNN-based speech synthesis with one BLSTM-RNN and with two BLSTM-RNNs. Finally, the combination of PSA-based vocoder and two BLSTM-RNNs was further validated in speech synthesis.

5.4.1 PSA-Based Vocoder with one BLSTM-RNN

The baseline system constructed in this experiment is a DNN-based speech synthesis system with one BLSTM-RNN which was trained with a conventional LSP-based vocoder (Conv-LSP). There are two additional types of parameters in our pro-

Table 14 The LSD, LF0's RMSE and LSPs' RMSE for conventional LSP-vocoder (Conv-LSP) and PSA-based vocoder (PSA-LSP) in DNN-based speech synthesis.

	LSD	LF0	LSP
Conv-LSP	3.517	0.183	1.089
PSA-LSP	3.504	0.192	1.102

Table 15 Preference scores with a 0.005 confidence interval between conventional LSP-vocoder (Conv-LSP) and PSA-based vocoder (PSA-LSP) in DNN-based speech synthesis.

Conv-LSP	PAS-LSP	N/P
0.23	0.35	0.42

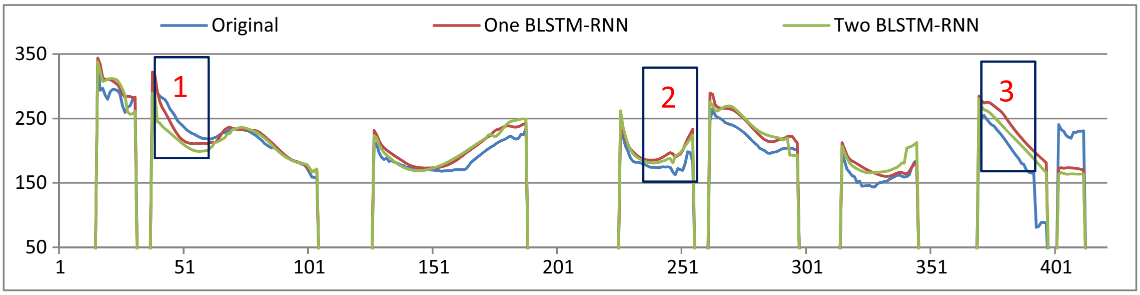

Figure 7 The pitch contour for the original sentence, generated by one BLSTM-RNN and generated by two BLSTM-RNNs.

posed PSA-based vocoder (PSA-LSP) when compared with a traditional LSP-based vocoder. They are the pitch-scaled spectrum (PSS) and the aperiodic measure (AP) both of which exist only in the voiced region. So, firstly, these two features will be interpolated with a continuous value in the unvoiced region. After that, the interpolated features will concatenate with LSPs, UV decisions and LF0s for every frame to construct the output of BLSTM-RNN. The input of BLSTM-RNN is kept the same as the baseline system. Finally, two BLSTM-RNN-based speech synthesis systems are constructed with a different output layers and two versions of synthesized speech were generated. The compared results are shown in Tables 14 and 15.

The objective measures are shown in Table 14 for these two speech synthesis systems. The LSD measure has been improved a little by the PSA-based vocoder when compared with the conventional LSP-based vocoder. But the RMSEs of LF0 and LSP were slightly increased. One possible reason is that more parameters have to be predicted in the output layer. But the subjective preference listening test has confirmed the effectiveness of PSA-based vocoder in DNN-based speech synthesis and that generated speech is more favored by about 12% (from 0.23 to 0.35) than conventional LSP-based vocoder.

5.4.2 Comparison of one and two BLSTM-RNNs

Next, two DNN-based speech synthesis systems are constructed: one with one BLSTM-RNN and the other with two BLSTM-RNNs. The vocoder used in both cases is the conventional LSP-based vocoder. The input and output for these two systems is kept the same. The only difference is to predict the speech parameters in one or two BLSTM-RNN. Fig. 7 shows an example of pitch contours for the original sentence, generated by one and two BLSTM-RNNs. It can be found that the pitch contour generated by two BLSTM-RNNs is much closer to that of the original sentence than that with one BLSTM-RNN. For example, in regions 1 and 3 in Fig. 7 the same slope as the original sentence is kept and in region 2 the same details of the original sentence is preserved.

Table 16 The LSD, RMSE for LF0 and RMSE for LSP for DNN based speech synthesis with one or two BLSTM-RNNs.

	LSD	LF0	LSP
One-BLSTM-RNN	3.517	0.183	1.089
Two-BLSTM-RNN	3.502	**0.157**	1.087

Highlighted data indicates the proposed technique

The difference of the pitch contour in Fig. 7 has been confirmed in the objective measures for generated speech in Table 16. The RMSE of generated LF0 has reduced by about 14.2% from one (at 0.183) to two BLSTM-RNNs (at 0.157) especially in DNN-based speech synthesis. This means that predicting the pitch contour in two steps at the phonemic level is more effective than in one step at the sentence level. Finally, the results of the subjective listening test described in Table 17 confirm again the superiority of the proposed structure with two BLSTM-RNNs.

5.4.3 Combination of PSA-Vocoder and two BLSTM-RNNs in DNN-Based Speech Synthesis

In the following we evaluate DNN-based speech synthesis by combining the PSA-vocoder and two BLSTM-RNNs. Synthesized speech is compared with that generated from HMM-based speech synthesis (HTS) and DNN-based based speech synthesis (DNN-SSS) both with the traditional LSP-based vocoder.

Table 17 Preference scores with 0.005 confidence interval for DNN-based speech synthesis with one or two BLSTM-RNNs.

One-BLSTM-RNN	Two-BLSTM-RNN	N/P
0.213	0.483	0.304

Table 18 The LSD, LF0's RMSE and LSP's RMSE for HMM-based, DNN-based with DNN-SSS and DNN-based speech synthesis with two BLSTM-RNNs and PSA-vocoder.

	LSD	LF0	LSP
HTS	3.831	0.176	1.088
DNN-SSS	3.912	0.189	1.098
TWO-BLSTM-RNN	**3.470**	**0.162**	**1.088**

Highlighted data indicates the proposed technique

Table 19 Preference scores with a 0.005 confidence interval for HMM-based and DNN-based speech synthesis with two BLSTM-RNNs, and for DNN-based speech synthesis with DNN and with two BLSTM-RNNs and PSA-vocoder.

HTS	DNN-SSS	Two-BLSTM-RNN	N/P
0.183	–	0.453	0.363
–	0.150	0.460	0.390

The results listed in Table 18 show that the objective measures have been improved when compared with HTS and DNN-SSS. For example, the LSD was reduced by about 9% and the LF0's RMSE has reduced by about 8% compared with HTS. Meanwhile, the results also show that HTS takes a lower objective measure than DNN-SSS. One possible reason for this is that the speech parameters in DNN-SSS are predicted directly by a local region of the input space which is different from the decision tree used in HTS. The subjective listening tests in Table 19 also indicate that generated speech from TWO-BLSTM-RNNs is much more favored by about 27% (from 0.183 to 0.453 in the top row) than that from HTS and by about 31% (from 0.15 to 0.46) than that from DNN-SSS.

6 Conclusion and Future Work

This paper proposed three techniques to improve the quality of synthesized speech in DNN-based synthesis. The first one is to parameterize the contextual features as a real-valued vector at the input of DNN-based speech synthesis model. We propose to encode the phoneme identity features as a real-valued vector in training the word embedded vector and to extract the POS and pause from the bottleneck layer of a prediction network. The second is to add an auxiliary categorization framework through multi-task learning for training DNN-based speech synthesis systems. Four types of secondary tasks have been considered in constructing the output layer. The third is to improve vocoder based on pitch-scaled analysis (PSA) in DNN-based speech synthesis. Three corresponding sets of experiments have been conducted to evaluate the proposed techniques. The experimental results demonstrate the superiority of the three proposed techniques when compared with the baseline systems.

With newly emerging DNN modeling techniques DNN-based speech synthesis still has rooms to improve. For example, parameterization directly from the input text for the contextual features still has a long way to go. Even though Bi-directional recurrent neural network with long short term memory (BLSTM-RNN) units has been verified as a good architecture for speech synthesis, it is still not easy to apply it directly into real-time applications for the burden of computation. So our future work will focus on these two aspects to improve DNN-based speech synthesis.

Acknowledgements This work is supported by the National High-Tech Research and Development Program of China (863 Program) (No. 2015AA016305), the National Natural Science Foundation of China (NSFC) (No.61403386), the Strategic Priority Research Program of the CAS (Grant XDB02080006) and partly supported by the Major Program for the National Social Science Fund of China (13&ZD189).

References

1. Hinton, G., Osindero, S., & Teh, Y. (2006). A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation, 18*, 1527–1554.
2. Hinton, G.-E. (2007). Learning multiple layers of representation. *Trends in Cognitive Sciences, 11*, 428–434.
3. LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., & Jackel, L. (1990) Handwritten digit recognition with a back-propagation network. In Advances in Neural Information Processing Systems (NIPS), pp. 396–404.
4. LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation, 1*(4), 541–551.
5. Mike, S., & Paliwal, K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing, 45*(11), 2673–2681.
6. Sepp, H., & Jürgen, S. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.
7. Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., & Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine, 29*(6), 82–97.
8. Graves, A., Mohamed, A., Hinton, G. (2013). Speech Recognition with Deep Recurrent Neural Network. In Proc. of ICASSP, pp. 6645–6649.
9. Abdel-Hamid, O., Mohamed, A., Jiang, H., Deng, L., Penn, G., & Yu, D. (2014). Convolutional Neural Networks for Speech Recognition. *In IEEE/ACM Trans. on Audio, Speech and Language Processing, 22*(10), 1533–1545.
10. Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Proc. of NIPS, 1*(2), 1097–1105.
11. K.M. He, X.Y. Zhang, S.Q. Ren and J. Sun, Deep Residual Learning for Image Recognition. In Proc. of CVPR, pp. 770–778, 2015.
12. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural Language Processing (Almost) from Scratch. *Journal of Machind Learning Research, 12*, 2493–2537.
13. Kim, Y. (2014) Convolutional Neural Networks for Sentence Classification. In: Proc. of EMNLP, pp. 1746–1751.
14. Cho, K., Merrienboer, B., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. (2014) Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In: Proc. of EMNLP.
15. Hunt, A. J., Black, A. W. (1996) Unit selection in a concatenative speech synthesis system using a large speech database. In Proc. of ICASSP, pp. 373–376.

16. H. Kawai, T. Toda, J. Ni, et al. (2004) XIMERA: A new TTS from ATR based on corpus-based technologies. In Proc. of Fifth ISCA Workshop on Speech Synthesis.
17. Ling, Z. H., Wang, R. H. (2007) HMM-based hierarchical unit selection combining Kullback-Leibler divergence with likelihood criterion. In Proc. of ICASSP, pp. 1245–1248.
18. Black, A. W., Zen, H., & Tokuda, K. (2007). Statistical parametric speech synthesis. *Proc. ICASSP, 4*, 1229–1232.
19. Zen, H., Tokuda, K., & Black, A. W. (2009). Statistical Parametric Speech Synthesis. *Speech Communication, 51*(11), 1039–1064.
20. T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura (1999) Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis. In Proc. of Eurospeech, pp. 2347–2350.
21. Ling, Z. H., Kang, S. Y., Zen, H., Senior, A., Schuster, M., Qian, X. J., Meng, H., & Deng, L. (2015). Deep Learning for Acoustic Modeling in Parametric Speech Generation. *Journal of IEEE Signal Processing Magazine, 32*, 35–52.
22. Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A Neural probabilistic language model. *Journal of Machine Learning Research*, 1137–1155.
23. Mikolov, T., Karafiat, M., Burget, L. (2010) J. "Honza" Cernocky and S. Khudanpur, "Recurrent neural network based language model. In Proc. of INTERSPEECH, pp. 1045–1048.
24. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research, 12*, 2493–2537.
25. Huang, E. H., Socher, R., Manning, C. D., & Ng, A. Y. (2012). Improving word representations via global context and multiple word prototypes. *Proc. of ACL, 1*, 873–882.
26. T. Mikolov, K. Chen, G. Corrado and J. Dean (2013) Efficient estimation of word representations in vector space. In Proc. of CoRR.
27. Kang, S., Qian, X., & Meng, H. (2013). Multi-distribution deep belief network for speech synthesis. In Proc. of ICASSP, pp.7962–7966.
28. Ling, Z.-H., Deng, L., & Yu, D. (2013) Modeling spectral envelopes using restricted Boltzmann machines for statistical parametric speech synthesis. In Proc. of ICASSP, pp. 7825–7829.
29. Fan, Y.-C., Qian, Y., Xie, F.-L. & Soong, F. K. (2014) TTS Synthesis with Bidirectional LSTM based Recurrent Neural Networks. In Proc. of Interspeech, pp.1964–1968.
30. Siniscalchi, S. M., Yu, D., Deng, L., & Lee, C.-H. (2013). Exploiting Deep Neural Networks for Detection-Based Speech Recognition. *Neurocomputing, 106*, 148–157.
31. C.-H. Lee and S. M. Siniscalchi, "An Information-Extraction Approach to Speech Processing: Analysis, Detection, Verification and Recognition," Proceedings of the IEEE, Vol. 101, No. 5, pp. 1089–1115, May 2013.
32. Caruana, R. (1997). Multitask learning. *Machine Learning Journal, 28*, 41–75.
33. Wu, Z., Valentini-Botinhao, C., Watts, O., & King, S. (2015). Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis. In Proc. of ICASSP, pp. 4460–4464.
34. Tokuda, K., Kobayashi, T., & Imai, S. (1995). Speech parameter generation from HMM using dynamic features. Proc. of ICASSP, pp. 660–663.
35. Song, E., Joo, Y.-S., & Kang, H.-G. (2015) Improved Time-Frequency Trajectory Excitation Modeling for a Statistical Parametric Speech Synthesis System. In Proc. of ICASSP.
36. Fan, B., Lee, S.-W., Tian, X.-H., Xie, L., & Dong, M.-H. (2015). A Waveform Representation Framework for High-Qaulity Statistical Parametric Speech Synthesis. In Proc. of APASIPA.
37. Hu, Q., Yamagishi, J., Richmond, K., Subramanian K., & Stylianou, Y. (2016) Initial Investigation of Speech Synthesis based on Complex-Valued Neural Networks. In Proc. of ICASSP, pp. 5630–5634.
38. Wen, Z. Q., Kawahara, H., & Tao, J. H., (2012) Pitch-Scaled Analysis based Residual Reconstruction for Speech Analysis and Synthesis. In Proc. of INTERSPEECH, pp. 374–377.
39. Jackson, P. J. B., & Shadle, C. H. (2001). Pitch-Scaled Estimation of Simultaneous Voiced and Trubulence-Noise Components in Speech. *IEEE Trans. On Speech Audio Processing, 9*(7), 713–726.
40. Soong, F.-K., & Juang, B.-H. (1984). Line spectrum pair (UP) and speech data compression. *Proc. of ICASSP, San Diego, 1*, 1.10.1–1.10.4.
41. Watts, O. (2013). Unsupervised learning for text-to-speech synthesis. PhD dissertation.
42. Chen, X., Xu, L., Liu, Z., Sun, M., & Luan, H. (2015). Joint learning of character and word embeddings. In International Joint Conference on Artificial Intelligence.
43. Wen, Z. Q., Li, Y., & Tao, J. H. (2016). The Parameterized Phoneme Identity Feature as a Continuous Real-Valued Vector for Neural Network based Speech Synthesis. In Proc. of INTERSPEECH.
44. Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In International Conference on Machine Learning.
45. Sun, F., Guo, J., Lan, Y., Xu, J., & Cheng, X. (2016). Inside out: Two jointly predictive models for word representations and phrase representations. In Proceedings of the 30th AAAI conference.
46. Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation, 14*, 1771–1800.
47. Bourlard, H., & Morgan, N. (1994). *Connectionist speech recognition*. Dordrecht: Kluwer Academic Publishers.
48. Bertsekas, D. P. (1999). *Nonlinear Programming* (2nd ed.). Belmont: Athena Scientific.
49. Rumelhart, D.-E., Hinton, G.-E., & Williams, R.-J. (1986). Learning representations by back-propagating errors. *Nature, 323*(6088), 533–536.
50. Zheng, Y. B., Wen, Z. Q., Liu, B., Li, Y. & Tao, J. H. (2016). An Initial Research Towards Accurate Pitch Extraction for Speech Synthesis based on Bidirectional Long Short-Term Memory Recurrent Neural Network. In Proc. of ICSP.
51. Su, H., Zhang, H., Zhang, X. L., & Gao, G. G. (2016). Convolutional Neural Network for Robust Pitch Determination. In Proc. of ICASSP, pp. 579–583.
52. Kawahara, H., Masuda-Katsuse, I., & de Cheveigné, A. (1999). 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, 27*(5), 187–207.
53. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., & Vesely, K. (2011) The Kaldi Speech Recognition Toolkit. In Proc. IEEE Workshop on Automatic Speech Recognition and Understanding.
54. Ephraim, Y., & Malah, D. (1985). Speech Enhancement using a Minimum Mean-Square Error Log-Spectral Amplitude Estimator. *IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSp-33*(2), 443–445.
55. Blin, L., Boeffard, O. & Barreaud, V. (2008). WEB-based listening test system for speech synthesis and speech conversion evaluation. In Proc. of LREC (Marrakech (Morocco)).

Zhengqi Wen received his B.S. degree from University Of Science and Technology of China (USTC), Heifei, in 2008 and received his the Doctor degree from Chinese Academy of Sciences (CAS), Beijing, in 2013. He is currently Associate Researcher in the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing. He current research interests include speech recognition and speech synthesis.

Kehuang Li received the B.S. degree in information engineering, and the M.S. degree in communication and information system from Shanghai Jiao Tong University. He also got an MS degree in Electrical and Computer Engineering from Georgia Institute of Technology. He is currently a Ph.D. candidate at the Georgia Institute of Technology. His research interests include signal processing, machine learning, and speech recognition.

Zhen Huang received the B.S. degree in Electrical& Computer Engineering from Southeast University, Nanjing, China in 2009. He received the dual M.S. degree in Electrical&Computer Engineering from Shanghai JiaoTong University and Georgia Institute of Technology in 2012. His research interests lie in the areas of speech recognition, deep learning, general machine learning, multimedia information retrieval and image processing, currently more focus on deep learning based speech recognition and adaptation.

Chin-Hui Lee is a Professor in the School of Electrical, and Computer Engineering, Georgia Institute of Technology. Before joining academia in 2001 he had 20 years of industrial experience ending in Bell Laboratories, Murray Hill, New Jersey, as a Distinguished Member of Technical Staff, and Director of the Dialogue Systems Research Department. Dr. Lee is a Fellow of the IEEE, and a Fellow of ISCA. He has published over 450 papers, and 30 patents, and was highly cited close to 30,000 times for his original contributions with an h-index of 65 on Google Scholar. He received numerous awards, including the Bell Labs President's Gold Award in 1998. He won the IEEE Signal Processing Society's 2006 Technical Achievement Award for "Exceptional Contributions to the Field of Automatic Speech Recognition". In 2012, he was invited by ICASSP to give a plenary talk on the future of speech recognition. In the same year he was awarded the International Speech Communication Association Medal in scientific achievement for "pioneering and seminal contributions to the principles and practice of automatic speech and speaker recognition".

Jianhua Tao received his PhD from Tsinghua University in 2001, and got his M.S. from Nanjing University in 1996. He is currently a Professor in NLPR, Institute of Automation, Chinese Academy of Sciences. His current research interests include speech synthesis and coding methods, human computer interaction, multimedia information processing and pattern recognition. He has published more than eighty papers on major journals and proceedings including IEEE Trans. on ASLP, and got several awards from the important conferences, such as Eurospeech, NCMMSC, etc. He serves as the chair or program committee member for several major conferences, including ICPR, ACII, ICMI, ISCSLP, NCMMSC etc. He also serves as the steering committee member for IEEE Transactions on Affective Computing, associate editor for Journal on Multimodal User Interface and International Journal on Synthetic Emotions, Deputy Editor-in-chief for Chinese Journal of Phonetics.