Interspeech 2018
2-6 September 2018, Hyderabad

Acoustic Modeling using Adversarially Trained Variational Recurrent Neural Network for Speech Synthesis

Joun Yeop Lee[1], *Sung Jun Cheon*[1], *Byoung Jin Choi*[1], *Nam Soo Kim*[1], *and Eunwoo Song*[2]

[1]Department of Electrical and Computer Engineering and INMC,
Seoul National University, Korea
[2]NAVER Corp., Seongnam, Korea

{jylee, sjcheon, bjchoi}@hi.snu.ac.kr, nkim@snu.ac.kr, eunwoo.song@navercorp.com

Abstract

In this paper, we propose a variational recurrent neural network (VRNN) based method for modeling and generating speech parameter sequences. In recent years, the performance of speech synthesis systems has been improved over conventional techniques thanks to deep learning-based acoustic models. Among the popular deep learning techniques, recurrent neural networks (RNNs) has been successful in modeling time-dependent sequential data efficiently. However, due to the deterministic nature of RNNs prediction, such models do not reflect the full complexity of highly structured data, like natural speech. In this regard, we propose adversarially trained variational recurrent neural network (AdVRNN) which use VRNN to better represent the variability of natural speech for acoustic modeling in speech synthesis. Also, we apply adversarial learning scheme in training AdVRNN to overcome oversmoothing problem. We conducted comparative experiments for the proposed VRNN with the conventional gated recurrent unit which is one of RNNs, for speech synthesis system. It is shown that the proposed AdVRNN based method performed better than the conventional GRU technique.

Index Terms: speech synthesis, variational recurrent neural network, adversarial learning, acoustic modeling, AdVRNN

1. Introduction

With recent advancement in applications for human-machine interaction, speech synthesis system with advanced performance became an essential component with high demand. Deep learning based speech synthesis system has shown impressive improvements in performance compared to the hidden Markov models (HMMs)-based speech synthesis [1]. Various deep learning techniques were proven to express nonlinear relation between text and speech. Since speech has complex time-dependencies, recurrent neural networks (RNNs) with long short term memory (LSTM) [2, 3, 4], simplified LSTM [5], or gated recurrent unit (GRU) [5, 6, 7] have been applied to improve the performance in acoustic modelling.

Although nonvariational RNNs introduced in [2, 3, 6] improved the performance over the non-recurrent deep neural networks, such methods have difficulty in capturing the variability in data due to the entirely deterministic structures. To model the variability in highly structured sequential data such as natural speech or handwriting, variational recurrent neural network (VRNN) is introduced in [8]. The VRNN generates an estimated input-like sequence conditioned on latent random prior and RNN state variable.

In this paper, we propose an acoustic modelling technique using the adversarially trained variational recurrent neural network (AdVRNN) as an alternative to the conventional RNNs for statistical parametric speech synthesis (SPSS). Unlike the VRNN, the proposed AdVRNN for SPSS takes the input of a linguistic feature sequence with the latent random prior and the RNN state variable. AdVRNN for SPSS generates not a sequence of linguistic features but a sequence of acoustic features. In this regard, AdVRNN is closer to the encoder-decoder model in [6] than an autoencoder. The AdVRNN is capable of modeling variability in a sequence efficiently than the vanilla encoder-decoder model due to latent random variable.

For training AdVRNN, an adversarial training scheme similar to the generative adversarial networks [9] is employed to capture the detailed structure of real acoustic features. A discriminator is introduced during the training phase to distinguish the generated acoustic feature sequence from the real data sequence. In order to avoid oversmoothing, which is one of the major problems in speech synthesis, a method to reducing the cost of discriminator was used instead of maximizing the variational lower bound. Unlike the sampled noise prior in [9], a latent random variable inferred from linguistic features and the RNN state variables are used to generate acoustic features in AdVRNN. It is shown that the proposed method performs better than the conventional RNN based speech synthesis such as GRU for both objective and subjective measure. The detailed structure and training scheme of the AdVRNN for SPSS is introduced in Section 3.

2. Background

In this section, we will give a brief review of the conventional VAE and VRNN.

2.1. Variational Autoencoder

Employing the structure of autoencoder, where the network aims at generating the input at the output layer, VAE introduces latent random variable to apply stochastic component in autoencoder and to model the variations in observations [10]. VAE is composed of an encoder network which maps the observed data $\mathbf{x}$ to latent random variable $\mathbf{z}$, and a decoder network which maps the latent random variable $\mathbf{z}$ to the output $\mathbf{x}$ same as the input. The prior distribution of latent random variable $\mathbf{z}$ is usually assumed to be standard Gaussian. In speech application, usually the observation is acoustic parameters. However, the difficulty in VAE comes from the intractability in inferencing the posterior distribution $p(\mathbf{z}|\mathbf{x})$. To overcome this challenge, VAE employs the variational approximated posterior $q(\mathbf{z}|\mathbf{x})$ with neural network. Using $q(\mathbf{z}|\mathbf{x})$, the variational lower bound is derived

10.21437/Interspeech.2018-1598

as follows:

$$\begin{aligned}\log p(\mathbf{x}) \geq = & -D_{KL}(q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})) \\ & + \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] \end{aligned} \quad (1)$$

where D_{KL} refers to Kullback-Leibler divergence (KL divergence) between $q(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{z}|\mathbf{x})$, measuring the similarity between the two distributions. Such KL divergence in the RHS of (1) implies the regularization of the encoder parameters, and the remaining term in the RHS refers to the reconstruction error between input and output of VAE. The reconstruction error forms as the VAE decoder network generates the input observation $\mathbf{x}$ from the $\mathbf{z}$ with distribution $\log p(\mathbf{x}|\mathbf{z})$. Both encoder and decoder are jointly trained by maximizing (1).

2.2. Variational Recurrent Neural Network

With highly structured sequential data, VAE can be extended into a recurrent framework. This idea, known as the VRNN, can model highly nonlinear dynamics of sequential data and capture the time dependency of sequences [8]. Similar to VAE, VRNN is composed of an encoder and a decoder.

2.2.1. Decoder

Unlike the VAE, the prior of the latent random variable $\mathbf{z}_t$ of the VRNN is not standard Gaussian distribution but conditioned on RNN state variable $\mathbf{h}_{t-1}$ as follows:

$$\begin{aligned} & p(\mathbf{z}_t) = \mathcal{N}(\mu_{p,t}, \sigma^2_{p,t}), \\ & [\mu_{p,t}, \sigma_{p_t}] = \varphi^{pri}(\mathbf{h}_{t-1}) \end{aligned} \quad (2)$$

where $\mu_{p,t}, \sigma_{p,t}$ denote the mean and standard deviation of prior distribution and φ^{pri} denotes neural network which models the prior distribution. The latent random variable can capture time dependency context using $\mathbf{h}_{t-1}$ in prior distribution. For the generation model distribution $p(\mathbf{x}_t|\mathbf{z}_t)$ is conditioned on $\mathbf{z}_t$ and $\mathbf{h}_{t-1}$ as follows:

$$\begin{aligned} & p(\mathbf{x}_t|\mathbf{z}_t) = \mathcal{N}(\mu_{x,t}, \sigma^2_{x,t}), \\ & [\mu_{x,t}, \sigma_{x,t}] = \varphi^{dec}(\phi^z(\mathbf{z}_t), \mathbf{h}_{t-1}) \end{aligned} \quad (3)$$

where $\mu_{x,t}, \sigma_{x,t}$ denote the mean and standard deviation of generation model distribution and φ^{dec} is a deep neural network which captures the generation model distribution and ϕ^z is an embedding network of $\mathbf{z}_t$. RNN state variable $\mathbf{h}_t$, uses previous state variable $\mathbf{h}_{t-1}$ and $\mathbf{x}_t$, $\mathbf{z}_t$ for updating the state variable as follows:

$$\mathbf{h}_t = \varphi^{rec}(\phi^x(\mathbf{x}_t), \phi^z(\mathbf{z}_t), \mathbf{h}_{t-1}) \quad (4)$$

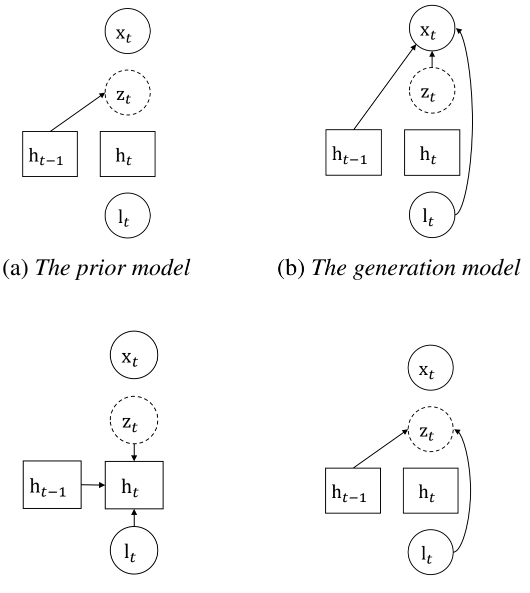
where φ^{rec} is a state transition function in the RNN and ϕ^x is embedding networks of $\mathbf{x}_t$.

2.2.2. Encoder

The inference model for encoder network at time $\mathbf{t}$ can be expressed using variational approximation posterior $q(\mathbf{z}_t|\mathbf{x}_t)$ which is a function of $\mathbf{x}_t$ and $\mathbf{h}_{t-1}$ as follows:

$$\begin{aligned} & q(\mathbf{z}_t|\mathbf{x}_t) = \mathcal{N}(\mu_{z,t}, \sigma^2_{z,t}), \\ & [\mu_{z,t}, \sigma_{z,t}] = \varphi^{enc}(\phi^x(\mathbf{x}_t), \mathbf{h}_{t-1}). \end{aligned} \quad (5)$$

where $\mu_{z,t}, \sigma_{z,t}$ denote the mean and standard deviation of $q(\mathbf{z}_t|\mathbf{x}_t)$ and φ^{enc} is the deep neural network which capture approximated posterior distribution.



(a) *The prior model* (b) *The generation model*

(c) *State transition model* (d) *Inference model*

Figure 1: *Graphical representation for AdVRNN opearation.*

2.2.3. Variational Lower Bound

For learning, the variational lower bound in (1) is modified as follows due to time dependency:

$$\begin{aligned} \mathbb{E}_{q(\mathbf{z}_{\leq t}|\mathbf{x}_{\leq t})}\Bigg[& \sum_{t=1}^{T}(-D_{KL}(q(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})\|p(\mathbf{z}_t|\mathbf{z}_{<t}, \mathbf{x}_{<t})) \\ & + \log p(\mathbf{x}_t|\mathbf{z}_{\leq t}, \mathbf{x}_{<t}))\Bigg]. \end{aligned} \quad (6)$$

The concepts of variational lower bound of VRNN is same as VAE which consist of KL divergence term which regularize the encoder parameters and reconstruction error term.

For more details about VAE and VRNN such as reparameterization tricks, the reader is refereed to [10, 8].

3. Speech Synthesis Using AdVRNN

Using the idea of VRNN, we propose an AdVRNN based speech synthesis technique which can model the complex nonlinear relation between linguistic feature sequence and acoustic feature sequence effectively. In this section, the structure of the acoustic model and the training procedure of the proposed technique are described.

3.1. AdVRNN based Acoustic Modeling

Using the aforementioned VRNN from section 2, observations can be generated using a latent random variable. Since a typical acoustic model in speech synthesis system takes a linguistic feature sequence as input and generates an acoustic feature sequence as output, the VRNN formulation is modified for speech synthesis application in a way shown in Figure 1.

3.1.1. Decoder

The prior distribution follows the same as (2). However, due to the mapping between linguistic feature sequence and acoustic feature sequence in speech synthesis, the generation model

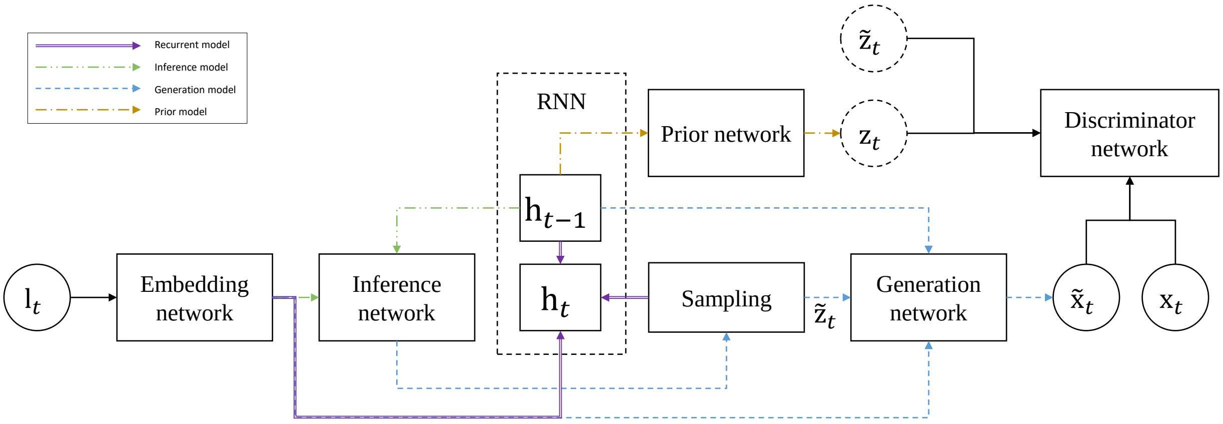


Figure 2: *Overall training procedure of AdVRNN.*

distribution is conditioned not only on $\mathbf{z}_t$ and $\mathbf{h}_{t-1}$ but also on linguistic feature $\mathbf{l}_t$ as follows:

$$\begin{aligned} &p(\mathbf{x}_t|\mathbf{z}_t,\mathbf{l}_t) = \mathcal{N}(\mu_{x,t},\sigma^2_{x,t}),\\ &[\mu_{x,t},\sigma_{x,t}] = \varphi^{dec}(\phi^z(\mathbf{z}_t),\phi^l(\mathbf{l}_t),\mathbf{h}_{t-1}) \end{aligned} \tag{7}$$

where ϕ^l is the embedding network of $\mathbf{l}_t$. Applying the similar modification, the state update equation in RNN can be expressed as follows:

$$\mathbf{h}_t = \varphi^{rec}(\phi^l(\mathbf{l}_t),\phi^z(\mathbf{z}_t),\mathbf{h}_{t-1}). \tag{8}$$

Note that the above state update is missing $\mathbf{x}_t$ since in synthesis stage, the error in synthesized speech propagates throughout the timesteps and can amplify the error to cause performance degradation.

3.1.2. Encoder

The true posterior is a function of $\mathbf{l}_t$ and $\mathbf{h}_{t-1}$ and can be approximated with $q(\mathbf{z}_t|\mathbf{l}_t)$ as follows:

$$\begin{aligned} &q(\mathbf{z}_t|\mathbf{l}_t) = \mathcal{N}(\mu_{z,t},\sigma^2_{z,t}),\\ &[\mu_{z,t},\sigma_{z,t}] = \varphi^{enc}(\phi^l(\mathbf{l}_t),\mathbf{h}_{t-1}). \end{aligned} \tag{9}$$

3.2. Training Procedure

The variational lower bound for AdVRNN can be derived using a similar approach to (6). However, the variational inference method is known to blur the generated samples [9, 11, 12]. In speech synthesis, this can cause oversmoothing of the generated speech which makes synthesized speech muffled. In this paper, we propose to use adversarial schemes to overcome the oversmoothing problems as in Figure 2. In figure 2, brown, blue, purple, green lines indicate the prior model, generation model, recurrent model, and inference model respectively as mentioned in section 3.1.

Following a typical generative adversarial network, the proposed method uses a discriminator to distinguish the followings:

1. For a real data sequence $\mathbf{x}_{\leq t}$, generate the corresponding latent variable sequence $\tilde{\mathbf{z}}_{\leq t}$ using inference model as defined in (9)
2. Generate $\tilde{\mathbf{x}}_{\leq t}$ and $\mathbf{z}_{\leq t}$ from the generation model and the prior model as defined in (2) and (7), respectively.

Then, a discriminator is built to discriminate between $\{\mathbf{x}_t, \tilde{\mathbf{z}}_t\}$ and $\{\tilde{\mathbf{x}}_t, \mathbf{z}_t\}$.

Discriminating $\{\mathbf{z}_t, \tilde{\mathbf{z}}_t\}$ is similar to KL divergence in (6) and discriminating $\{\mathbf{x}_t, \tilde{\mathbf{x}}_t\}$ is similar with reconstruction term in (6) where it has similar meaning with variational lower bound. The rest of training procedure follows the same as typical GAN training[9].

4. Experiments

In order to evaluate the performance of the proposed AdVRNN based speech synthesis system, several experiments on objective measurements and subjective listening tests were conducted.

For the experiments, a Korean speech database spoken by a male speaker was applied. Speaker provided 2,250 utterances of narrative speech data amounting to about 230 minutes. Among 2,250 utterances we use 2,000 utterances for training and 200 utterances for validation and 50 utterances for test. Each utterance was sampled at 16kHz and 20 ms Hamming window was applied with 5 ms frame shift for acoustic feature extraction. STRAIGHT vocoder were used to extract the acoustic feature [13]. For the spectrum feature, 25th-order mel-scaled cepstrum vector was used, and for the excitation feature, 1-dimensional logarithmic fundamental frequency (lf0) and 5-dimensional band aperiodicity (bap) were used. To make a continuous lf0 sequence, the lf0 values of the unvoiced region were filled during the normalization process. Also dynamic feature Δ and $\Delta\Delta$ were attached for each features. The extracted acoustic feature is normalized to follow white Gaussian in order to use the acoustic feature as the target $\mathbf{x}_t$ for the speech synthesis systems. For input linguistic feature, 547-dimensional binary feature for categorical linguistic contexts and 12-dimensional numeric feature for numerical linguistic contexts, position and duration were used together.

We used a deep hybrid GRU-based deterministic system for baseline speech synthesis to compare the performance. For simplicity we will call deep hybrid GRU as hybrid GRU or GRU. Such hybrid system was configured to have two GRU layers above two feedforward hidden layers with rectified linear unit (ReLU). Every layer is consisted of 256 nodes. For training the GRU-based speech synthesis system, the Adam optimizer in [14] was used.

For AdVRNN the configurations of model is as follows:

- ϕ^l, ϕ^z: two hidden layers with 256 ReLU nodes.
- ϕ^x: two hidden layers with 512 ReLU nodes.
- φ^{pri}: two feedforward hidden layers with 256 ReLU node. The output layer is composed of linear layer for $\mu_{p,t}$ and softplus layer for $\sigma_{p,t}$. The dimension of the latent random variable $\mathbf{z}_t$ is 256.
- φ^{dec}: two feedforward hidden layers with 1024 tanh nodes. The output layer is composed of linear layer for $\mu_{x,t}$ and softplus layer for $\sigma_{x,t}$.
- φ^{enc}: two feedforward hidden layers with 512 ReLU nodes. The output layer is composed of linear layer for $\mu_{z,t}$ and softplus layer for $\sigma_{z,t}$.
- For RNN, we use a GRU with 32 nodes.
- For discriminator, four feedforward hidden layers with the bottom three feedforward layers composed of ReLU layers with 256 nodes and the top feedforward hidden layer of a ReLU layer with 128 nodes were used. The output layer is 1 -dimensional sigmoid layer to output whether the inputs is real or not.

We used Adagrad optimizer in [15] to train AdVRNN and we used Tensorflow[16], a library for deep learning, for both GRU and AdVRNN implementations in our experiments.

Table 1: *Objective measurement of comparative models.*

	MCD	RMSE of f0	bap distance
GRU	6.203	**23.061**	2.420
AdVRNN	**5.808**	24.386	**2.312**

4.1. Objective performance evaluation

We compared the outputs of the two algorithms mentioned above: the conventional GRU approach and the proposed AdVRNN approach. For objective measure, we used averaged mel-cepstral distance (MCD) in dB scale, root mean square error (RMSE) of f0 in Hz, and bap distance. The results of the objective performance tests are shown in Table 1.

The results show that the AdVRNN approach is more effective for modeling highly structured speech data than GRU approach. However, the performance of GRU is better in RMSE of f0 measurement. We consider the reason for f0 degradation is due to the unvoiced part in normalization process. The interpolated part of unvoiced region to make continuous lf0 does not account for a real lf0 value, and this can influence the lf0 decoding process. Therefore, the performance of the proposed system related to f0 could be improved if an accurate continuous pitch contour is available.

Table 2: *Results of MOS test: GRU and AdVRNN based speech synthesis.*

	GRU	AdVRNN
MOS	2.836 ± 0.946	3.623 ± 0.955

4.2. Subjective performance evaluation

We also performed a subjective listening test to compare the AdVRNN with the GRU based speech synthesis. 11 participants listened to 20 sentences from each method, in which the sentences were randomly chosen from 50 test sentences. Each listener was provided with the speech samples in a random order and was asked to measure the speech quality in terms of the mean opinion score (MOS). Each subject provided scores in the range of [1,5] with a large value indicating high performance. The results are shown in Table 2 from which we can find that the proposed method outperformed the conventional GRU method. From this results it is shown that the AdVRNN has better intelligibility and quality than the discriminative GRU model.

5. Conclusions

In this paper, we proposed using a VRNN as an alternative method for acoustic modeling in speech synthesis system. Since speech contains high variability information, we apply the VRNN, which can express the variability within the highly structured data efficiently. Also, instead of using the conventional variational lower bound, we used adversarial training scheme to increase dynamic range for synthesized speech data. We call this VRNN with adversarial training scheme as AdVRNN. From the experimental results, it is shown that the proposed AdVRNN based method outperforms the conventional RNN-based method for acoustic modeling.

6. Acknowledgements

This research was supported by Search & Clova, NAVER Corp., Seongnam, Korea

7. References

[1] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. IEEE Int. Conf. on Acoust., Speech and Signal Process.*, 2013, pp. 7962–7966.

[2] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[3] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5, pp. 602 – 610, Jul. 2005.

[4] H. Zen and H. Sak, "Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis," in *Proc. IEEE Int. Conf. on Acoust., Speech and Signal Process.*, 2015, pp. 4470–4474.

[5] Z. Wu and S. King, "Investigating gated recurrent networks for speech synthesis," in *Proc. IEEE Int. Conf. on Acoust., Speech and Signal Process.*, 2016, pp. 5140–5144.

[6] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[7] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[8] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," in *Proc. Advances in Neural Inform. Process. Syst.*, 2015, pp. 2980–2988.

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Advances in Neural Inform. Process. Syst.*, 2014, pp. 2672–2680.

[10] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[11] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *Proc. Int. Conf. on Mach. Learn.*, 2016, pp. 1558–1566.

[12] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville, "Adversarially learned inference," in *Proc. Int. Conf. on Learn. Representations*, 2017.

[13] H. Kawahara, "Speech representation and transformation using adaptive interpolation of weighted spectrum: vocoder revisited," vol. 2, 1997, pp. 1303–1306.

[14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[15] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul., pp. 2121–2159, 2011.

[16] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, pp. 265–283.