RawNet: Fast End-to-End Neural Vocoder

Yunchao He, Haitong Zhang, Yujun Wang

Xiaomi Inc.

heyunchao@xiaomi.com, wangyujun@xiaomi.com

Abstract

Neural networks based vocoders have recently demonstrated the powerful ability to synthesize high quality speech. These models usually generate samples by conditioning on some spectrum features, such as Mel-spectrum. However, these features are extracted by using speech analysis module including some processing based on the human knowledge. In this work, we proposed RawNet, a truly end-to-end neural vocoder, which use a coder network to learn the higher representation of signal, and an autoregressive voder network to generate speech sample by sample. The coder and voder together act like an auto-encoder network, and could be jointly trained directly on raw waveform without any human-designed features. The experiments on the Copy-Synthesis tasks show that RawNet can achieve the comparative synthesized speech quality with LPCNet, with a smaller model architecture and faster speech generation at the inference step.

Index Terms: neural vocoder, speech synthesis, raw waveform modelling, end-to-end vocoder

1. Introduction

Traditional vocoding approaches [1] [2] [3] are commonly composed of an speech analysis module and a waveform generation module. The analysis module is responsible for extracting the acoustic features from the raw waveform while the waveform generator for re-constructing audio signal from the features. Some commonly used acoustic features are extracted, based on some simplified speech production models, such as the source-filter model [4] [5] [6]. For example, in [1] and [2], the acoustic features used include the log fundamental frequency (lf0), voice/unvoiced binary value (uv), the spectrum and band aperiodicities. However, the underlying assumption of these models, for one thing make it complicated to generate the waveform, for another often introduce some flaw into the generated speech.

More recently, neural vocoders use neural networks to directly learn the transformation from the acoustic features to audio waveform such as WaveNet [7], LPCNet [8], WaveGlow [9] and FFTNet [10]. They could partly overcome the above mentioned disadvantages of the traditional methods, by getting rid of the complicated human-designed speech analysis and generation step. These neural network-based methods that directly synthesize raw speech waveform from acoustic features, could achieve the state-of-art performance in text-to-speech synthesis. However, the waveform generation is quite slow due to the complicated model structure and the autoregressive property. In addition, the performance of these neural vocoder is also partly limited by the conditioning features, which are extracted based on the simplified speech production models.

The acoustic feature is a low-denominational representation of raw waveform. In a text-to-speech task, it's often predicted by the acoustic models, and is then used to reconstruct the predicted waveform. As the acoustic model is trained to minimize the gap between the ground-true acoustic features and predicted features, it's better to use these feature which is easy to be predicted by acoustic model, and easy to be extracted from raw waveform, and easy to reconstruct waveform with high quality. These three condition could be used to check if it's a good acoustic feature for speech synthesis.

To extract representative features from raw waveform, raw waveform based methods have been explored in many speech-related tasks. In speech recognition, [11] proposed the recognition model which is based directly on the raw waveform, and achieve better result than the model trained with hand-crafted acoustic features. In [12], the raw waveform is directly fed into the neural model for both speech and speaker recognition tasks. [12] shows the benefits gained in terms of model convergence, performance, and computational efficiency. [13] explores the representative feature directly from a large number of sound data, and yields the state-of-art result in acoustic object classification task. In [14], a fully convolutional network is used to enhance speech directly using raw waveform as model input and target.

Inspired by the success of the above methods, it is possible to further improve the existing neural vocoder by embedding the feature extractor model as part of the vocoder network and jointly optimizing it within the whole vocoder framework. In this paper, we propose an truly end-to-end neural vocoder architecture called *RawNet* to accomplish this goal. Here the term *end-to-end* means that RawNet takes the raw signal as input and can generate raw waveform as output, as autoencoder model works. RawNet is composed of a coder network which extracts acoustic features from raw waveform, and a voder network which reconstructs high quality speech waveform from features. These two components correspond to the analysis and synthesis module of a traditional vocoder. To our knowledge, this is the first time to use a single unified network to train the feature extraction network and speech synthesis network directly on raw waveform.

The rest of the paper is organized as follows. Section 2 introduces some related works, including speech feature extraction, using auto-encoder model for processing speech signal, and some popular neural vocoders in the field of speech synthesis. Section 3 presents the proposed model RawNet, and some crucial training strategies. Section 4 shows the experimental settings and results. Conclusion and future works are provided in the section 5.

2. Related Works

The architecture of RawNet is like to an autoencoder model, in which the encoder and decoder networks could correspond the coder and voder network of RawNet. There are some researches in employing an auto-encoder for extracting relative parameters for speech synthesis task. For example, [15] [16] use an autoencoder to extract excitation parameters, which is required by a traditional vocoder. In [17], an autoencoder based, non-linear

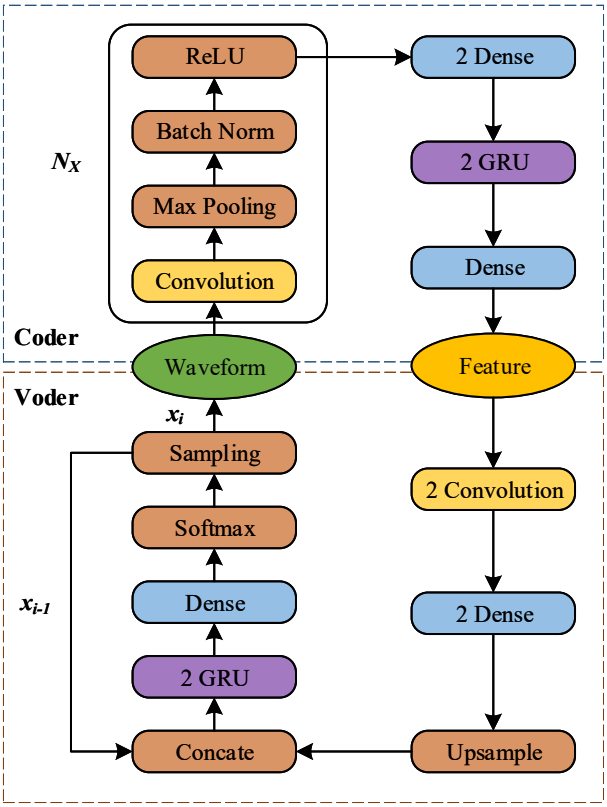

Figure 1: *The model architecture of RawNet, which mainly consists of two parts: coder and voder. The upper part is the coder network which extracts acoustic features from raw waveform, and the bottom part is the voder network which generates speech.*

and data-driven method is used to extract low-dimensional features from the FFT spectral envelop, instead of using the speech analysis module based on human knowledge. [17] also concludes that the proposed model outperforms the one which is based on the conventional feature extraction. The difference of RawNet to the above mentioned methods is that, RawNet directly takes waveform samples as input instead of just treats autoencoder as a feature dimension reduction method.

Our work resembles these works in that we use the similar auto-encoder based model framework for extracting higher representative features for speech synthesis. However, the novelties and contributions of our work is that (1) we directly extract the desired features from the raw waveform instead of modelling on the FFT spectral envelop, (2) we embed the feature extraction network into the unified end-to-end model to generate speech.

3. RawNet

This Section introduce the RawNet model. Figure 1 shows its overview architecture, which includes a coder network that extracts acoustic features from raw waveform, and a voder network that generates waveform conditioned on the acoustic features. These two parts are jointly trained in a single network, but could be used separately, corresponding to the analysis and synthesis procedure in a tradition vocoder system. More details are provided in this section.

3.1. Coder network

The coder network is used for feature extraction. The main components of the network are stacked convolutional layers, dense layers and GRU layers, as shown in the upper part of figure 1. By stacking multiple convolutional layers, the network can learn the high-level representation through a serious of lower-level filters. The convolutional architecture we used is similar to the convolutional network proposed in [13], which is used to learn the sound representation. By extending the network with GRU and dense layer, we empower the model with the ability to capture long-term relationship.

Since sound can vary in temporal length, the coder network has to handle inputs with variable lengths. It can be done by control the stride step in the convolutional layers, and the pooling size of pooling layers. As convolutional layers are invariant to location, we can convolve each layer to control the output length. Consequently, the frame size of the learned acoustic features are only determined by the convolutional and pooling layers.

3.2. Voder network

The voder network is for speech generation given acoustic features. Its structure is similar to that of LPCNet [8], but we make some modifications. When generating the next sample, LPCNet takes as the inputs the current predicted sample, current predicted excitation, global features from frame-rate network and linear prediction of current sample. Different from LPCNet's complicated input information, the voder network of RawNet only takes the current predicted sample and the conditioning acoustic features as input, which are concatenated together to be fed to the following layers.

The extracted acoustic features are first fed into two convolutional layers, followed by two dense layers. The output of the dense layer has the same length with the frame-length, and then is up-sampled to audio-sample length. In out experiments, we use a simple up-sampling method, i.e. repeating it K times, where K is the frame size. The up-sampled features, concatenated with previous predicted sample, are feed into a 2 GRU layers and a DualFC layer and a softmax function. Finally, sample can be generated by sampling.

Rather than scaling the sample values into a fixed range of values before feeding them into the network, we use u-law to apply the companding transformation to the sample values. An embedding representation is learned for each u-law level, essentially learning a set of non-linear functions to be applied to the u-law values.

3.3. Sampling method

It's reported in the paper of LPCNet [8] and FFTNet [10] that directly sampling from the output distribution can sometimes result in excessive noise. To address this problem, FFTNet proposed a conditional sampling method, which is multiplying the output logits by a constant value, i.e c=2, for voiced sound, and keep them remained for the unvoiced region. LPCNet replaces the binary voicing decision with a pitch correlation, which could be used to scale the output logits continuously.

In our experiments, we compared the multinomial sampling, conditional sampling, LPCNet's pitch correlation based sampling, and the simple argmax method. However, we found that the simple argmax method could generate very clear samples with the least noise, which is in accord with the results of the original WaveNet [7] and [18].

As the conditional sampling and LPCNet's method require pitch and pitch correlation to scale the output logits, and our Coder network does not learn these information explicitly. To use the conditional sampling or LPCNet's pitch correlation based sampling methods, we should extract pitch and pitch correlation as additional acoustic features. For example, we used the REAPER [19] tool to extract these features in the comparison experiment.

3.4. Noise injection

Due to the training error, the synthesized samples always contain some amount of noises. When generating samples, the network will generate samples that get noisier over time because of the auto-regressive property. If the voder network takes these noisy samples as input to generate the next sample, more and more randomness would be introduced to the network during training. As a result, the output samples will contain some clicking artifacts. To address this problem, we inject random noise to the input during training.

When training, we inject some Gaussian noises from $\mathcal{N}(0, 0.2)$ to the raw signal before feeding them into the voder network. And Gaussian noises from $\mathcal{N}(0, 0.1)$ are injected to the input of coder network. This injection strategy is adopted to ensure that the model could see different training data at each training iteration and could avoid over-fitting effectively, as the experiments indicate.

3.5. Post-synthesis denoising

Injecting noise enable the networks to see more training data and to avoid over-fitting problem. An additional benefit of injecting noise is to reduce the clicking artifact in the voiced part of sound. However, it also introduces a small amount of buzz noise to the silence part of unvoiced sound. The noise is sometimes audible with a low magnitude and only occurs in the silent part. Therefore, we apply a simple energy-based method [20] which is a baseline method in voice activity detection to reduce these noises. Experiments show that this method could almost eliminate these noises.

4. Experiments

To evaluate the power of RawNet, We conduct an AB listening test to compare the quality of the generated speech from RawNet and LPCNet. We will open-source the code with two pre-trained models and some generated samples, which is available at https://github.com/candlewill/RawNet.

4.1. Experimental setup

As the proposed system can be either speaker-independent or speaker-dependent, we evaluate the model in both setting using three different datasets. The CMU Arctic dataset [21] is used to train a speaker-independent vocoder. The CMU ARCTIC consists of around 1150 utterances for each speaker with both female and male. To reduce the accent variance, four speakers were selected consisting of two male speakers, *bdl* and *rms*, and two female speakers, *slt* and *clb*. For speaker-dependent experiments, we use a private Chinese dataset called mufei and LJ-Speech 1.1 [22] dataset. The former contains 20-hour audio from a single female speaker, while the latter consists of ∼24-hour of speech from a single female speaker. We excluded 100 samples from each dataset for evaluation test.

When training, the input to coder net is a short audio clip,

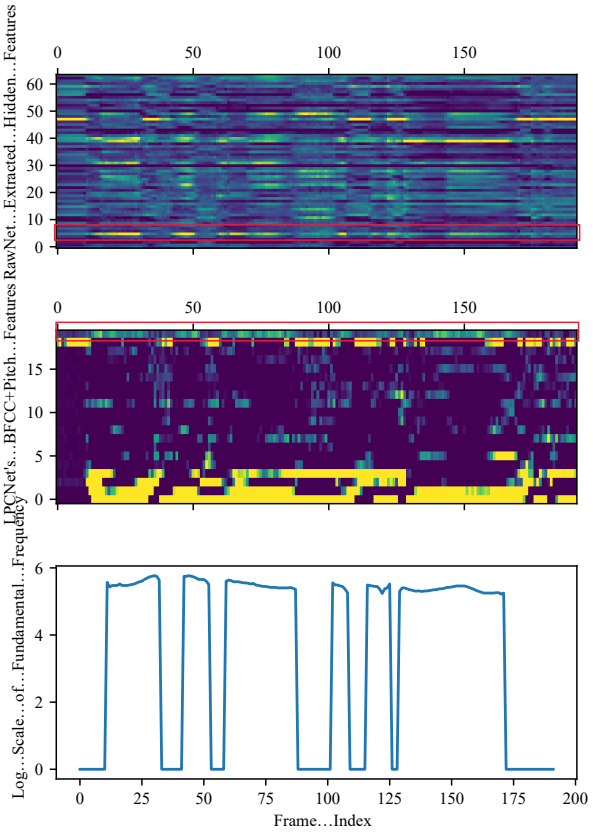

Figure 2: *Acoustic features comparison: the top layer shows the embedded features learned from the Coder; the middle layer is the illustration of the features used in LPCNet; the bottom layer is the F0 contour of the speech.*

which contains 3200 samples (i.e. 200ms for 16k speech). The clip is randomly selected from the original wave. The output of the coder are 20 frames of features, with 64 dimensions per frame. The training epoch in our experiments is 1500, with a batch size of 128*4. Training was performed on four Nvidia P40 GPUs with 22GB memory size. The implementation frameworks is Keras/TensorFlow. The cross-entropy loss is used as the loss function in the experiments. The weight matrices of the network are initialized with the normalized initialization, and the bias vectors are initialized to be 0. AMSGra [23]] optimization method (Adam variant) is used to update the training parameters, with an initialized learning rate of 1e-2.

4.2. Subjective evaluation

AB preference tests were conducted to assess the generated speech quality. In AB preference tests, for each task, we randomly selected 15 paired samples A and B from RawNet and LPCNet. There are 20 raters participating in the test, with 10 female and 10 male raters. The raters were asked to choose the sample with better quality. As Figure 3 shows, the generated speech by RawNet gains more preferences than those of LPCNet. More specifically, the difference in the speaker-independent task is larger than those in the speaker-dependent task.

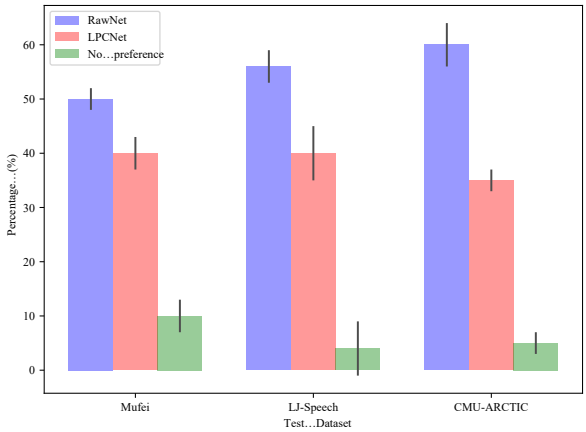

Figure 3: *A/B Preference Test Result of RawNet and LPCNet in three different datasets. The X-axis represents the datasets, while the y-axis indicates the percentage of preference.*

4.3. Visualization

Figure 2 illustrates the features extracted by RawNet coder, Bark-frequency cepstral coefficients (BFCC) [24] and pitch parameters (period, correlation) used in LPCNet, and the pitch contour of one single utterance. The extracted features are similar in a short time, which is in accordance with the fact that speech is stable or periodic in a short time. This shows implicitly that the learned features should be reasonably good.

The bottom layer presents the F0 contour of the utterance, with the zero value indicating the current phone is unvoiced. The middle layer also show the F0 information in the first dimension, which is highlighted by a red box. From the top layer, we can observe that the F0 can be extracted by the coder. As highlighted by the red box, the dark area indicates small F0 value or unvoiced and vice versa. This overall F0 contour is similar to that shown in the bottom layer. It indicates that the Coder of RawNet can learn the low-level information from signal, without human prior knowledge.

The effect of using post-synthesis denoising can be illustrated in Figure 4. In the top spectrogram, we can see the "clicks" in the highlighted box. After using the post-synthesis denoising, we can remove the "click" almost completely.

5. Conclusion

This paper proposes a new vocoder, which uses the raw waveform as input and output the raw waveform. The coder and voder can be trained jointly. Training such a network is a challenge, but we adopted several tricks to get the network perform well. As a result, the subjective evaluation shows that our proposed model can produce more natural/preferred speech than the recently proposed LPCNet. Visualization of the learned features helps illustrating that RawNet can extract reasonably good features from the raw waveform.

This work only proposes a plausible vocoder based on raw waveform. Some interesting future work based on this model can be conducted. For example, the embedded features learned from our coder can be used in other speech synthesis framework or any other speech-related tasks.

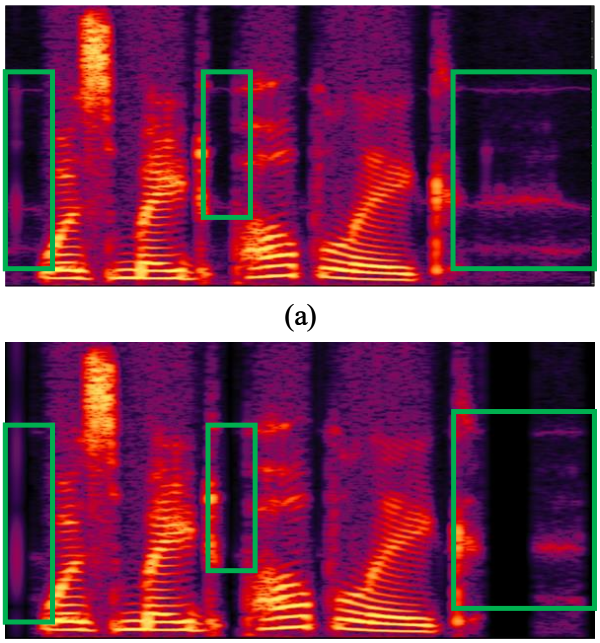

Figure 4: *The spectrogram comparison: the top layer indicates the spectrogram of utterance generated by RawNet without post-synthesis denoising, while the bottom layer gives the one with post-synthesis denoising. The green box region points out the difference.*

6. ACKNOWLEDGMENTS

The AB preference test is conducted with the help of Xiaomi AI Lab PM team. The computation resource is provided and maintained by Xiaomi SRE Team. The private mufei dataset is provided by Xiaomi AI Lab Speech Team. We thank them all.

7. References

[1] M. Morise, F. Yokomori, and K. Ozawa, "World: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[2] H. KAWAHARA, "Straight: Exploitation of the other aspect of vocoder," *The Journal of the Acoustical Society of Japan*, vol. 63, no. 8, pp. 442–449, 2007.

[3] Y. Agiomyrgiannakis, "Vocaine the vocoder and applications in speech synthesis," in *ICASSP*, 2015.

[4] P. Hedelin, "A tone oriented voice excited vocoder," in *ICASSP'81. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 6. IEEE, 1981, pp. 205–208.

[5] R. McAulay and T. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 4, pp. 744–754, 1986.

[6] Y. Stylianou, "Harmonic plus noise models for speech, combined with statistical methods, for speech and speaker modification," *Ph. D thesis, Ecole Nationale Superieure des Telecommunications*, 1996.

[7] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[8] J. Valin and J. Skoglund, "Lpcnet: Improving neural speech synthesis through linear prediction," *CoRR*, vol. abs/1810.11846, 2018. [Online]. Available: http://arxiv.org/abs/1810.11846

[9] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," *arXiv preprint arXiv:1811.00002*, 2018.

[10] Z. Jin, A. Finkelstein, G. J. Mysore, and J. Lu, "Fftnet: A real-time speaker-dependent neural vocoder," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*. IEEE, 2018, pp. 2251–2255. [Online]. Available: https://doi.org/10.1109/ICASSP.2018.8462431

[11] N. Zeghidour, N. Usunier, G. Synnaeve, R. Collobert, and E. Dupoux, "End-to-end speech recognition from the raw waveform," *arXiv preprint arXiv:1806.07098*, 2018.

[12] M. Ravanelli and Y. Bengio, "Speech and speaker recognition from raw waveform with sincnet," *arXiv preprint arXiv:1812.05920*, 2018.

[13] Y. Aytar, C. Vondrick, and A. Torralba, "Soundnet: Learning sound representations from unlabeled video," in *Advances in Neural Information Processing Systems*, 2016.

[14] S.-W. Fu, Y. Tsao, X. Lu, and H. Kawai, "Raw waveform-based speech enhancement by fully convolutional networks," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2017, pp. 006–012.

[15] S. Vishnubhotla, R. Fernandez, and B. Ramabhadran, "An autoencoder neural-network based low-dimensionality approach to excitation modeling for hmm-based text-to-speech," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2010, pp. 4614–4617.

[16] T. Raitio, A. Suni, L. Juvela, M. Vainio, and P. Alku, "Deep neural network based trainable voice source model for synthesis of speech with varying vocal effort," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[17] S. Takaki and J. Yamagishi, "A deep auto-encoder based low-dimensional feature extraction from fft spectral envelopes for statistical parametric speech synthesis," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 5535–5539.

[18] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," *arXiv preprint arXiv:1601.06759*, 2016.

[19] D. Talkin, "Reaper: Robust epoch and pitch estimator," *Github: https://github. com/google/REAPER*, 2015.

[20] K. Sakhnov, E. Verteletskaya, and B. Simak, "Approach for energy-based voice detector with adaptive scaling factor." *IAENG International Journal of Computer Science*, vol. 36, no. 4, 2009.

[21] J. Kominek and A. W. Black, "The cmu arctic speech databases," in *Fifth ISCA workshop on speech synthesis*, 2004.

[22] K. Ito, "The lj speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[23] S. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," in *International Conference on Learning Representations*, 2018.

[24] B. C. Moore, *An introduction to the psychology of hearing*. Brill, 2012.