Investigation of training data size for real-time neural vocoders on CPUs

Keisuke Matsubara1,2,*, Takuma Okamoto2, Ryoichi Takashima1, Tetsuya Takiguchi1, Tomoki Toda3,2, Yoshinori Shiga2 and Hisashi Kawai2

1Graduate School of System Informatics, Kobe University, 1–1 Rokkodaicho, Nada-ku, Kobe, 657–8501 Japan
2National Institute of Information and Communications Technology, 3–5 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619–0289 Japan
3Information Technology Center, Nagoya University, Furo-cho, Chikusa-ku, Nagoya, 464–8601 Japan

Keywords: Speech synthesis, Neural vocoder, LPCNet, Parallel WaveGAN

1. Introduction

In recent years, neural speech synthesis technology in text-to-speech has been developed rapidly. WaveNet [1], which is an autoregressive generative raw audio model, was a great turning point in recent developments. Neural vocoders such as the WaveNet vocoder [2] have achieved much better quality than traditional source-filter vocoders [3]. In particular, Tacotron 2 with a WaveNet vocoder can synthesize high quality speech that is indistinguishable from natural speech [4].

Because of the autoregressive architecture, the WaveNet vocoder has a problem with the slow inference speed and thus is limited in its application. To solve this problem, various neural vocoder models have been proposed, and some can synthesize speech waveforms in real-time, even in a restricted environment with mobile CPUs [5,6]. These models are mostly trained on large sets of training data of more than 10 hours. In a real scenario, however, there are many cases in which it is difficult to collect such a large set of training data. Therefore, it is important to investigate how many training data are required for real-time neural vocoders.

In this paper, we investigate the relation between the number of training data and synthesized speech quality in LPCNet [5] and Parallel WaveGAN [6], which can implement real-time synthesis on CPUs. Specifically, we prepare several sets of training data from 1/8 to 9 hours to train these models. Then we evaluate the synthesized speech quality using the mean opinion score (MOS) [7]. To expand the synthesis frequency range of LPCNet, which was originally developed with a sampling frequency of 16 kHz [5], we provide LPCNet with 20-dimensional bark cepstra to obtain a sampling frequency of 24 kHz. Additionally, we investigate the inference speeds by increasing the number of CPU cores in real-time, even in a restricted environment with a mobile CPU.

2. Real-time neural vocoders on CPUs

We briefly introduce LPCNet and Parallel WaveGAN which can synthesize speech waveforms in real-time on CPUs. Additionally, we provide LPCNet with 20-dimensional bark cepstra to obtain a sampling frequency of 24 kHz.

2.1. LPCNet

LPCNet is a neural vocoder model based on WaveRNN [8], which has a recurrent neural network architecture. LPCNet infers residual signals between natural speech and predicted speech computed using linear prediction coding (LPC). WaveRNN infers 16 bit audio samples using dual-softmax, whereas LPCNet infers residual samples compressed using 8 bit μ-law coding, which can suppress quantization errors. Therefore, LPCNet can synthesize high quality speech waveforms while reducing the network model size [5].

LPCNet consists of two neural network blocks called the frame rate network and sample rate network. The input features consist of 18-dimensional Bark-Frequency Cepstrum Coefficients (BFCCs) and two pitch parameters (period and correlation) for a sampling frequency of 16 kHz.

Predicted samples using LPC are computed as follows:

\[ p_t = \sum_{k=1}^{M} a_k s_{t-k} \]  

where \( p_t \) and \( s_t \) denote the predicted samples and natural samples at time \( t \), respectively. \( a_k \) is the \( k \)-th linear prediction coefficient.

The linear prediction coefficients \( a_k \) are calculated from the input BFCCs. Specifically, the BFCCs are first converted to a linear-frequency power spectral density (PSD). Then the PSD is converted to an autocorrelation by applying the inverse FFT. Finally, the prediction coefficients are computed from the autocorrelation using the Levinson-Durbin algorithm.

The frame rate network extracts intermediate features from the input acoustic features. The sample rate network receives a previous 1-step natural sample, a predicted sample using LPC and the output from the frame rate network to infer a current residual sample.

The sample rate network is also an autoregressive model that consists of the Gated Recurrent Unit (GRU), and therefore it requires a long time for inference by nature as WaveNet. However, the sparse coding that forces the lowest value of a weight matrix to zero is applied to accelerate inference. Then LPCNet can synthesize speech waveforms in real-time, even in a restricted environment with a mobile CPU.

*Work performed while interning at NICT.
*e-mail: kmatsubara@stu.kobe-u.ac.jp

[doi:10.1250/ast.42.65]
2.1.1. LPCNet for a sampling frequency of 24 kHz

Original LPCNet was proposed as a neural vocoder that can synthesize speech waveforms sampled with a sampling frequency of 16 kHz. To synthesize a higher fidelity speech waveform from that of 24 kHz, we expand 18-dimensional BFCCs to 20-dimensional ones. Specifically, we expand the dimension of the filter bank as shown in Fig. 1. This expansion follows a voice compression method called Opus codec [9]. In this method, the bands are divided at even intervals at low frequencies, and divided following the Bark scale at high frequencies. We apply the 20-dimensional Opus filter bank to cover the range from 0 to 12 kHz that can be sampled using 24 kHz audio.

2.2. Parallel WaveGAN


The input features are white noise and acoustic features, and the generator WaveNet synthesizes all speech samples simultaneously. The discriminator is trained to classify generated samples as “fake” and ground truth samples as “real.” The loss function consists of adversarial loss and STFT loss. Adversarial loss is a typical loss function on GAN architecture. Additionally, STFT loss in the frequency domain is applied to improve the stability and efficiency of generator training in the same manner as Parallel WaveNet [11]. Although the STFT resolution is a trade-off between time and frequency domains, this problem can be solved by applying multi-resolution STFT loss to maintain high accuracy in both domains.

Parallel WaveGAN has no autoregressive structure, and therefore all speech samples can be generated simultaneously to synthesize speech waveforms rapidly. Then it is able to synthesize speech waveforms in real-time with only CPUs°. Additionally, it can be trained as a parallel generative model directly without knowledge distillation unlike Parallel WaveNet [11].

3. Experiments

3.1. Experimental setup

We conducted experiments for analysis-synthesis on LPCNet and Parallel WaveGAN neural vocoders to evaluate the relation between the number of training data and synthesis quality. We used 7,697 sentences (about 10 hours) uttered by a Japanese female speaker in the JSUT corpus [12] where the sampling frequency was 48 kHz. In the experiments, the speech samples were downsampled to 24 kHz. Additionally, we removed silent parts in the speech samples by applying forced alignment using the Julius speech recognition toolkit [13]. At most, 7,497 utterances (9 hours) were used for training. 100 utterances were used for validation, and the remaining 100 utterances were used for evaluation. Seven types of training data were prepared: 9, 5, 3, 1, 1/2, 1/4 and 1/8 hours. As baselines, we also evaluated the WORLD [3], WaveNet [2] and WaveGlow [14] vocoders.

To ensure the reliability of the experiments, we also conducted experiments using CMU ARCTIC, which is an English multi-speaker corpus [15]. We used the speech samples of one male and one female speaker (bdl and slt), which comprised 1,131 utterances for each speaker. The speech samples were recorded at 32 kHz, and we downsampled them to 24 kHz. 1,111 utterances (about 1 hour) were used for training and 15 utterances were used for evaluation. We prepared three types of training data: 1, 1/2 and 1/4 hours. As the baseline, we also evaluated the WORLD and WaveNet vocoders in the experiments using the CMU ARCTIC corpus.

In the WORLD vocoder, 35-dimensional mel-cepstra with warping coefficient $\alpha = 0.455$, and 3-dimensional parameters for the smooth vocal tract spectrum and aperiodicity components were obtained from the original WORLD spectrum and aperiodicity coefficients (1,025 dimensions) and the vocoded waveforms were synthesized using the compressed acoustic features.

The network structure of LPCNet was the same as that in [5]. The input features consisted of 20-dimensional BFCCs, pitch period and pitch correlation. To calculate the BFCCs, spectrum analysis was performed with a window length of 20 ms and a frame shift of 10 ms, and the Bark-scale filter bank was applied. Pitch calculation was based on an open-loop cross-correlation search.

The network structure of Parallel WaveGAN was the same as that in [6]. The input features were 80-band log-mel spectrograms with a band-limited frequency range (80 to 7,600 Hz). The window and shift lengths were set to 20 ms and 12.5 ms, respectively.

The network structure WaveNet vocoder was the same as that in [2]. We applied time-invariant noise shaping [16] to suppress perceptual noise components caused by the prediction error. The input features were 80-band log-mel spectrograms. The window and shift lengths were set to 42.7 ms and 5 ms, respectively.

The network structure of WaveGlow was the same as that in [17]. The input features were also 80-band log-mel spectrograms as used in Parallel WaveGAN.

As the training optimizer, Adam [18] was used for WaveNet and LPCNet, and RAdam [19] was used for Parallel WaveGAN and WaveGlow. All models were trained on NVIDIA V100 GPUs. The number of GPUs was 1, 2, 4 and 4 in LPCNet, Parallel WaveGAN, WaveNet and WaveGlow, respectively.

In LPCNet, the network model was trained using TensorFlow and a C-based implementation was used for inference as in the original code.° By contrast, other neural vocoders were implemented using PyTorch both for training and inference.

°https://github.com/kan-bayashi/ParallelWaveGAN

°https://github.com/mozilla/LPCNet
We performed MOS tests to evaluate the subjective perceptual quality. In the experiment using the JSUT corpus, 18 native Japanese speakers without hearing loss listened to the synthesized speech samples using headphones and were asked to assess the quality of the speech samples using a five-level rating. The number of experimental conditions was 19, including the ground truth, and there were 20 utterances for each condition. In the experiment using the CMU ARCTIC corpus, 15 native English speakers listened to the synthesized speech samples using headphones and were asked to assess the quality in the same manner as the experiment using JSUT. The number of experimental conditions was 18, and there were 15 utterances for each condition.

3.2. Evaluation

3.2.1. Real-time factor evaluation

Table 1 shows the results of real-time factors (RTFs) in the neural vocoders for inference using an NVIDIA Tesla V100 GPU or Intel Xeon 6152 CPUs, and Fig. 2 shows the transition of the RTFs with respect to the number of CPU cores in LPCNet and Parallel WaveGAN. As shown in Table 1, we found that LPCNet synthesized speech waveforms as fast as 0.24 RTF with a single CPU core. The RTF of Parallel WaveGAN with a single CPU core was 2.38, and did not realize real-time synthesis. However, as shown in Fig. 2, the RTF to 0.41 can be improved by increasing the number of CPU cores to at most 16 cores. By contrast, the RTF of LPCNet was not affected by the number of CPU cores. This is because LPCNet is an autoregressive model and parallel processing cannot be performed even if the number of CPU cores is increased. To improve the RTF of Parallel WaveGAN for inference, a C-based implementation instead of PyTorch is required as in LPCNet.

3.2.2. Subjective evaluation

Figure 3 shows the result of the MOS tests using the JSUT corpus. When we used 9 hours of training data, the WaveNet vocoder achieved the highest score. As the number of training data decreased, the synthesis quality of both LPCNet and Parallel WaveGAN deteriorated. However, according to the t-tests, there was no significant difference in the results from 1 hour to 9 hours. In the case of 1/8 hour, although the synthesis quality of Parallel WaveGAN was worse than that of WORLD, that of LPCNet remained higher than that of WORLD. In conclusion, we found that LPCNet and Parallel WaveGAN could be trained sufficiently with about 1 hour of training data. We assume that the reason that LPCNet was better quality than Parallel WaveGAN or WaveGlow was that past waveforms can be used for the autoregressive model. Additionally, the prediction residual is whitened and can achieve the same effect as the time-invariant noise shaping [16], which suppresses perceptual noise components. In this experiment, Parallel WaveGAN had a slightly lower quality than LPCNet and WaveGlow with a significant difference in the t-test. Although the weight for the adversarial loss was the same as that in [6], the synthesis quality of Parallel WaveGAN might be improved by adjusting the hyperparameters to match the corpus.

Figure 4 shows the results of the MOS tests using the CMU ARCTIC corpus. Similar to the experiment using the JSUT, the result of LPCNet had a high score 1 hour of data, and the synthesis quality significantly deteriorated in proportion to the decrease in the number of training data according to the t-tests. The results of Parallel WaveGAN using the female speaker showed a tendency similar to that of JSUT, but there was no significant difference between the results for 1/2 hour and 1 hour. In the experiments using the male speaker, the result for 1 hour was worse than that for 1/2 hour because

| Table 1 Real-time factors for inference using an NVIDIA Tesla V100 GPU or Intel Xeon 6152 CPUs. (*) denotes the number of CPU cores. |
|-------------------------------|------------|------------|
| Model                        | RTF-CPU    | RTF-GPU    |
| WaveNet                      | 2.064 (16) | 196        |
| LPCNet                       | 0.24 (1)   | 0.22       |
| Parallel WaveGAN             | 0.41 (16)  | 0.02       |
| WaveGlow                     | 2.1 (44)   | 0.07       |

Fig. 2 Real-time factors for inference that result from increasing the number of CPU cores.

Fig. 3 Result of the MOS test with 18 listening subjects using the JSUT corpus. Confidence level of the error bars was 95%.
some samples included a beeping noise, although it outperformed that of WORLD. Additionally, the synthesized speech of the male speaker was slightly noisy compared with that of the female speaker. As the results using the JSUT corpus demonstrate, hyperparameters such as the weight of adversarial loss also need to be properly adjusted to train Parallel WaveGAN to match the corpus especially in male speech.

Consequently, these experimental results indicate that LPCNet and Parallel WaveGAN can be trained sufficiently with about 1 hour of training data.

4. Conclusions
We trained LPCNet and Parallel WaveGAN with few training data and evaluated their performance. We found that LPCNet and Parallel WaveGAN can be trained sufficiently with about 1 hour of training data. Additionally, from the evaluation of inference speeds, we found that LPCNet and Parallel WaveGAN can synthesize speech waveforms in real-time with single or multiple CPU cores.

References

Fig. 4 Result of the MOS test with 15 listening subjects using the CMU ARCTIC corpus. Confidence level of the error bars was 95%.