GAUSSIAN LPCNET FOR MULTISAMPLE SPEECH SYNTHESIS

Vadim Popov, Mikhail Kudinov, Tasnima Sadekova

Huawei Technologies Co. Ltd.
Moscow, Russia

ABSTRACT

LPCNet vocoder has recently been presented to TTS community and is now gaining increasing popularity due to its effectiveness and high quality of the speech synthesized with it. In this work, we present a modification of LPCNet that is 1.5x faster, has twice less non-zero parameters and synthesizes speech of the same quality. Such enhancement is possible mostly due to two features that we introduce into the original architecture: the proposed vocoder is designed to generate 16-bit signal instead of 8-bit μ-companded signal, and it predicts two consecutive excitation values at a time independently of each other. To show that these modifications do not lead to quality degradation we train models for five different languages and perform extensive human evaluation.

***Index Terms*—** LPCNet, neural vocoder, multisample speech synthesis

1. INTRODUCTION

Deep learning has been evolving greatly during this decade and now its techniques are applied to numerous tasks including that of text-to-speech synthesis (TTS). Traditional hand-engineered compound pipelines are being replaced by neural approaches. The most popular one consists of two parts: a backend model that converts text into a sequence of acoustic features and a vocoder that generates speech conditioned on these features. Tacotron [1], Transformer TTS [2] and FastSpeech [3] are among possible solutions that can be used for the first part of this neural TTS pipeline.

As far as vocoders are concerned, the first truly successful neural-based model was WaveNet [4]. Dilated causal convolutions that are the main building blocks of WaveNet significantly increase receptive field which allows this model to keep track of long-term correlation between speech samples. WaveNet reached state-of-the-art synthesis quality and outperformed concatenative methods. However, there are several problems with this architecture. For example, synthesized speech often suffers from noise that causes large spectral distortion in a high-frequency band [5]. Also, what is more important, WaveNet's autoregressive nature makes this model inapplicable to real-time synthesis.

There are several architectures (ClariNet [6], Parallel WaveNet [7], WaveGlow [8]) that use the concept of flow (e.g. Inverse Autoregressive Flow [9]) to enable parallel synthesis: a series of invertible trainable transformations is applied to some simple distribution (e.g. standard normal in ClariNet) so that resulting distribution resembles that of real speech signal samples. Other architectures try to overcome computational complexity of autoregressive vocoders by applying a number of sophisticated compression and sampling techniques (e.g. WaveRNN [10]) or by designing new structure of convolution blocks (e.g. FFTNet [11]).

All of these models can achieve very good results without any use of classical techniques of speech synthesis. However, in many recent studies very good results have been achieved by combining deep learning approach with classical Source-Filter model of speech production (e.g. [5, 12, 13, 14]). The Source-Filter model is based on a strong assumption of independence of "source" associated with periodic glottal excitation (for vowels) or turbulent noise (for fricatives) and "filter" associated with the form of the vocal tract. Such simplification leads to a very efficient algorithmic and hardware implementation of speech coding and speech synthesis based on linear filtering. For instance vocal tract transfer function can be approximated reasonably well with an all-pole linear filter of sufficient order. In case of linear predictive coding (LPC) coefficients of the filter can be calculated from the output of the feature generation neural network (e.g. from mel-spectrum generated by Tacoton). The remaining part (i.e. generating excitation signal) can be solved by another neural network. This is the core idea of LPCNet [15]. It has become quite popular [16] due to high quality of the synthesized speech and efficiency both in terms of time and memory.

We propose a modification of LPCNet aimed at improving efficiency even further without any degradation of speech quality. Although being conceptually the same, our modification has two major architecture differences from the model described in the original LPCNet paper. First, we generate 16-bit signal instead of 8-bit μ-companded signal. It allows us to get rid of the signal embedding matrix which is necessary for the original LPCNet since each of 256 possible values of signal is represented as a separate class. To enable 16-bit sampling we design our LPCNet so that it predicts mean and variance parameters of univariate Gaussian distribution $\mathcal{N}(\mu_t, \sigma_t^2)$. Thus, excitation e_t is sampled from $\mathcal{N}(\mu_t, \sigma_t^2)$ rather than from categorical distribution produced by the softmax layer. The second difference is that two consecutive excitation values are sampled at a time independently of each other. Although such approach makes an implicit assumption of conditional independence of two consecutive excitation values, it turns out that it does not have negative impact on the quality of the generated speech. At the same time, it significantly reduces time necessary for synthesis.

Throughout this paper we will refer to LPCNet model from the original paper as Original LPCNet and to our modification of the original architecture as Gaussian LPCNet, or Multisample Gaussian LPCNet. It's worth noting that in this modification only excitation e_t is generated in multisample and non-autoregressive way whereas we still need all M previous speech signal samples $s_{t-M+1}, .., s_t$ (where M is the order of LPC analysis) to generate the next sample s_{t+1} (see Section 3 for details).

Our paper is structured as follows: in Section 2 we give an overview of Original LPCNet model; in Section 3 we describe our proposed modifications that lead to efficiency improvement; in Section 4 we describe an experimental setup and compare Original and Gaussian LPCNet performance in terms of quality and efficiency.

978-1-5090-6631-5/20/$31.00 ©2020 IEEE

We conclude in Section 5.

2. ORIGINAL LPCNET

LPCNet utilizes Source-Filter model of speech production with Linear Predictive Coding [17] being used for speech signal analysis. An all-pole filter $H(z)$

$$H(z) = \frac{1}{1 - \sum_{k=1}^{M} a_k z^{-k}} \tag{1}$$

is known to be a good approximation of vocal tract transfer function. Such filter choice leads to a very simple signal representation:

$$s_t = e_t + p_t, \quad p_t = \sum_{k=1}^{M} a_k s_{t-k}, \tag{2}$$

where M is the order of LPC analysis, a_k are LPC coefficients, s_t is speech signal, p_t is its linear prediction and e_t is excitation, or residual signal.

2.1. General Architecture

Original LPCNet uses 18-band Bark-frequency cepstrum [18] to compute 16 LPC coefficients. Excitation is predicted by a combination of two neural networks: encoder and decoder. Encoder network operates on non-overlapping 10-ms frames and processes input vectors consisting of 18 Bark-scale cepstral coefficients and 2 pitch parameters (period and correlation). The processing is performed with one-dimensional convolutions and the result $\mathbf{f}$ is sent to the decoder. Decoder network operates at sample level (16 kHz speech signals are considered). It is based on two-layer Gated Recurrent Unit RNN (GRU [19]) and predicts excitation e_t based on acoustic feature vector $\mathbf{f}$ that comes from encoder and corresponds to the current frame, previous excitation e_{t-1}, previous signal value s_{t-1} and current linear prediction p_t. After e_t is generated, it is added to linear prediction p_t that is calculated using previous 16 signal samples $s_{t-16}, .., s_{t-1}$ and LPC coefficients $a_1, .., a_{16}$ obtained from 18 Bark-scale coefficients corresponding to the current frame. This algorithm is summarized in Figure 1.

2.2. Algorithm Details

Before feeding speech signal to Original LPCNet, a first order linear pre-emphasis filter $E(z)$ is applied to it:

$$E(z) = 1 - \alpha z^{-1} \tag{3}$$

The model produces excitation by sampling from categorical distribution parameterized by the outputs of the softmax layer in the decoder. It is impractical to use a large number of softmax classes, so it's necessary to apply some kind of quantization to signals which the model deals with. Original LPCNet uses non-linear 8-bit μ-law quantization. However, this quantization introduces audible noise into high frequencies. Applying pre-emphasis filter to the input and de-emphasizing (by applying inverse filter) output signal helps to reduce the perceived noise.

Since every signal value is represented as one of 256 classes, it is necessary to map each class to some embedding. Original LPCNet uses 128-dimensional embeddings.

As far as the most time-consuming part in this model (GRU layers in sample-level decoder) is concerned, pruning methods are used to make the least important weights equal to zeros. Since the second GRU layer $\mathrm{GRU_B}$ is significantly smaller than the first layer $\mathrm{GRU_A}$ (16 units vs 384 units), weight pruning is applied only to $\mathrm{GRU_A}$. This layer is pruned to have a specific sparse structure for efficient vectorization.

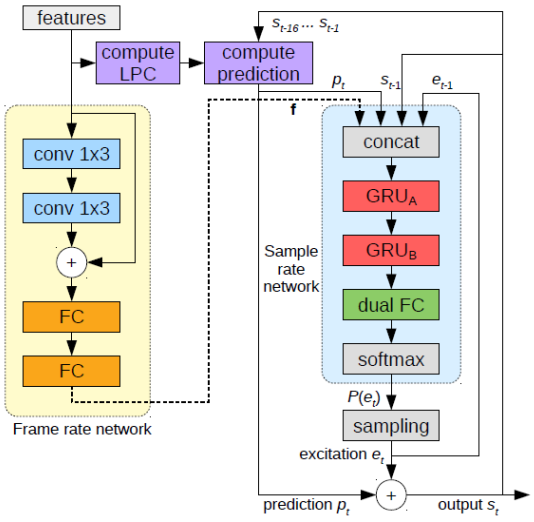

Fig. 1. Original LPCNet algorithm [15]. Conversion between μ-law and linear scales as well as pre-emphasizing and de-emphasizing are omitted for clarity.

The last detail we want to pay attention to is output distribution modification. When excitation is sampled directly from the output distribution, resulting speech signal has excessive noise (mostly clicking sounds). To cope with this issue, the authors propose to change the output distribution: probability of each class given by softmax layer is multiplied by some frame-level constant (that is dependent on pitch correlation in that frame) so as to lower the temperature of sampling process for voiced frames. Then, class probabilities are normalized to be a valid probability distribution. Finally, classes with probabilities smaller than some threshold are forced to have zero probability to rid excitation sampling procedure from unnecessary outliers.

3. MULTISAMPLE GAUSSIAN LPCNET

As we mentioned in the introduction, Gaussian LPCNet differs from Original model in two important aspects: it operates on 16-bit signal samples and it predicts two excitation samples at a time. It becomes possible due to architecture changes in LPCNet decoder. Figure 2 summarizes these changes. As for encoder, we don't modify it.

Since Gaussian LPCNet operates on signal that's not quantized, pre-emphasis filter is no longer necessary. Nevertheless, we tried training Multisample Gaussian LPCNet both on pre-emphasized speech signal and signal without pre-emphasis and found no difference in quality of resulting models.

3.1. Gaussian Distribution for 16-bit Prediction

While Parallel WaveNet [7] uses mixture of logistics output distribution for the teacher model in knowledge distillation scheme, ClariNet [6] authors show that a single univariate Gaussian distribution

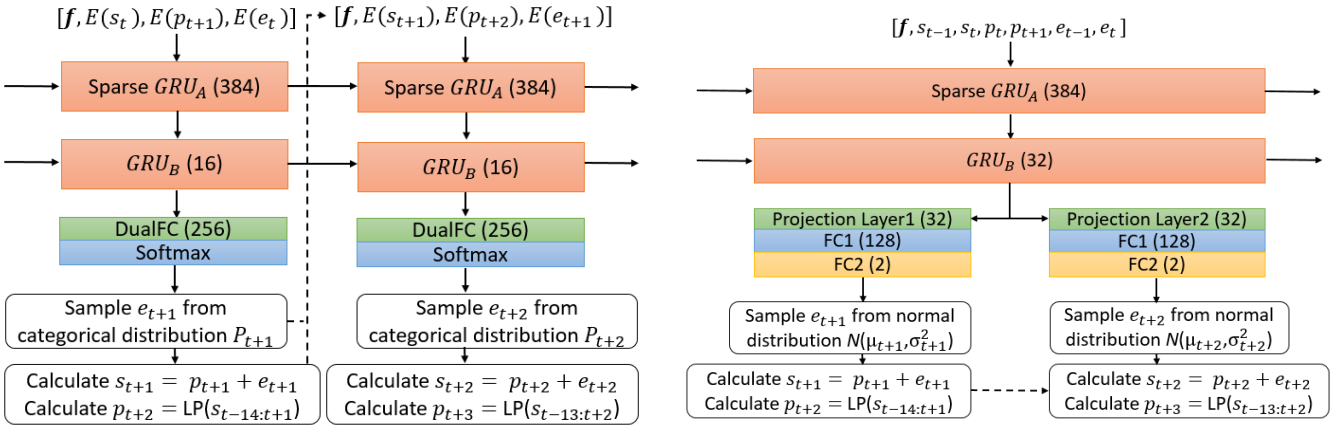

Fig. 2. Original LPCNet decoder (on the left) and Multisample Gaussian LPCNet decoder (on the right). Conversion between μ-law and linear scales in Original LPCNet decoder as well as output distribution modification in both decoders are omitted for clarity. 128-dimensional encoder output vector is denoted by $\mathbf{f}$. LP denotes calculating linear prediction based on frame-level LPC coefficients and formula (2). Numbers in parentheses stand for numbers of units in the corresponding layers. $E(\cdot)$ means 128-dimensional signal embedding.

is sufficient for high-quality modelling of speech signal samples in WaveNet architecture. Our studies also showed that in most cases Original LPCNet output excitation distribution is unimodal, so it's no use to model excitation with the mixture of some distributions. That's why in order to make our model suitable for generating 16-bit samples we experimented only with single continuous unimodal distributions (discrete categorical distribution is not a good choice because the number of classes is too big). Negative log-likelihood was chosen as loss function and 16-bit integer signal values were normalized to belong to $(-1.0, 1.0)$ interval for training purposes.

Laplace, Gaussian and generalized Gaussian distributions were considered. Generalized Gaussian density function is given by

$$p(x; \mu, \alpha, \beta) = \frac{\beta}{2\alpha\Gamma(1/\beta)} e^{-(|x-\mu|/\alpha)^\beta}, \quad (4)$$

where β and α are positive and $\Gamma(\cdot)$ is gamma-function. This distribution generalizes both Laplace and Gaussian distributions by adding a shape parameter β.

Laplace distribution performed worse than both Gaussian and generalized Gaussian which, in their turn, synthesized speech of the same quality. Since Gaussian distribution is easier to sample from than generalized Gaussian, we decided to go on with it.

Although excitation e_t should be sampled from Gaussian distribution $\mathcal{N}(\mu_t, \sigma_t^2)$, Gaussian LPCNet predicts parameters μ_t and $\log \sigma_t$ rather than μ_t and σ_t to avoid situations when negative value of σ_t is predicted by the network. Also, $\log \sigma_t$ is clipped by a constant -9 from below for numerical stability during training. These two tricks are borrowed from ClariNet [6].

3.2. Multisample excitation prediction

Even though the choice of continuous output distribution allows us not to use embedding matrix which significantly decreases number of parameters (mostly in input-to-hidden matrices in $\mathrm{GRU_A}$ layer), it has almost no effect on synthesis time because multiplication of $\mathrm{GRU_A}$ input-to-hidden matrix by an embedding vector is already highly optimized in Original LPCNet (by pre-computing matrix product of input-to-hidden matrix and embedding matrix). So, in order to reduce time complexity we change the architecture so as to predict two consecutive excitation samples at a time.

The modified architecture is summarized in Figure 2. We had to increase the number of $\mathrm{GRU_B}$ units to 32 since Multisample Gaussian LPCNet with 16 $\mathrm{GRU_B}$ units produced very noisy speech. Gaussian distribution parameters for excitation samples e_{t+1} and e_{t+2} are calculated as follows:

$$h_t^{(j)} = W_j h_t, \quad [\mu_{t+j}, \log \sigma_{t+j}]^\mathbf{T} = FC2(FC1(h_t^{(j)})) \quad (5)$$

where $j = 1, 2$ and h_t is the output of $\mathrm{GRU_B}$ layer. W_1 and W_2 are 32x32 projection matrices, fully connected layer FC1 has 128 units and $tanh$ activation function, fully connected layer FC2 has 2 units and no activation. Thus, in Multisample Gaussian LPCNet GRU is run once per two samples rather than once per each (similar approach is used in SampleRNN [20] for unconditional generation).

3.3. Gaussian Distribution Modification

If we sample excitation e_t directly from Gaussian distribution with parameters μ_t and σ_t^2 predicted by the network, synthesized speech will contain some amount of noise. As in Original LPCNet, this noise will mostly take shape of clicking sounds. So, we slightly modify the distribution that we sample from.

Output distribution modification consists of two stages. On the first one, we get rid of variance outliers:

$$\hat{\sigma_t} = \min\{\sigma_t, \sigma_{t-1}, .., \sigma_{t-n_\sigma+1}\}, \quad (6)$$

where n_σ is a hyperparameter tuned on validation set. In all of our experiments $n_\sigma = 8$ worked well. The idea behind such modification is the following: in general, variance seems to be hard to predict, so the network sometimes makes mistakes. Clicking sounds occur when the network predicts variances that are too large, that's why we need to avoid variance outliers that correspond to large values. We analyzed excitation signals and came to conclusion that in many cases excitation is homoscedastic (i.e. has the same finite variance) during 10-ms frame. Thus, it is quite easy to correct mistakes in variance by some robust aggregation of a small number n_σ of previous variance values. Actually, any aggregation function that is robust to

Table 1. Mean Opinion Scores for ground truth records and speech synthesized with two LPCNet models.

Dataset	Language	Gender	Duration	Listeners	Ground Truth	Original LPCNet	Gaussian LPCNet
LJSpeech [21]	English	Female	24h	50	4.80 ± 0.03	4.51 ± 0.04	$\mathbf{4.61 \pm 0.04}$
CSS10 [22]	German	Female	17h	20	4.54 ± 0.07	$\mathbf{4.48 \pm 0.07}$	4.40 ± 0.07
Internal	Russian	Male	26h	40	4.67 ± 0.04	4.52 ± 0.05	$\mathbf{4.61 \pm 0.04}$
CSS10	Spanish	Male	24h	30	4.81 ± 0.03	4.72 ± 0.04	$\mathbf{4.77 \pm 0.04}$
CSS10	French	Male	19h	20	4.54 ± 0.06	$\mathbf{4.49 \pm 0.07}$	4.43 ± 0.07

outliers with large values can be used (e.g. median), but we chose simply to compute minimum.

The second stage resembles the trick from Original LPCNet algorithm. Since we sample from $\mathcal{N}(\mu_t, {\hat{\sigma_t}}^2)$ a huge number of times (16k for a second of audio), extreme values of excitation e_t (e.g. near the boundaries -1.0 and 1.0) are sometimes generated even if predicted variance is close to correct. That's why instead of sampling from $\mathcal{N}(\mu_t, {\hat{\sigma_t}}^2)$ we sample from truncated normal distribution with the same parameters μ_t and $\hat{\sigma_t}$ and with support $[\mu_t - \hat{\sigma_t}, \mu_t + \hat{\sigma_t}]$ – it guarantees that we never generate excitation e_t that lies outside this interval. In practice, we use acceptance-rejection method [23] to sample from truncated normal distribution which is quite effective since acceptance ratio is very high (most of samples from $\mathcal{N}(\mu_t, {\hat{\sigma_t}}^2)$ lie inside the interval $[\mu_t - \hat{\sigma_t}, \mu_t + \hat{\sigma_t}]$).

4. EVALUATION

4.1. Human Evaluation

Results of subjective evaluation of two LPCNet algorithms are shown in Table 1. We chose crowd-sourcing platform Figure Eight to perform human evaluation and Mean Opinion Score (MOS) as target metric. LPCNet models synthesized speech conditioned on ground truth acoustic features rather than features generated by some backend model because our goal was to test vocoder quality only.

Original and Multisample Gaussian LPCNet models were trained on five different datasets representing different languages and speaker genders. Datasets contained short audio records (most of them between 4 and 12 seconds) downsampled to 16kHz sampling frequency. Hyperparameters (e.g. n_σ and number of units in $\mathrm{GRU_B}$ layer) were tuned on a validation set containing 10 utterances. Test set consisted of 20 held-out utterances for each language. Each of these utterances was synthesized by both Gaussian and Original LPCNet models. Ground truth 16kHz records were also added to the test set. Each of resulting 60 records was evaluated by 20-50 listeners (depending on the language) which were either chosen based on language criterion (only verified speakers of the language were allowed to participate in evaluation) or geographical criterion (all the listeners were selected from specified countries where the target language is official and spoken by the majority of population). Participants were asked to estimate quality of speech on five-point Likert scale, i.e. to classify a record as "Bad" (1 point), "Poor" (2 points), "Fair" (3 points), "Good" (4 points) or "Excellent" (5 points). Listeners were asked to pay attention to overall clarity of speech, to presence of background noise (e.g. clicking sounds) or other sonic artifacts and to correctness and naturalness of sounds pronunciation. Some utterances from validation set were also used in evaluation to check that listeners did not choose answers randomly: all participants who made more than one mistake on validation set (gave less than three points to ground truth records or more than three points to obviously bad sounding records that were generated for this specific purpose of testing listeners' attention) were excluded from experiment.

Table 2. Time and memory efficiency evaluation.

LPCNet	Total parameters	Non-zero parameters	RTF
Original	$1.240k$	$843k$	0.23
Gaussian	$796k$	$399k$	0.15

Mean Opinion Scores and 95% confidence intervals are reported in Table 1. The results show that Multisample Gaussian and Original LPCNet models synthesize speech of approximately the same quality. Also, we can conclude that Multisample Gaussian LPCNet is more sensitive to quality and amount of training data – we see that it performs better when trained on datasets with longer overall duration and better quality (as estimated by the listeners). Gaussian LPCNet operates on the whole continuum of excitation values rather than on discrete excitation space and probably this is the reason why it needs more data of better quality to perform better.

A small subset of speech samples used in subjective evaluation is available at `https://grog37.wixsite.com/li1jkdaw`.

4.2. Time and Memory Efficiency

To measure time efficiency of speech synthesis of two LPCNet architectures, we calculated real time factor (RTF). It is defined as time necessary to synthesize a piece of audio divided by duration of this audio. Speech samples were synthesized on 1.80GHz Intel CPU. Total number of non-zero parameters was calculated to evaluate memory efficiency. Note that for Multisample Gaussian LPCNet exactly the same pruning technique was applied to $\mathrm{GRU_A}$ layer, so it has the same sparse structure as $\mathrm{GRU_A}$ in Original LPCNet. The results presented in Table 2 show that Gaussian LPCNet is 1.5x faster and has twice less non-zero parameters.

5. CONCLUSION

In this work we've presented Multisample Gaussian LPCNet – a modified version of LPCNet vocoder that is significantly smaller, 1.5x faster and synthesizes speech of equally high quality. Multisample Gaussian LPCNet generates unquantized 16-bit speech signal and predicts two excitation samples at a time thus increasing the algorithm efficiency and making it even more attractive for mobile devices. Besides, released time and memory resources can be used to enhance backend model for better overall TTS performance.

Future work on Multisample Gaussian LPCNet includes increasing number of excitation samples generated at a time and making this model suitable for operating on speech signals with sampling frequency higher that 16kHz.

6. REFERENCES

[1] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu, "Natural TTS Synthesis by Conditioning Wavenet on Mel Spectrogram Predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 4779–4783.

[2] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu, "Neural Speech Synthesis with Transformer Network," in *AAAI*, 2018.

[3] Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu, "FastSpeech: Fast, Robust and Controllable Text to Speech," *ArXiv*, vol. abs/1905.09263, 2019.

[4] Aron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "WaveNet: A Generative Model for Raw Audio," in *9th ISCA Speech Synthesis Workshop*, 2016, pp. 125–125.

[5] K. Tachibana, T. Toda, Y. Shiga, and H. Kawai, "An Investigation of Noise Shaping with Perceptual Weighting for Wavenet-Based Speech Generation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 5664–5668.

[6] Wei Ping, Kainan Peng, and Jitong Chen, "ClariNet: Parallel Wave Generation in End-to-End Text-to-Speech," *CoRR*, vol. abs/1807.07281, 2018.

[7] Aaron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis Cobo, Florian Stimberg, Norman Casagrande, Dominik Grewe, Seb Noury, Sander Dieleman, Erich Elsen, Nal Kalchbrenner, Heiga Zen, Alex Graves, Helen King, Tom Walters, Dan Belov, and Demis Hassabis, "Parallel WaveNet: Fast High-Fidelity Speech Synthesis," in *Proceedings of the 35th International Conference on Machine Learning*, Jennifer Dy and Andreas Krause, Eds., Stockholmsmssan, Stockholm Sweden, 10–15 Jul 2018, vol. 80 of *Proceedings of Machine Learning Research*, pp. 3918–3926, PMLR.

[8] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A Flow-based Generative Network for Speech Synthesis," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 3617–3621.

[9] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling, "Improved Variational Inference with Inverse Autoregressive Flow," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds., pp. 4743–4751. Curran Associates, Inc., 2016.

[10] Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron van den Oord, Sander Dieleman, and Koray Kavukcuoglu, "Efficient Neural Audio Synthesis," in *Proceedings of the 35th International Conference on Machine Learning*, Jennifer Dy and Andreas Krause, Eds., Stockholmsmssan, Stockholm Sweden, 10–15 Jul 2018, vol. 80 of *Proceedings of Machine Learning Research*, pp. 2410–2419, PMLR.

[11] Zeyu Jin, Adam Finkelstein, Gautham J. Mysore, and Jingwan Lu, "FFTNet: a Real-Time Speaker-Dependent Neural Vocoder," in *The 43rd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018.

[12] Eunwoo Song, Kyungguen Byun, and Hong-Goo Kang, "ExcitNet vocoder: A neural excitation model for parametric speech synthesis systems," 2018.

[13] Lauri Juvela, Bajibabu Bollepalli, Junichi Yamagishi, and Paavo Alku, "GELP: GAN-Excited Linear Prediction for Speech Synthesis from Mel-spectrogram," 2019.

[14] Bajibabu Bollepalli, Lauri Juvela, and Paavo Alku, "Generative Adversarial Network-Based Glottal Waveform Model for Statistical Parametric Speech Synthesis," *Interspeech 2017*, Aug 2017.

[15] J. Valin and J. Skoglund, "LPCNet: Improving Neural Speech Synthesis through Linear Prediction," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 5891–5895.

[16] Zvi Kons, Slava Shechtman, Alex Sorin, Carmel Rabinovitz, and Ron Hoory, "High quality, lightweight and adaptable TTS using LPCNet," 2019.

[17] Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2001.

[18] B.C.J. Moore, *An introduction to the psychology of hearing*, Brill, fifth edition, 2012.

[19] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio, "On the Properties of Neural Machine Translation: Encoder–Decoder Approaches," in *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, Doha, Qatar, Oct. 2014, pp. 103–111, Association for Computational Linguistics.

[20] Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron C. Courville, and Yoshua Bengio, "SampleRNN: An Unconditional End-to-End Neural Audio Generation Model," *CoRR*, vol. abs/1612.07837, 2016.

[21] Keith Ito, "The LJ Speech Dataset," 2017.

[22] Kyubyong Park and Thomas Mulc, "CSS10: A Collection of Single Speaker Speech Datasets for 10 Languages," in *Proc. Interspeech 2019*, 2019, pp. 1566–1570.

[23] Paul Glasserman, "Monte Carlo Methods in Financial Engineering," 2003.