International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 07 | July 2022 www.irjet.net p-ISSN: 2395-0072

Rendering Of Voice By Using Convolutional Neural Network And With The Help Of Text-To-Speech Module

Dhiren Patel, Aashay Ingle, Saurabh Joshi, Mandar Hegiste

[1].*B.sc Computer Science Graduate, Mumbai University, Mithibai College, Maharashtra - 400056*
[2].*Mechanical Engineering Graduate, Mumbai University, Viva College, Maharashtra - 401305*
[3].*Software Engineer Graduate, Mumbai University, Xavier Institute of Technology, Maharashtra*
[4].*Mechanical Engineering Graduate, Mumbai University, Viva College, Maharashtra - 401305*

---***---

Abstract - *This paper describes a novel text-to-speech (TTS) technique based on deep convolutional neural networks (CNN), without any recurrent units. Recurrent neural network (RNN) has been a standard technique to model sequential data recently, and this technique has been used in some cutting-edge neural TTS techniques. However, training RNN component often requires a very powerful computer, or very long time typically several days or weeks. Recent other studies, on the other hand, have shown that CNN-based sequence synthesis can be much faster than RNN-based techniques, because of high parallelizability. The objective of this paper is to show an alternative neural TTS system, based only on CNN, that can alleviate these economic costs of training. In our experiment, the proposed Deep Convolutional TTS can be sufficiently trained only in a night (15 hours), using an ordinary gaming PC equipped with two GPUs, while the quality of the synthesized speech was almost acceptable.*

Key Words: **Text-to-speech, deep learning, convolutional neural network, attention, sequence-to-sequence learning.**

1. INTRODUCTION

Text-to-speech (TTS) is getting more and more common recently, and is getting to be a basic user interface for many systems. Toencourage further use of TTS in various systems, it is significant to develop a handy, maintainable, extensible TTS component that is accessible to speech non-specialists, enterprising individuals and small teams who do not have massive computers.

Traditional TTS systems, however, are not necessarily friendly for them, as these systems are typically composed of many domain- specific modules. For example, a typical parametric TTS system is an elaborate integration of many modules e.g. a text analyzer

$F0$ generator, a spectrum generator, a pause estimator, and a vocoder that synthesize a waveform from these data, etc.

Deep learning sometimes can unite these internal building blocks into a single model, and directly connects the input and the output; this type of technique is sometimes called 'end-to-end' learning. Although such a technique is sometimes criticized as 'a black box,' nevertheless, an end-to-end TTS system named Tacotron, which directly estimates a spectrogram from an in- put text, has achieved promising performance recently, without intensively-engineered parametric models based on domain-specific knowledge. Tacotron, however, has a drawback that it exploits many recurrent units, which are quite costly to train, making it almost infeasible for ordinary labs without luxurious machines to study and extend it further. Indeed, some people tried to implement open clones of Tacotron but they are struggling to reproduce the speech of satisfactory quality as clear as the original work.

The purpose of this paper is to show Deep Convolutional TTS (DCTTS), a novel, handy neural TTS, which is fully convolutional. The architecture is largely similar to Tacotron but is based on a fully convolutional sequence-to-sequence learning model similar to the literature We show this handy TTS actually works in a reasonable setting. The contribution of this article is twofold: (1) Propose a fully CNN-based TTS system which can be trained much faster than an RNN-based state-of-the-art neural TTS system, while the sound quality is still acceptable. (2) An idea to rapidly train the attention, which we call 'guided attention,' is also shown.

1.1 Related Work

Neural speech synthesis: Recently, there is a surge of interest in speech synthesis with neural networks, including Deep Voice 1 [Arik et al., 2017a], Deep Voice 2 [Arik et al., 2017b], Deep Voice 3 [Ping et al., 2018], WaveNet [Oord et al., 2016a], SampleRNN [Mehri et al., 2016], Char2Wav [Sotelo et al., 2017], Tacotron [Wang et al., 2017] and VoiceLoop [Taigman et al., 2018]. Among these methods, sequence-to-sequence models [Ping et al., 2018, Wang et al., 2017, Sotelo et al., 2017] with attention mechanism have much simpler pipeline and can produce more natural speech [e.g., Shen et al., 2017]. In this work, we use Deep Voice 3 as the baseline multi-speaker model, because of its simple convolutional architecture and high efficiency for training and fast model adaptation. It should be noted that our techniques can be seamlessly applied to other neural speech synthesis models.

Few-shot generative modeling: Humans can learn new generative tasks from only a few examples, which motivates research on few-shot generative models. Early studies mostly focus on Bayesian methods. For example, hierarchical Bayesian models are used to exploit compositionality and causality for few-shot generation of characters [Lake et al., 2013, 2015] and words in speech [Lake et al., 2014]. Recently, deep neural networks achieve great successes in few-shot density estimation and conditional image generation [e.g., Rezende et al., 2016, Reed et al., 2017, Azadi et al., 2017], because of the great potential for composition in their learned representation. In this work, we investigate few-shot generative modeling of speech conditioned on a particular speaker. We train a separate speaker encoding network to directly predict the parameters of multi-speaker generative model by only taking unsubscribed audio samples as inputs.

Speaker-dependent speech processing: Speaker-dependent modeling has been widely studied for automatic speech recognition (ASR), with the goal of improving the performance by exploiting speaker characteristics. In particular, there are two groups of methods in neural ASR, in alignment with our two voice cloning approaches. The first group is speaker adaptation for the whole-model [Yu et al., 2013], a portion of the model [Miao and Metze, 2015, Cui et al., 2017], or merely to a speaker embedding [Abdel-Hamid and Jiang, 2013, Xue et al., 2014]. Speaker adaptation for voice cloning is in the same vein as these approaches, but differences arise when text-to-speech vs. speech-to-text are considered [Yamagishi et al., 2009]. The second group is based on training ASR models jointly with embeddings. Extraction of the embeddings can be based on i-vectors [Miao et al., 2015], or bottleneck layers of neural networks trained with a classification loss [Li and Wu, 2015]. Although the general idea of speaker encoding is also based on extracting the embeddings directly, as a major distinction, our speaker encoder models are trained with an objective function that is directly related to speech synthesis. Lastly, speaker-dependent modeling is essential for multi-speaker speech synthesis. Using i-vectors to represent speaker-dependent characteristics is one approach [Wu et al., 2015], however, they have the limitation of being separately trained, with an objective that is not directly related to speech synthesis. Also they may not be accurately extracted with small amount of audio [Miao et al., 2015]. Another approach for multi-speaker speech synthesis is using trainable speaker embeddings [Arik et al., 2017b], which are randomly initialized and jointly optimized from a generative loss function.

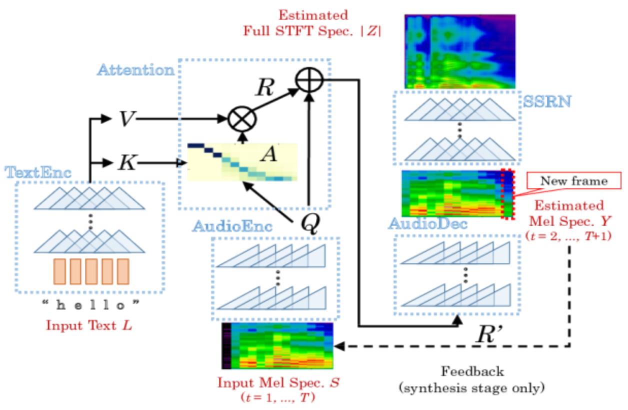
Voice conversion: A closely related task of voice cloning is voice conversion. The goal of voice conversion is to modify an utterance from source speaker to make it sound like the target speaker, while keeping the linguistic contents unchanged. Unlike voice cloning, voice conversion systems do not need to generalize to unseen texts. One common approach is dynamic frequency warping, to align spectra of different speakers. Agiomyrgiannakis and Roupakia [2016] proposes a dynamic programming algorithm that simultaneously estimates the optimal frequency warping and weighting transform while matching source and target speakers using a matching-minimization algorithm. Wu et al. [2016] uses a spectral conversion approach integrated with the locally linear embeddings for manifold learning. There are also approaches to model spectral conversion using neural networks [Desai et al., 2010, Chen et al., 2014, Hwang et al., 2015]. Those models are typically trained with a large amount of audio pairs of target and source speakers.

1.2 Methodology

Our DCTTS model consists of two networks: (1) Text2Mel, which synthesize a mel spectrogram from an input text, and (2) Spectrogram Super-resolution Network (SSRN), which convert a coarse mel spectrogram to the full STFT spectrogram. Fig. 1 shows the overall architecture of the proposed method. 3.1. Text2Mel: Text to Mel Spectrogram Network We first consider to synthesize a coarse mel spectrogram from a text. This is the main part of the proposed method. This module consists of four submodules: Text Encoder, Audio Encoder, Attention, and Audio Decoder. The network TextEnc first encodes the input sentence $L = [l1, \ldots, lN] \in CharN$ consisting of N characters, into the two matrices $K, V \in R\ d\times N$. On the other hand, the network AudioEnc encodes the coarse mel spectrogram $S(= S1:F,1:T)$ $\in R\ F \times T$, of previously spoken speech, whose length is T, into a matrix $Q \in R\ d\times T$. $(K, V) = TextEnc(L)$. (1) $Q = AudioEnc(S1:F,1:T)$. (2) An attention matrix $A \in R\ N\times T$, defined as follows, evaluates how strongly the n-th character ln and t-th time frame S1:F,t are related, $A = softmaxn\text{-}axis(K\ TQ/\sqrt{}\ d)$. (3) $Ant \sim 1$ implies that the module is looking at n-th character ln at the time frame t, and it will look at ln or ln+1 or characters around them, at the subsequent time frame t + 1. Whatever, let us expect those are encoded in the n-th column of V . Thus a seed $R \in R\ d\times T$, decoded to the subsequent frames S1:F,2:T +1, is obtained as $R = Att(Q, K, V) := V\ A$. (Note: matrix product.) (4) The resultant R is concatenated with the encoded audio Q, as $R\ 0 = [R, Q]$, because we found it beneficial in our pilot study. Then, the concatenated matrix $R\ 0 \in R\ 2d\times T$ is decoded by the Audio Decoder module to synthesize a coarse mel spectrogram, $Y1:F,2:T +1 = AudioDec(R\ 0)$. (5) The result Y1:F,2:T +1 is compared with the temporally-shifted ground truth S1:F,2:T +1, by a loss function Lspec(Y1:F,2:T +1|S1:F,2:T +1), and the error is back-propagated to the network parameters. The loss function was the sum of L1 loss and the binary divergence Dbin, $Dbin(Y\ |S) := Eft[-Sft \log Yft - (1 - Sft) \log(1 - Yft)] = Eft[-SftY\hat{}ft + \log(1 + \exp Y\hat{}ft)]$, (6) where $Y\hat{}ft = logit(Yft)$. Since the binary divergence gives a nonvanishing gradient to the network, $\partial Dbin(Y\ |S)/\partial Y\hat{}ft \propto Yft - Sft$, it is advantageous in gradient-based training. It is easily verified that the spectrogram error is non-negative, $Lspec(Y\ |S) = Dbin(Y\ |S) + E[|Yft - Sft|] \geq 0$, and the equality holds iff $Y = S$.

3.1.1. Details of TextEnc, AudioEnc, and AudioDec Our networks are fully convolutional, and are not dependent on any recurrent units. Instead of RNN, we sometimes take advantages of dilated convolution [32, 13, 24] to take long contextual information into account. The top equation of Fig. 2 is the content of TextEnc. It consists of the character embedding and the stacked 1D non-causal convolution. A previous literature [2] used a heavier RNN-based component named 'CBHG,' but we found this simpler network also works well. AudioEnc and AudioDec, shown in Fig. 2, are composed of 1D causal convolution layers with Highway activation. These convolution should be causal because the output of AudioDec is feedbacked to the input of AudioEnc in the synthesis stage. 3.2. Spectrogram Super-resolution Network (SSRN) We finally synthesize a full spectrogram $|Z| \in R F 0{\times}4T$, from the obtained coarse mel spectrogram $Y \in R$ $F \times T$, by a spectrogram super-resolution network (SSRN). Upsampling frequency from F to F 0 is rather straightforward. We can achieve that by increasing the convolution channels of 1D convolutional network. Upsampling in temporal direction is not similarly done, but by twice applying deconvolution layers of stride size 2, we can quadruple the length of sequence from T to $4T = T 0$. The bottom equation of Fig. 2 shows SSRN. In this paper, as we do not consider online processing, all convolutions can be non-causal. The loss function was the same as Text2Mel: sum of binary divergence and L1 distance between the synthesized spectrogram SSRN(S) and the ground truth |Z|.

Figure -1: Guided Attention



Guided Attention Loss: Motivation, Method and Effects In general, an attention module is quite costly to train. Therefore, if there is some prior knowledge, it may be a help incorporating them into the model to alleviate the heavy training. We show that the simple measure below is helpful to train the attention module. In TTS, the possible attention matrix A lies in the very small subspace of $R\ N{\times}T$. This is because of the rough correspondence of the order of the characters and the audio segments. That is, if one reads a text, it is natural to assume that the text position n progresses nearly linearly to the time t, i.e., $n \sim at$, where $a \sim N/T$. This is Fig. 3. Comparison of the attention matrix A, trained with and without the guided attention loss Latt(A). (Left) Without, and (Right) with the guided attention. The test text is "icassp stands for the international conference on acoustics, speech, and signal processing." We did not use the heuristics described in section 4.2. the prominent difference of TTS from other seq2seq learning techniques such as machine translation, in which an attention module should resolve the word alignment between two languages that have very different syntax, e.g. English and Japanese. Based on this idea, we introduce another constraint on the attention matrix A to prompt it to be 'nearly diagonal,' Latt(A) = Ent[AntWnt], where Wnt = 1 − exp{−(n/N − t/T) 2 /2g 2 }. In this paper, we set g = 0.2. If A is far from diagonal (e.g., reading the characters in the random order), it is strongly penalized by the loss function. This subsidiary loss function is simultaneously optimized with the main loss Lspec with equal weight. Although this measure is based on quite a rough assumption, it improved the training efficiency. In our experiment, if we added the guided attention loss to the objective, the term began decreasing only after ∼100 iterations. After ∼5K iterations, the attention became roughly correct, not only for training data, but also for new input texts. On the other hand, without the guided attention loss, it required much more iterations. It began learning after ∼10K iterations, and it required ∼50K iterations to look at roughly correct positions, but the attention matrix was still vague. Fig. 3 compares the attention matrix, trained with and without guided attention loss. 4.2. Forcibly Incremental Attention in Synthesis Stage In the synthesis stage, the attention matrix A sometimes fails to look at the correct characters. Typical errors we observed were (1) it occasionally skipped several characters, and (2) it repeatedly read a same word twice or more. In order to make the system more robust, we heuristically modify the matrix A to be 'nearly diagonal,' by a simple rule as follows. We observed this device sometimes alleviated such misattentions. Let nt be the position of the character to be read at tth time frame; nt = argmaxnAn,t. Comparing the current position nt and the previous position nt−1, unless the difference nt − nt−1 is within the range −1 ≤ nt − nt−1 ≤ 3, the current attention position forcibly set to An,t = δn,nt−1+1 (Kronecker's delta), to forcibly make the attention target incremental, i.e., nt − nt−1 = 1

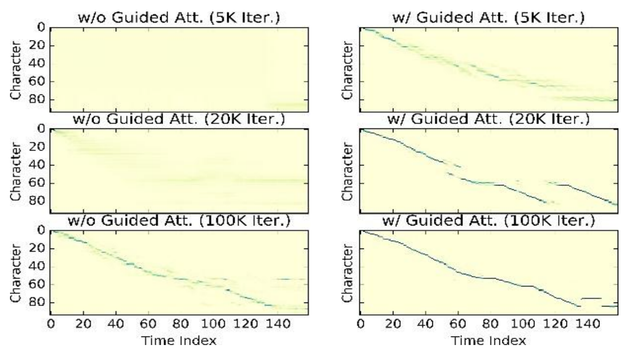


Figure 2 : Forcibly Incremental Attention

Forcibly Incremental Attention in Synthesis Stage In the synthesis stage, the attention matrix A sometimes fails to look at the correct characters. Typical errors we observed were (1) it occasionally skipped several characters, and (2) it repeatedly read a same word twice or more. In order to make the system more robust, we heuristically modify the matrix A to be 'nearly diagonal,' by a simple rule as follows. We observed this device sometimes alleviated such misattentions. Let nt be the position of the character to be read at tth time frame; nt = argmaxnAn,t. Comparing the current position nt and the previous position nt−1, unless the difference nt − nt−1 is within the range −1 ≤ nt − nt−1 ≤ 3, the current attention position forcibly set to An,t = δn,nt−1+1 (Kronecker's delta), to forcibly make the attention target incremental, i.e., nt − nt−1 = 1.

2. SUMMARY AND FUTURE WORK

This paper described a novel text-to-speech (TTS) technique based on deep convolutional neural networks (CNN), as well as a technique to train the attention module rapidly. In our experiment, the proposed Deep Convolutional TTS can be sufficiently trained only in a night (~15 hours), using an ordinary gaming PC equipped with two GPUs, while the quality of the synthesized speech was almost acceptable. Although the audio quality is far from perfect yet, it may be improved by tuning some hyper-parameters thoroughly, and by applying some techniques developed in deep learning community. We believe this handy method encourages further development of the applications based on speech synthesis. We can expect this simple neural TTS may be extended to other versatile purposes, such as emotional/non-linguistic/personalized speech synthesis, singing voice synthesis, music synthesis, etc., by further studies. In addition, since a neural TTS has become this lightweight, the studies on more integrated speech systems e.g. some multimodal systems, simultaneous training of TTS+ASR, and speech translation, etc., may have become more feasible. These issues should be worked out in the future.

3. CONCLUSIONS

This paper implements a method to clone and render voice using Convolutional Neural Networks instead of widely-used traditional complete Neural Networks which helps to reduce training time significantly. We also implement an additional system of 'attention' which helps increase precision in case of long sentences. Although, fine parametric tuning can be performed to further increase the output quality and accuracy, there is surely a comparable scope with existing systems in the near future..

We are aiming to also classify the type of plasmodium parasite for which we would use machine learning technologies.

REFERENCES

[1] I. Goodfellow et al., Deep Learning, MIT Press, 2016, http: //www.deeplearningbook.org.

[2] Y. Wang et al., "Tacotron: Towards end-to-end speech synthesis," in Proc. Interspeech, 2017, arXiv:1703.10135. [3] A. Barron, "Implementation of Google's Tacotron in TensorFlow," 2017, Available at GitHub, https://github. com/barronalex/Tacotron (visited Oct. 2017).

[4] K. Park, "A TensorFlow implementation of Tacotron: A fully end-to-end text-to-speech synthesis model," 2017, Available at GitHub, https://github.com/Kyubyong/ tacotron (visited Oct. 2017).

[5] K. Ito, "Tacotron speech synthesis implemented in TensorFlow, with samples and a pre-trained model," 2017, Available at GitHub, https://github.com/keithito/ tacotron (visited Oct. 2017).

[6] R. Yamamoto, "PyTorch implementation of Tacotron speech synthesis model," 2017, Available at GitHub, https:// github.com/r9y9/tacotron_pytorch (visited Oct. 2017).

[7] J. Gehring et al., "Convolutional sequence to sequence learning," in Proc. ICML, 2017, pp. 1243–1252, arXiv:1705.03122.

[8] H. Zen et al., "Statistical parametric speech synthesis using deep neural networks," in Proc. ICASSP, 2013, pp. 7962–7966.

[9] Y. Fan et al., "TTS synthesis with bidirectional LSTM based recurrent neural networks," in Proc. Interspeech, 2014, pp. 1964–1968.

[10] H. Zen and H. Sak, "Unidirectional long short-term memory recurrent neural network with recurrent output layer for lowlatency speech synthesis," in Proc. ICASSP, 2015, pp. 4470– 4474.

[11] S. Achanta et al., "An investigation of recurrent neural network architectures for statistical parametric speech synthesis.," in Proc. Interspeech, 2015, pp. 859–863.

[12] Z. Wu and S. King, "Investigating gated recurrent networks for speech synthesis," in Proc. ICASSP, 2016, pp. 5140–5144.

[13] A. van den Oord et al., "WaveNet: A generative model for raw audio," arXiv:1609.03499, 2016.