IE643: Deep Learning : Theory and Practice **Aug-Dec 2020**

Paper: DeepVoice 2: Multi-Speaker Neural Text-to-Speech

Lecturer: P. Balamurugan *Scribes: Shashank Kumar*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

Motivation

Text-to-speech (TTS) synthesis has a variety of applications ranging from communication devices, digitally talking bots to media broadcasting like smart TVs and even to IoT. The technology is expanding to other industries like finance, health care, e-learning, telecommunication and robotics. Most TTS systems are built for a single artificial voice and multi-speaker voice systems require much more data and efforts.

Introduction

The paper proposes to an all-neural multi-speaker TTS system. The model aims to solve the multi-speaker TTS problem by using a single model, and a lot less data per speaker when compared to single-speaker systems. The paper is based on the following two elements -

1. **DeepVoice 2** [**1**], an architecture improved upon **DeepVoice 1** [**3**]
2. A **WaveNet-based** [**5**] spectrogram-to-audio neural vocoder which is used with **Tacotron** [**6**]

The general structure is the same as DeepVoice 1. The neural network creates trainable speaker embeddings into DeepVoice2 and Tacotron using the single speaker models as a baseline. The network to capture speaker embeddings for the speaker characteristics is trained from scratch.

Methodologies

The text is first converted into phonemes using pronunciation dictionary, which are then fed for duration prediction, upscaling and generating the frequency $F0$. This frequency is finally passed to the vocal model for speech synthesis.

For the multi speaker task, each of the models is augmented with a single low-dimensional speaker embedding vector per speaker, thus allowing near-complete weight sharing between speakers. Each model has its own set of speaker embeddings, which are incorporated into multiple portions of the model. The following sections describe the models proposed by the paper for the TTS task.

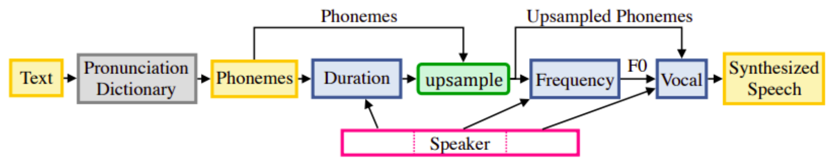

Figure 0.1: Inference system diagram

Segmentation Model

Estimation of phoneme locations is formulated as an unsupervised learning task. The segmentation model uses a **convolutional-recurrent architecture** with **connectionist-temporal-classification (CTC) loss** [**7**]. The phoneme pairs are first classified and then the boundary is extracted between them. The major change in DeepVoice 2 as compared to DeepVoice 1 is the use of batch-normalization and residual connections in the convolutional layers.

$$h^{(l)} = relu(h^{(l)} + BN(W^{(l)} * h^{(l-1)}))$$

where $h^{(l)}$ is the output of the l-th layer, $W^{(l)}$ is the convolutional filterbank, $*$ is the convolutional operation and BN is the **batch-normalization operation** [**8**].
For the multi-speaker task, batch-normalized activations are multiplied by site-specific speaker embedding g_s, which is shared by all the layers.

$$h^{(l)} = relu(h^{(l)} + BN(W^{(l)} * h^{(l-1)}) \cdot g_s)$$

Duration Model

Duration prediction is modelled as a sequence labeling problem using **conditional random field (CRF)** [**9**]. The phonemes are assigned into log-scaled buckets. Quantizing the duration prediction and introducing pairwise dependence implied by CRF improves synthesis quality.
The multi-speaker model uses speaker-dependent t recurrent initialization and input augmentation. A site-specific embedding is used to initialize RNN hidden states, and another site-specific embedding is provided as input to the first RNN layer by concatenating it to the feature vectors.

Frequency Model

The predicted phoneme durations are upsampled from a per-phoneme input features to a per-frame input for the frequency model. Each frame is of 10ms. Phonemes are either extended or split into multiple frames so as to maintain the size of 10ms. DeepVoice 2 consists of two layers. Firstly, f_{GRU} is predicted by a single layer bidirectional **gated recurrent unit (GRU)** [**10**] followed by affine projection. The second prediction, f_{CONV} is made by adding up the contributions of multiple convolutions with varying convolution widths and a single output channel.

$$f = \omega \cdot f_{GRU} + (1 - \omega) \cdot f_{CONV}$$

where ω is the mixture ratio predicted from the hidden state using an affine projection and a sigmoid activation. The normalized prediction f is converted is converted to true frequency F_0 prediction as follows.

$$F_0 = \mu_{F_0} + \sigma_{F_0} \cdot f$$

where μ_{F_0} and σ_{F_0} are the mean and standard deviation of F_0, respectively.
The multi-speaker model uses recurrent intialization with a single site-specific speaker-embedding. The mean (μ_{F0}) and standard deviation (σ_{F_0}) are made as trainable parameters.

$$F_0 = \mu_{F_0} \cdot (1 + softsign(V_\mu^T g_f)) + \sigma_{F_0} \cdot (1 + softsign(V_\sigma^T g_f)) \cdot f$$

where g_f is a site-specific speaker embedding, μ_{F_0} and σ_{F_0} are trainable scalar parameters initialized to the F_0 mean and standard deviation on the dataset, and V_μ and V_σ are trainable parameter vectors.

Vocal Model

The vocal model is based on **WaveNet architecture** with a two-layer bidirectional **QRNN [11]** conditioning network. The 1 x 1 convolution between the gated tanh nonlinearity and the residual connection is removed and the same conditioner bias for every layer of the WaveNet is used, instead of generating a separate bias for every layer as was done in Deep Voice 1.
The multi-speaker vocal model uses only input augmentation, with the site-specific speaker embedding concatenated onto each input frame of the conditioner

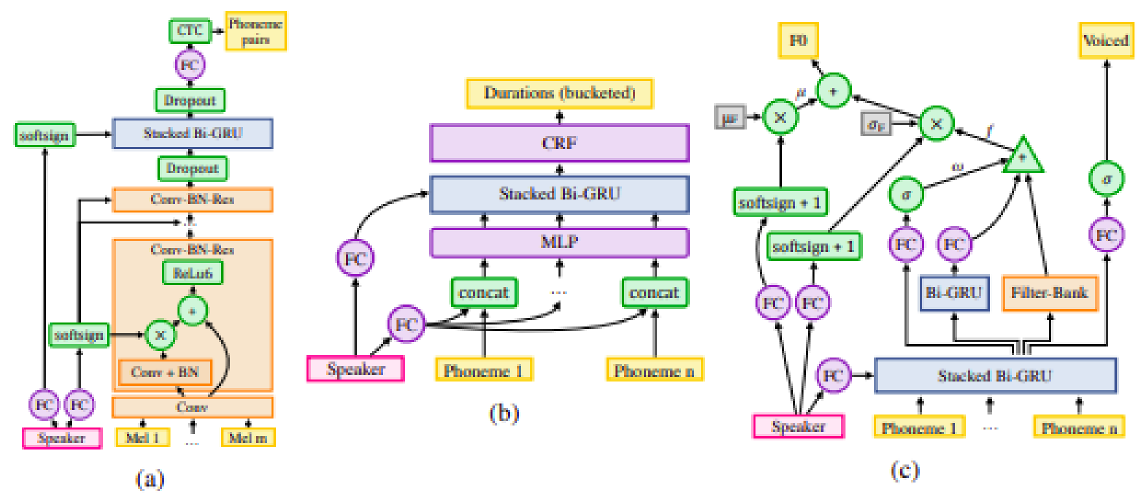

Figure 0.2: Multi-speaker architecture for (a)segmentation, (b) duration (c) frequency models

Experiments

- All the aforementioned models are trained on the **VCTK dataset** with 44 hours of speech, which contains 108 speakers with approximately 400 utterances each. The models are also trained on an internal dataset of audiobooks, which contains 477 speakers with 30 minutes of audio each (for a total of 238 hours).
- To incorporate each speaker's unique voice signature to influence the model, the speaker embeddings are injected into multiple portions of the model. The following patterns tend to improve the performance.

  - **Site-Specific Speaker Embeddings:** For every use site in the model architecture, transform the shared speaker embedding to the appropriate dimension and form through an affine projection and a non-linearity.
  - **Recurrent Initialization:** Initialize recurrent layer hidden states with site-specific speaker embeddings.
  - **Input Augmentation:** Concatenate a site-specific speaker embedding to the input at every timestep of a recurrent layer.
  - **Feature Gating:** Multiply layer activations element-wise with a site-specific speaker embedding to render adaptable information flow.

- In the **segmentation model**, the same site-specific embedding is shared for all the convolutional layers. In addition, the recurrent layers are initialized with a second site specific embedding. Each layer shares the same site-specific embedding, rather than having a separate embedding per layer.
- The **vocal model** is able to generate somewhat distinct-sounding voices even without speaker embeddings because of the distinctive features provided by the frequency and duration models. But having speaker embeddings in the vocal model increases the audio quality significantly.
- When training **multi-speaker Tacotron** variants, the model performance is highly dependent on model hyperparameters, and that some models often fail to learn attention mechanisms for a small subset of speakers. Due to the sensitivity of the model to hyperparameters and data.
- If the speech in each audio clip does not start at the same timestep, the models are much less likely to converge to a meaningful attention curve and recognizable speech. Thus, all initial and final silence is trimmed in each audio clip
- The **Tacotron character-to-spectrogram architecture** consists of a convolution-bank-highway-GRU (CBHG) encoder, an attentional decoder, and a CBHG post-processing network. Incorporating speaker embeddings into the CBHG post-processing network degrades output quality, whereas incorporating speaker embeddings into the character encoder is necessary. Augmenting the attentional decoder with speaker embeddings is helpful.
- A **Wavenet-based neural vocoder** produces higher quality audio from linear spectrograms than the original **Griffin-Lim algorithm** used in Tacotron. The architecture for the same is as shown below.

Results

Single-speaker speech synthesis

To compare the results, MOS evaluations using the **CrowdMOS framework [12]** is used. The results show conclusively that the DeepVoice 2 architecture improves the quality significantly over DeepVoice 1. The table also demonstrate that converting Tacotron-generated spectrograms to audio using WaveNet is preferable to using the iterative Griffin-Lim algorithm. The results are summarized in the table below with 95% confidence intervals.

Model	Sample Frequency	MOS
DeepVoice 1	16KHz	2.05 $\pm$0.24
DeepVoice 2	16KHz	2.96 $\pm$0.38
Tacotron (Griffin-Lim)	24KHz	2.57 $\pm$0.28
Tacotron (WaveNet)	24KHz	4.17 $\pm$0.18

Multi-speaker speech synthesis

MOS evaluations are used as in the single-speaker case in order to evaluate the quality of the synthesized audio, framework, and the results are summarized below. The Deep Voice 2 model can approach an MOS value that is close to the ground truth, when low sampling rate and companding/expanding taken into account.

Dataset	Multi-Speaker Model	Sample Frequency	MOS	Accuracy
VCTK	DeepVoice 2 (20-layer WaveNet)	16KHz	2.8 $\pm$0.13	99.9%
VCTK	DeepVoice 2 (40-layer WaveNet)	16KHz	3.21 $\pm$0.13	100%
VCTK	DeepVoice 2 (60-layer WaveNet)	16KHz	3.42 $\pm$0.12	99.7%
VCTK	DeepVoice 2 (80-layer WaveNet)	16KHz	3.53 $\pm$0.12	99.9%
VCTK	Tacotron (Griffin-Lim)	24KHz	1.68 $\pm$0.12	99.4%
VCTK	Tacotron (20-layer WaveNet)	24KHz	2.51 $\pm$0.13	60.9%
VCTK	Ground Truth Data	48KHz	4.65 $\pm$0.06	99.7%
Audiobooks	Deep Voice 2 (80-layer WaveNet)	16KHz	2.97 $\pm$0.17	97.4%
Audiobooks	Tacotron (Griffin-Lim)	24KHz	1.73 $\pm$0.22	93.9%
Audiobooks	Tacotron (20-layer WaveNet)	24KHz	2.11 $\pm$0.20	66.5%
Audiobooks	Ground Truth Data	44.1KHz	4.63 $\pm$0.04	98.8%

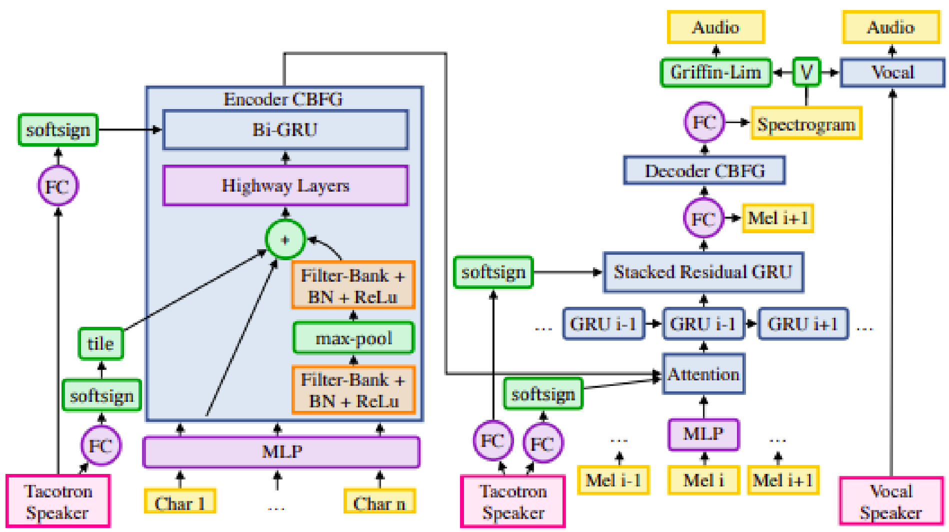

Figure 0.3: Tacotron with speaker conditioning in the encoder CBHG module and decoder with two wats to convert spectrogram to audio: Griffin-Lim or our speaker-conditioned vocal model

Conclusions

The paper explores on how entirely-neural speech synthesis pipelines may be extended to multispeaker TTS via low-dimensional trainable speaker embeddings. It presents DeepVoice 2, an improved single-speaker model. Next, it demonstrates the applicability of the technique by training both multi-speaker Deep Voice 2 and multi-speaker Tacotron models, and evaluate their quality through MOS. In conclusion, the speaker embedding technique can be used to create high quality TTS systems and conclusively show that neural speech synthesis models can learn effectively from small amounts of data spread among hundreds of different speakers.

How the paper helped with the project?

The predecessor of the paper - **DeepVoice 1: Real-time neural Text-To-Speech** marked the beginning of the journey of speech synthesis based on neural networks. **DeepVoice 2: Multi-Speaker Neural Text-to-Speech** took it a step further by expanding the horizon to multiple speakers. **DeepVoice 3: Scaling Text-to-Speech with Convolutional Sequence Learning** [2] took it even a step further. The project paper assigned to us is **Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis** [4]. This paper draws inspiration from the results achieved in DeepVoice series of papers, specifically DeepVoice 2 and DeepVoice 3. DeepVoice 2 helper us understand the foundations laid by it in the field of multi-speaker TTS.

References

[1] *Deep Voice 2: Multi-Speaker Neural Text-to-Speech, 2017* Sercan Arik and Gregory Diamos and Andrew Gibiansky and John Miller and Kainan Peng and Wei Ping and Jonathan Raiman and Yanqi Zhou

[2] *Deep Voice 3: Scaling Text-to-Speech with Convolutional Sequence Learning, 2018* Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan O. Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, John Miller

[3] *Deep voice: Real-time neural text-to-speech, 2017* S. O. Arik, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, J. Raiman, S. Sengupta, and M. Shoeybi

[4] *Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis, 2019* Ye Jia, Yu Zhang, Ron J. Weiss, Quan Wang, Jonathan Shen, Fei Ren, Zhifeng Chen, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, Yonghui Wu

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

[6] *Tacotron: Towards end-to-end speech synthesis, 2017* Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio

[7] *Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks, 2006* A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber

[8] *Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015* S. Ioffe and C. Szegedy

[9] *Neural architectures for named entity recognition, 2016* G. Lample, M. Ballesteros, K. Kawakami, S. Subramanian, and C. Dyer

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

[11] *Quasi-recurrent neural networks, 2017* J. Bradbury, S. Merity, C. Xiong, and R. Socher

[12] *Crowdmos: An approach for crowdsourcing mean opinion score studies, 2011* F. Ribeiro, D. Florêncio, C. Zhang, and M. Seltzer