Voice Style Cloning for Chinese Speech

Ziqi Chen **Haiyun Wang** **Luoyi Yang**
Department of Civil and Environmental Engineering
Stanford University
{chenzq, haw091, luoyiy95}@stanford.edu

Abstract

In this project, we focus on training and evaluating an end-to-end text-to-speech synthesis system named Tacotron. Different from traditional text-to-speech model which requires engineers to train three modules including a frontend text analyzer, an acoustic module, and an audio synthesizer, Tacotron synthesizes speech from characters directly and allows engineers to skip a lot of feature engineering work (Wang et al., 2017). Based on our current training data size, the best audio clip generated by Tacotron in Chinese scores 0.79 out of 1 when compared with the original voice.

1 Introduction

Miming or cloning any person's vocal style has been interesting. We found it really popular as we have seen many YouTube videos using celebrity voices to make new audio clips. However, the quality of newly generated audio clips are usually not so good. We also notice the majority of these are in English. Therefore, we look into modern text-to-speech systems that can be trained to read Chinese in a specific vocal style. For instance, if we use 10 hours of a Chinese actress's voice to train the model, later on, we hope to hear the Chinese actress reading any input Chinese texts.

2 Related work

From investigating the current text-to-speech synthesizing methods, especially for Non-English languages, we discover multiple related early work done by researchers around the world on Chinese, Japanese, and German. But conventional text to speech synthesizers use phonemes, diphones, demi-syllables or syllables as speech units to synthesize (Hakoda et al, 1990), and the selection of the most efficient speech unit is difficult without deep learning approaches. There has been a proposal of a simple four layer RNN prosodic synthesizer for mandarin Chinese text-to-speech (Chen et al, 1998), and the focus is on learning human phonological rules in the speech, and the evaluation method is only human evaluation for naturalness and completement instead of objective evaluation matrix. Recently, the WaveNet (van den Oord et al., 2016) came out as a relatively powerful audio synthesizing tool to mimic human voice, but it requires pre-processing of the audio on linguistic conditions, and is therefore not an end-to-end approach to effectively perform voice style cloning. We utilized Tacotron (Wang et al., 2017), which performs end-to-end speech text to speech synthesize instead of focusing on improving prosodic features compared to the previous approaches. The DeepVoice Neural Network (Arik et al., 2017) published nearly at the same time as Tacotron can further perform real time neural text to speech synthesis, and it changed all the steps in typical text-to-speech process to deep neural networks.

,

3 Dataset and Features

We use an open-source online data set from data-baker.com [1]. The file Chinese Standard Mandarin Speech Copus10000 Sentences) includes 100000 Chinese sentences (approximately 10 hours) read by a single Chinese female broadcaster. We originally planned to split our dataset based on the 20% dev set 20% test set and 60% training set rule. However, due to a limitation of computational power, we first use a training set with 1000 sentences and then increase the dataset to 2000 sentences. We also attempt to incorporate another dataset named AI Shell. Different from the first dataset in which all the sentences are read by a single female broadcaster, AIshell [2] contains approximately 10000 Chinese sentences read by 400 people from various age groups with different genders and accents. Since we are interested in voice cloning, we realize it makes more sense if the training data are homogeneous in voice so that the model can better clone that voice. Therefore, we proceed with data set from data-baker.com.

4 Tacotron

The architecture of Tacotron is displayed in Figure 1. Tacotron can be divided into three main parts: encoder, decoder and post-processing net. Encoder extracts sequential representation of the text by applying a series of non-linear transformations to the embeddings of the characters. Later on, the encoder representation is passed to a tanh attention decoder where the attention module is applied to each step of decoding. The decoder uses a fully connected output layer to predict the output targets and we use an 80-band mel-scale spectrogram (Wang et al., 2017). In the end, post-processing synthesizes a spectrogram from the targets generated by the decoder using a Griffin-Lim Synthesizer (Wang et al., 2017).

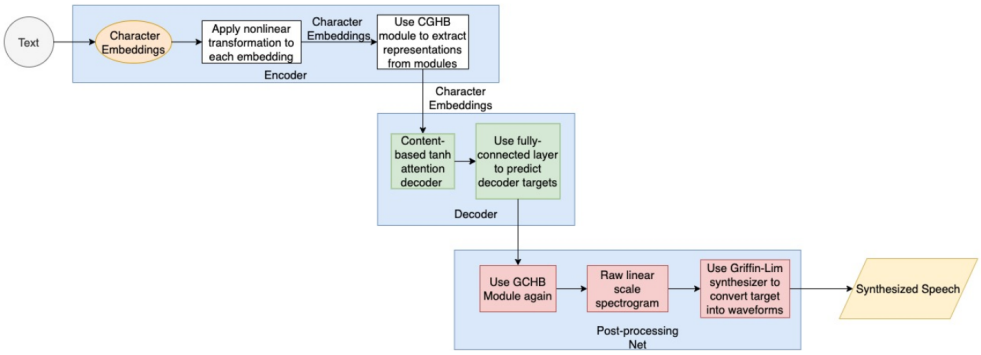

Figure 1: Framework of the Tracotron

Loss function L used for training Tacotron is a combination of two simple l_1 loss functions:

$$L = L_{mel-scale\ spectrogram} + L_{linear-scale\ spectrogram}$$

where $L_{mel-scale\ spectrogram}$ represent the loss from seq2seq decoder; $L_{linear-scale\ spectrogram}$ represents the loss from post-processing net (Wang et al., 2017).

5 Experiments

We first pre-process the label for each audio to form the structure of audio number plus Chinese phonetic alphabet. Then we generate <label, audio> pairs as our training data. We start training Tacotron with the first dataset Chinese Standard Mandarin Speech Copus (10000 Sentences). Due to constraint on computational power, we chop the dataset and train the model with different input data sizes for 5000 steps. After obtaining recognizable Chinese audio with 1000 training examples, we move froward to tuning hyperparameters and changing optimization methods. We set the original Tacotron as baseline, which adopts an initial learning rate (LR) of 0.002 with decay, a batch size (BS)

[1] `https://www.data-baker.com/open_source.html`

[2] `http://openslr.org/33`

of 32, and Adam Optimizer. To improve Tacotron's performance, we test combinations of different initial learning rates, batch sizes and optimization methods. We double the learning rate to 0.004 with the expectation of faster loss convergence during training. We also increase the batch size to 64 attempting to accelerate the overall training process. Besides, we implement RMSprop Optimization instead of Adam Optimizer to foster faster gradient descent.

6 Results/Discussion

Loss Evaluation

We firstly evaluate trained models in terms of loss at training step of 500. As shown in Table 1, when we increase batch size only, we fail to see much difference in the loss from the baseline at the training step of 500. Although increased batch size can help decrease the number of parameter updates (Smith et al., 2018), which saves computation power, it slows the learning for Tacotron. In other trials, loss decreases when we double the learning rate. The loss further declines when both batch size and learning rate are doubled. This matched the expectation of faster learning and convergence. We also observe that Adam optimizer significantly outperforms RMSprop optimizer. Indeed, the loss of RMSprop starts to oscillate around 0.14 from the training step of 7000. We think batch size and learning rate need tuning for the RMSprop Optimizer to make it converge. Considering time constraint and better performance of Adam Optimizer, we cease to the further investigate RMSprop Optimization method as part of our future work. Given the comparison of loss among all models, we may conclude that doubling batch size and initial learning rate simultaneously from the baseline can boost the performance of Tacotron.

Table 1. Loss Comparison

Optimizer		LR = 0.002	LR = 0.004
Adam	BS = 32	0.19281 (baseline)	0.18998
	BS = 64	0.19286	0.1656
RMSprop	BS = 32	0.23615	Not Trained

* Loss at training step 500

Synthesizes Audio Performance Evaluation

To more thoroughly evaluate the quality of trained models, we use both machine scoring and human scoring to compare synthesized audios from all models trained to the step of 10000, where almost all our models can produce recognizable audios.

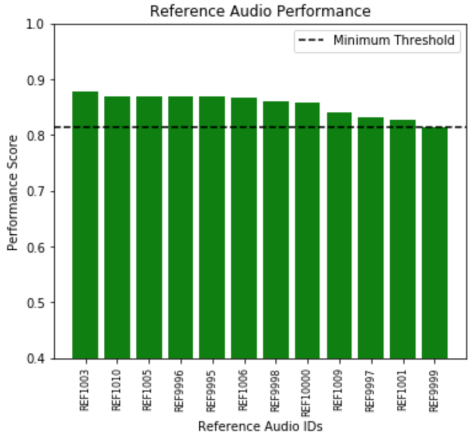

Figure 2: Setting Target Score for Synthesized Audio

For objective machine scoring, we adopt Resemblyer, a deep learning python audio analyzer for fake speech detection, to score our audio clips. Given a speech, this model generates a vector (embedding) of 256 values that summarizes the characteristics of the voice with an emphasis on speech naturalness

and style. The model then scores the synthesized speech against the reference (real) audio based on the difference in their embeddings. We firstly modify Resemblyer to evaluate 12 randomly selected real audio clips from our test dataset and determine 0.81, the minimum score among the 12 audios, as our training target so that the score of synthesized audio can reach this target.

We then evaluate six examples synthesized by each of five training methods listed in Table 1 and summarize the comparison of scores in Figure 3. Among the five methods, we find combination 5 (LR=0.004, BS=64, Adam Optimizer) being the best, combination 3 (LR=0.002, BS=32, RMSprop Optimizer) being the worst and the rest fall in the middle. Although synthesized audio from combination 5 does not reach the target threshold, we have made substantial improvement compare to the baseline model.

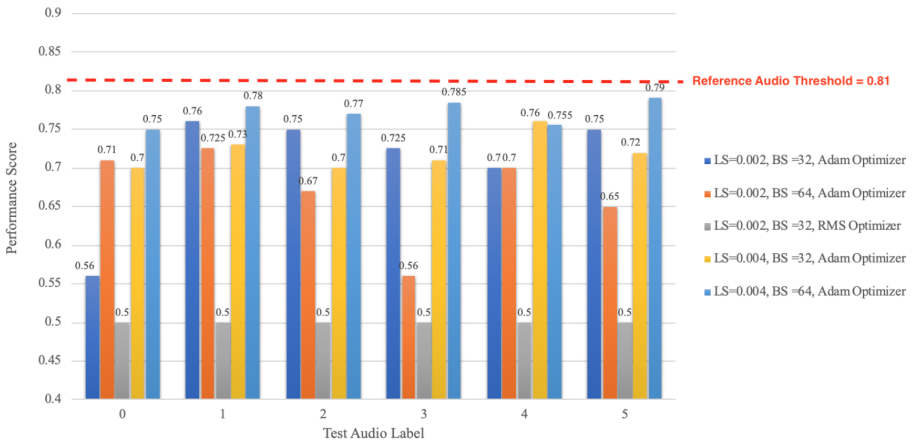

Figure 3: Machine Evaluation Chart

For human scoring, we send out evaluation surveys to 10 people and ask them to score the audios from 0 to 1 based on speech naturalness, style, and correctness. Then, we take the average of the 10 responses for each audio clip as the subjective evaluation score. As shown in Figure 4, human score ranking generally matches the machine evaluation results. We confirm that combination 5 is the best training approach among all the combinations that we have tried so far.

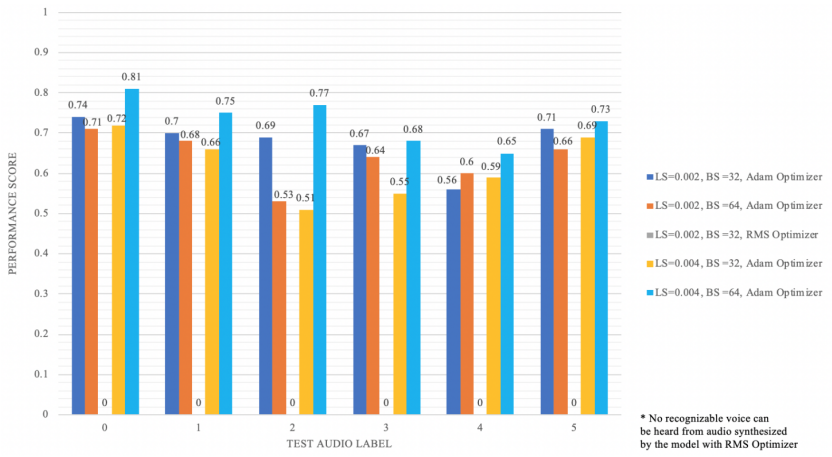

Figure 4: Human Evaluation Chart

Thereafter, We indeed resume our training on combination 5 from step 10000, but find no substantial improvement from training step 10000 to 15000. The performance at step 15000 doesn't improve as expected due to the relatively small training dataset, as the model only learns a limited amount of character pronunciation in the 1000 samples. Therefore, we think further training will not significantly improve model performance with the 1000 audios in the training set. Considering that, we incorporate

1000 more audios and started to retrain combination 5 for Tacotron, which increases the total training set size to 2000. So far, we have trained this model for 10000 steps, but the model is not sufficiently trained to yield a decent results as shown in Figure 5. Further training would be needed to observe a better training performance.

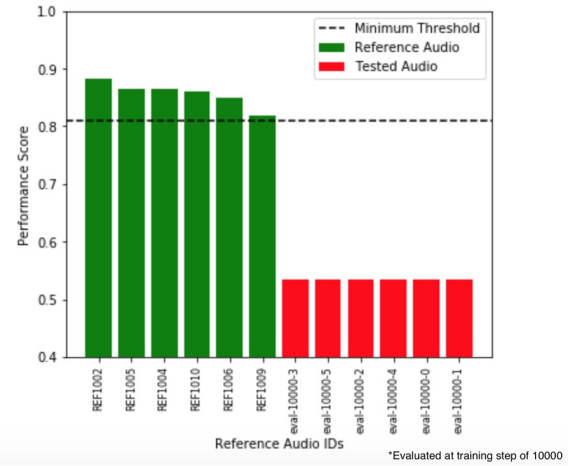

Figure 5: Machine Evaluation For Model Trained with 2000 Audios

7 Conclusion

In this project, we have researched on improving Tacotron, a text-to-speech model, to achieve voice cloning. We have generated five sets of synthesized audios with different combinations of hyperparameters and optimizers. We have set up our training target and evaluated the synthesized audios both subjectively and objectively. After finding the idealist setting (LR=0.004, BS=64, Adam Optimizer), we have tried to improve the model performance to reach our target threshold by increasing training steps as well as enlarging the training dataset.

8 Future Work

We will continue to train Tocatron with 2000 audios to better improve the performance of the model, so that hopefully it will reach our training target of 0.81. After that, we plan to further extend the functionality of Tacotron. So far, we have found a real-time voice cloning model that can perform text-to-speech tasks using the voice recorded by any person in real time. The real-time voice cloning model takes the an outside voice embedding of a speaker and generates a spectrogram based on the voice embedding and input text (Ye et al., 2018). Different from Tacotron which requires a large training dataset of audios from a particular speaker to perform cloning, the real-time voice cloning model only takes 10 seconds of recorded audio as an input variable and performs decent cloning. We tested that the original voice cloning model works well reading English text but not Chinese text. We find the incorporation of Tacotron in the Synthesizer of The real-time voice cloning model, so we plan to replace the original synthesizer with our Tacotron model to improve the Chinese language compatibility of the real-time cloning model.

Contributions

- Ziqi Chen: Idea Brainstorming, Proposal, Setup Tacotron, Milestone, Tuning Hyperparameters, Setup Resemblyer, Machine Evaluation, Final Report
- Haiyun Wang: Idea Brainstorming, Proposal, Milestone, Setup Resemblyer, Loss Evaluation, Machine Evaluation, Human Evaluation, Final Report
- Luoyi Yang: Idea Brainstorming, Proposal, Setup Tacotron, Milestone, Tuning Hyperparameters, Loss Evaluation, Human Evaluation, Final Report

Code

We modified Tacotron and Resemblyer slightly for our training and evaluation. Code can be found at `https://stanford.box.com/s/cl8stmyylah8fegujuofd861s01u9m5y`

Source Code of Tacotron:`https://github.com/boltomli/tacotron`

Source Code of Resemblyer:`https://github.com/resemble-ai/Resemblyzer`

References

Aaron 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. *arXiv preprint arXiv:1609.03499,* 2016.

Chen, Sin-Horng, et al. "An RNN-Based Prosodic Information Synthesizer for Mandarin Text-to-Speech." *IEEE Transactions on Speech and Audio Processing, vol. 6, no. 3, 1998, pp. 226–239.,* doi:10.1109/89.668817.

Hakoda, K., et al. "Japanese Text-to-Speech Synthesizer Based on Residual Excited Speech Synthesis." *ICSLP90 International Conference on Spoken Language Processing,* 1990.

Sercan Arik, Mike Chrzanowski, Adam Coates, Gregory Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Jonathan Raiman, Shubho Sengupta, and Mohammad Shoeybi. Deep voice: Real-time neural text-to-speech. *rXiv preprint arXiv:1702.07825,* November 2017.

Smith, Samuel L., et al. "Don't Decay the Learning Rate, Increase the Batch Size." *ArXiv.org*, 24 Feb. 2018, arxiv.org/abs/1711.00489.

Ye Jia, Yu Zhang, Ron J. Weiss, Quan Wang, Jonathan Shen, Fei Ren, Zhifeng Chen, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, Yonghui Wu. Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis. *Advances in Neural Information Processing Systems 31 (2018), 4485-4495,* 2018.

Wang Yuxuan , RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous. Tacotron: Towards end-to-end speech synthesis. *In Proc. Interspeech, pages 4006–4010,* August 2017.