CS230

Using Discriminators to Transfer Vocal Style on end-to-end TTS Systems

Justin A. Rose
Department of Computer Science
Stanford University
justrose@stanford.edu
*

Abstract

This paper discusses an attempt to transfer vocal speech style using discriminative techniques. The model uses an efficient implementation of a TTS system that has been pretrained on an existing speaker's vocal distribution and uses a convolutional discriminator so that the TTS system's generated mel spectrograms will fit the distribution of a target speaker. Although we were able to train the original TTS model to generate realistic sounding examples from the source speaker (MOS of 3.1), when trying to transfer to a target, the model failed to generalize and produced unintelligible speech with a MOS of 1.2. We believe this to have been an issue involving the fickle training of GANs and think that further investigation into hyperparameter tuning and recent advances in improving the stability of adversarial training could significantly improve performance.

1 Introduction

Text-to-speech systems (TTS) have shown incredible improvements due to recent advances in deep learning techniques. However, most models require vast amounts paired, labeled audio data with transcriptions to train. Given the abundance of existing recordings that could capture a specific person's speaking style, but do not containing any labeled transcriptions, we believe it to be a worthwhile task to explore the potential of transferring vocal style from one speaker to another. By not needing transcriptions, this could drastically reduce the efforts needed to create a new TTS system on a different speaker, a huge step forward for modern human-computer interaction. More specifically, in the case of most TTS systems, character embeddings are input to some deep neural network which outputs a mel spectrogram. Additional modern models can then convert these mel spectrograms to high quality audio. By conditioning these produced mel spectrograms to have a similar context to a source speaker's output and at the same time conditioning them to be similar to spectrograms from a target distribution by using a discriminator, a model could theoretically produce TTS in a target distribution, while only having paired data from a source distribution. As far as we are aware, an end-to-end style transfer for TTS systems has not been attempted.

2 Related work

For many years, concatenative synthesis with unit selection [1], the familiar process of combining short audio fragments together to create audio clips, was leading edge in speech synthesis. Later

*Github repo can be found at https://github.com/justinarose/dc_tts

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

attempts using statistical parametric synthesis [2] helped to solve some of concatenative synthesis's shortcomings, especially artifacts along stitch's boundaries. However, the past few years have seen incredible advances in machine learning techniques to generate audio. WaveNet [3], a project by DeepMind, revolutionized these techniques by directly modeling the raw wave outputs of audio clips, instead of using vocoders like previous techniques. Tacotron 2 [4], a recent project by Google, continued on this attempt by creating a fully end-to-end TTS model, which would produce audio samples only from character inputs to the model, not needing any hand engineer feature like phoneme duration or predicted log fundamental frequency that WaveNet used. Although Tacotron 2 is the state of the art, some attempts [5] have tried to make TTS models more efficient, by replacing recurrent connections with convolutions and attention mechanism, which takes advantage of parallelization techniques of modern computers.

In addition to TTS systems, Generative Adversarial Networks (GANs) [6] have shown immense success in modeling data distributions. By training two models in parallel, one to produces new examples that fit an existing data distribution (the generator) and one to discern between real and generated examples (the discriminator), deep systems can be used to create novel yet realistic samples from a data distribution. Such techniques have been used in image synthesis quite successfully. For example, Stack GAN++ [7] recently demonstrated remarkable results in generating high quality images from text descriptions, the seemingly analogous task to speech synthesis. Although some recent work done by Baidu on vocal cloning [8] has shown some success in vocal style transfer, such techniques still used small amounts of labeled data.

3 Dataset and Features

For our source data distribution, we used the LJ Speech [9] dataset, which consists of 13,100 short audio clips from a female speaker and their corresponding transcriptions. For the target vocal style without text data, we chose 6,229 clips of British male speakers from the VCTK dataset [10][2] (even though this dataset does have transcriptions, we did not use them). Since our evaluation metric of the mean opinion score (MOS) is subjective, for development and testing we input twenty short sentences from the standardized set of Harvard sentences [3] to the model, and listened to their results to determine how well the model was performing.

For preprocessing the audio clips, which were sampled at 22,050 Hz, we clipped the silence from the ends of the audio clips by thresholding to 15 decibel and computed the short-time Fourier transform for each clip, which is a way of converting a raw audio sample into matrix of frequencies with respect to time. We used a window size of 0.05 seconds (1102 samples) and a hop size of 0.0125 seconds (276 samples). From the Fourier transform, we could then get for each audio clip the magnitude spectrograms and the mel spectrograms. The mel spectrograms are computed from magnitude spectrograms and are a common technique used to target human speech frequencies. Our mel spectrograms had 80 mel bands. Finally, we normalized the decibel level of the spectrograms so that there wouldn't be variability in the volume for the model during training.

For the text data, we used a simple vocabulary that consisted of the letters of the alphabet, a padding token, an end of speech token, and simple punctuation (space, single quote, period, and question mark). Text was converted to lower case, stripped of accents and extra spaces, and mapped from our vocabulary to a learned character embeddings of dimension 128.

[2]For VCTK, we used speakers 226, 227, 232, 243, 254, 256, 258, 259, 270, 273, 237, 278, 279, 286, and 287

[3]`https://en.wikipedia.org/wiki/Harvard_sentences`

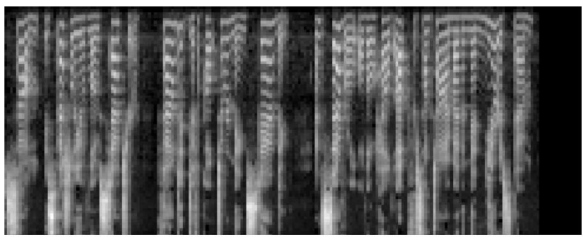


Figure 1: A sample mel spectrogram

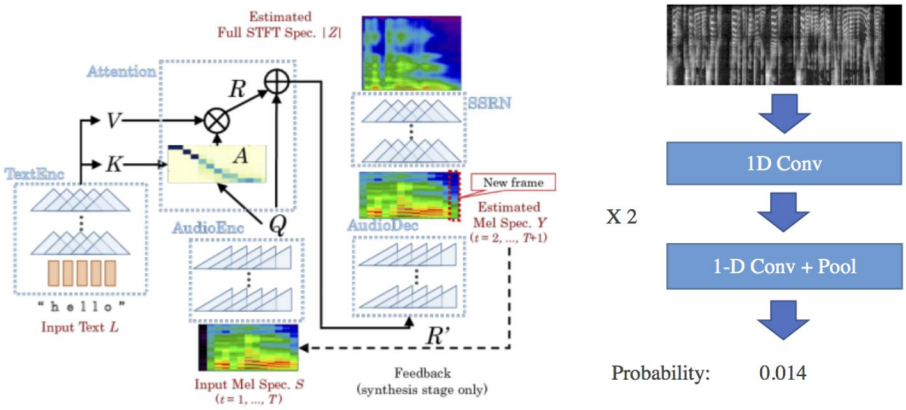


Figure 2: (Left) Tachibana's model (Right) Our discriminator model

4 Methods

4.1 Our Model

For our baseline TTS model, we used one implemented by Tachibana et al. [5], which uses only convolutions and guided attention. Because it doesn't use any recurrent connections, the model can run many more operations in parallel and is much more efficient than other deep models. Although Tacotron 2 is the state of the art, given our limited hardware and compute time, we thought that using a simpler model would allow for more experimentation for our project. We will give a brief overview of the Tachibana model and then discuss our additions to the network and its objective functions.

The Tachibana model consists of two trainable networks: Text2Mel, which takes in character embeddings and outputs mel spectrograms, and Spectrogram Super-resolution Network (SSRN), which turns mel spectrograms into full magnitude spectrograms. These magnitude spectrograms can then be turned into audio clips by the Griffin-Lim algorithm. The SSRN network is essentially just a series of one-dimensional convolutions and since our changes didn't affect the model at all, we will leave a detailed description of that part to the original paper. The Text2Mel network had four submodules. A text encoding module, an audio encoding module, an attention module, and an audio decoding module. These modules follow a similar paradigm of queries, keys, and values matrices, which can be found in other fully attention based papers [11]. The text encoding module takes in as input the a sentence of length N and outputs keys and values matrices K and V of size d by N. The audio encoding module takes in as input the target audio spectrogram and outputs a queries matrix Q of size d by T, where T is the length of the audio clip. An attention matrix $A = \text{softmax}(K^T Q/\sqrt{d})$ is generated, where A_{nt} refers to how much the model should attend to the n-th character when generating the t-th time frame of the spectrogram. The resultant matrix is calculated by $R = VA$. The model concatenates this with the encoded audio Q to create a new $2d$ by T matrix R' and feeds this through the audio decoder module to create a mel spectrogram. Each of these submodules consists of convolutions, rectified linear unit activation functions, and highway connections, and their specific details can be also left to the original paper.

Viewing the Text2Mel model as a generator, our contribution to the network was a convolutional discriminator, which took the outputted mel spectrograms of Text2Mel as input and outputted a zero or one depending on whether it thought the spectrogram was from the target distribution (British male speaker) or generated distribution. Since Text2Mel could output variable length spectrograms, we only considered the first 80 timesteps of the generated audio. We clipped/zero-padded the input spectrograms accordingly. The spectrograms were run through a series of two one-dimensional convolution layers followed by max one-dimensional pooling twice. This output was flattened and run through a dense layer and sigmoid, which output a probability from zero to one on whether the spectrogram was from the target distribution.

4.2 Objective Functions

Since we want the Text2Mel model to produce spectrograms that are both contextually and stylistically similar, we defined two loss functions $L_{context}$ and L_{style}. The contextual loss refers to how similar the generated spectrogram from text is to its corresponding LJSpeech spectrogram. The stylistic loss refers to how well the generated spectrogram fooled the discriminator into believing it came from the target distribution. For the contextual loss, we simply used the loss used in Tachibana's original paper, which is an attention loss, L1 loss, and binary divergence loss, which is essentially pixel-wise log loss. The attention loss is specific to Tachibana's model and is called guided attention. It is used so that the attention matrix is roughly diagonal, because the spectrogram timestamps and character encodings should be roughly attended along the diagonal. For the style loss, we simply use the typical GAN generator loss, which tries to maximize the generated spectrograms likelihood of fooling the discriminator. The total loss for the Text2Mel model is defined as the weighted sum of the contextual and style losses, weighted by a factor of β. The discriminator is the typical discriminator loss, which tries to classify real spectrograms as coming from the target distribution and generated spectrograms as coming from the fake distribution.

$$L_{con}(Y, S) = L_{att}(A) + \frac{1}{FT}\sum_{f,t}|Y_{ft} - S_{ft}| - (S_{ft}\log Y_{ft} + (1 - S_{ft})\log(1 - Y_{ft})) \tag{1}$$

$$L_{att}(A) = \frac{1}{nt}\sum_{n,t} A_{nt}W_{nt}, \text{where } W_{nt} = 1 - \exp\left(-(n/N - t/T)^2/2g^2\right) \tag{2}$$

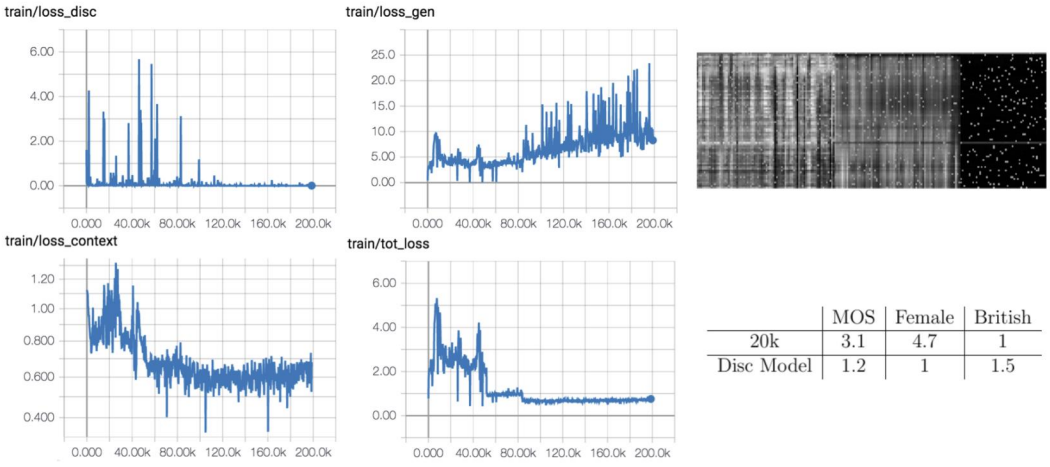
$$L_d(Y_{tar}, Y_{gen}) = -\frac{1}{m}\sum_i(\log D(Y_{tar}^{(i)}) + \log(1 - D(Y_{gen}^{(i)}))) \tag{3}$$

$$L_{tot}(Y_{gen}) = \beta L_{con}(Y_{gen}, S) + (1 - \beta)L_g(Y_{gen}) \tag{4}$$

$$L_g(Y_{gen}) = -\frac{1}{m}\sum_i \log D(Y_{gen}^{(i)}) \tag{5}$$

5 Experiments and Results

Our code started from a Tensorflow [12] implementation of Tachibana's model [13]. We began by first training the SSRN model on mel and magnitude spectrograms from both distributions of speakers. We used an Adam optimizer with Noam learning rate decay, which starts up to a learning rate of 10^{-3} and decreases after 100,000 batch iterations to $2 * 10^{-4}$. We used a batch size of 32 and trained SSRN for 20,000 iterations. Once SSRN was trained, we pretrained Text2Mel only on its



	MOS	Female	British
20k	3.1	4.7	1
Disc Model	1.2	1	1.5

Figure 3: (Left) Loss graphs during training (Top-Right) Sample produced spectrogram (Bottom-Right) MOS table with British and female probabilities for pretrained 20k model and discriminator trained model

contextual loss, to try to have it start from a better initialized state for when we began to train with our discriminator. We pretrained the model for $20,000$ iterations and determined that the audio quality was good enough. Finally, we pretrained the discrimator to discriminator between true spectrograms from the LJ Speech distribution and true spectrograms from the VCTK British male distribution. These hyperparameters for training, such as the learning rate and batch size, were determined based on what the original paper had used.

Once both models were pretrained, we put both models together and trained them in series. For the hyperparameter β which weighs the contextual and generator loss, we began with a value of 0.5. However, after about 30,000 more iterations we realized that this was weighing the generator loss to heavily and causing the spectrograms had adverse effects on the spectrogram quality. We tried again from the preinitialized weights to train the network, this time with a β value of 0.9. After waiting around 30,000 iterations, these generated sample were slightly better than the original ones, but still were fairly indistinguishable. We tried changed β to 0.99 and let it train until it reached 200,000 iterations.

Our results for this training can be seen above. We report the loss values, a sample produced spectrogram, as well as our metric, mean opinion score (MOS), which averages the rating of the audio samples from a dozen in-house raters (we aren't reporting confidence interval due to their obvious statistical insignificance). If we wished to get a more reliable measure of MOS from more people, we would have used a service such as Amazon Mechanical Turk, but since we lacked the time and financial resources to get that running, our small survey sufficed. We also had people rate how feminine an audio clip sounded and how much it sounded like a British male. As one can see, our original pretrained model produce relatively high samples of female audio quality. However, once we trained on the generator, the audio quality diverged and turned into unintelligible ticks and muttering. Although they rated the second set as slightly more male, it wasn't high enough to claim that there were significant results.

Upon examining the loss functions, it also doesn't seem apparent what may have gone wrong. When using an initial value of $\beta = 0.9$, the generator and discriminator seem to be alternating between which was doing better or worse and it wasn't until switching β to 0.99 that the discriminator began to overpower the generator. Further, even though the total loss was decreasing, the audio quality failed to rival the original quality of the pretrained network. Looking at the generated spectrogram, one can visually see (having looked at other spectrograms) that it is synthetic. The long horizontal and vertical lines throughout are atypical for normal spectrograms and signal that it is the Text2Mel model's errors that are causing the network to fail.

It is known that training GANs can be an extraordinarily difficult problem [14]. Their combined loss functions are highly nonconvex and can easily get stuck in a local minimum. We think this is the best explanation as to what must have happened, because the audio samples quality remained relatively similar even as we went from 30,000 iterations to 200,000 iterations. We suspect that more efforts into stabilizing the GAN training could lead to successful results in the future. Furthermore, we believe that the original context and style loss probably shouldn't be conditioned on the same spectrogram, due to their conflicting goals, and instead believe an attempt similar to CycleGAN [15] might work instead. A CycleGAN approach would have two functions F and F^{-1} which would take in spectrograms from one distribution (like the male speaker) and change them to another distribution (like the female speaker). Given that this network has to have cycle consistence, where $F^{-1}(F(M)) \approx M$ for some spectrogram M, we believe that this approach might be more stable to train and would also not involve propogating the gradients through the Text2Mel model. If we had more time, this would most likely be our next approach.

6 Conclusion

Overall, we still believe discriminative style transfer to be an interesting area of research for voice synthesis. It's potential impact on text to speech and human computer interaction are enormous and could save time in labeling and testing various TTS models out in the real world. Although we were able to successfully train a TTS model on LJSpeech, when we tried to change this to transfer to VCTK data, its results were non-significant, most likely due to getting stuck at a local minimum training the GAN. In the future, we think applying stable GAN training techniques would significantly improve the results, and look forward to other people's attempts.

7 Contributions

This paper was done by a group of one person.

References

[1] A. J. Hunt and A. W. Black, “Unit selection in a concatenative speech synthesis system using a large speech database,” in Proc. ICASSP, 1996, pp. 373–376.

[2] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, “Speech parameter generation algorithms for HMMbased speech synthesis,” in Proc. ICASSP, 2000, pp. 1315– 1318.

[3] Aaron Van Den Oord, et al. "Wavenet: A generative model for raw audio." arXiv preprint arXiv:1609.03499, 2016.

[4] Jonathan Shen, et al. "Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions." arXiv preprint arXiv:1712.05884, 2017.

[5] Hideyuki Tachibana, et al. "Efficiently Trainable Text-to-Speech System Based on Deep Convolutional Networks with Guided Attention." arXiv preprint arXiv:1710.08969, 2017.

[6] Goodfellow, Ian, et al. "Generative adversarial nets." Advances in neural information processing systems. 2014.

[7] Han Zhang, et al. "Stack GAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks." arXiv preprint arXiv:1710.10916, 2017.

[8] Jitong Chen, et al. "Neural Voice Cloning with a Few Samples." arXiv preprint arXiv:1802.06006, 2018.

[9] Keith Ito, et al. "The LJ Speech Dataset." `https://keithito.com/LJ-Speech-Dataset/`, 2017.

[10] Christophe Veaux, et al. "CSTR VCTK Corpus," 2010.

[11] Ashish Vaswani, et al. "Attention is all you need." Advances in Neural Information Processing Systems. 2017.

[12] Martín Abadi, et al. "TensorFlow: A System for Large-Scale Machine Learning." OSDI. Vol. 16. 2016.

[13] Kyubyong Park. "A Tensorflow Implementation of DC-TTS: yet another text-to-speech model." `https://github.com/Kyubyong/dc_tts/` 2018.

[14] Martin Arjovsky and Léon Bottou. "Towards principled methods for training generative adversarial networks." arXiv preprint arXiv:1701.04862 (2017).

[15] Jun-Yan Zhu, et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks." arXiv preprint arXiv:1703.10593 (2017).