A WaveNet Based Framework for Generating Singing Voice (TAU Advanced Methods in ML course, 2018)

Andrey Leshchenko * **Nir Raviv** * **Roi Tabach** *

Abstract

In 2016, Van Den Oord et al. presented WaveNet- a Deep neural network with the ability to generate raw audio waveforms. Since than, there have been massive work in the area of Text to Speech and Speech to Text using WaveNet and its successors. There has been some work regarding generating human singing voices, and we wanted to expand on that, as well as improve on their results. We created a TTS based system that generates singing voice from a note track, with results better than the baseline, although not better than previous work.

1. Introduction

The goal of our project is to generate singing voices that sound as close to human as possible.

Compared to text, the problem of creating singing voices is more complex, as the system has to generate matching pitch, timbre and melody. Another problem is the lack of large datasets of professional recordings with timed lyrics.

Our overall approach to solving the problem is to use existing state-of-the-art systems for TTS, and training them on "text" which is made from a list of musical notes. We have a novel note detection method, built on top of the MELODIA pitch detection algorithm (Salamon & Gomez, 2012). We then pass the musical notes stream, encoded as text, as the text that goes into the Tacotron 2 network.

The novelty in our work is twofold: First by the way we extract the musical notes, and second by the novel usage of a working TTS system as a Notes-to-Singing system.

. Correspondence to: Nir Raviv <nirraviv@mail.tau.ac.il>, Roi Tabach <roi.tabach@gmail.com>, Andrey Leshchenko <andrey.leshenko@gmail.com>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

2. Related Work

During the recent years many neural networks architecture based on fully-visible probabilistic autoregressive generative models are applied in many fields, such as natural images (PixelCNN) (van den Oord et al., 2016), raw audio waveform (WaveNet) (Van Den Oord et al., 2016) and video (Video Pixel Networks) (Kalchbrenner et al., 2016). These methods predict the distribution for each sample conditioned on all previous ones, while remaining efficiently trained.

Since WaveNet was published in 2016, there have been several major attempts at the problem of synthesizing speech with neural networks, including Deep Voice 1 (Arik et al., 2017), 2 (Arık et al., 2017), and 3 (Ping et al., 2018), Tacotron (Wang et al., 2017), Char2Wav (Sotelo et al., 2017), and others. Deep Voice 1 and 2 are built as TTS pipelines, separating grapheme-to-phoneme conversion, duration and frequency prediction, and waveform synthesis. Deep Voice 3, Tacotron and Char2Wav propose sequence-to-sequence models for neural TTS. Tacotron is a neural text-to-spectrogram conversion model, which uses Griffin-Lim (Griffin & Lim, 1984) for spectrogram-to-waveform synthesis. Char2Wav predicts the parameters of the WORLD vocoder (Morise et al., 2016) and uses a SampleRNN (Mehri et al., 2016) conditioned upon WORLD parameters for waveform generation. These works achieve state of the art performance, with human listeners rating as more natural sounding than the previous state of the art methods of concatenative methods.

Different singing generators are based on statistical parametric methods centered around Hidden Markov Models (HMM) which allow joint modeling of timbre and musical expression but perform less well than the previous methods by causing "buzzy" sound (Oura et al., 2012). This work was extended to feed-forward DNNs (Nishimura et al., 2016).

Merlijin *et al.* (Blaauw & Bonada, 2017) presented a synthesizer which can generate synthetic singing voice given musical score and phonetic lyrics. They are using different models to learn phonetic timing, pitch and timbre combining autoregressive generative model with mixture density output, instead of softmax, which allows skewness or truncated

distribution and multiple modes.

3. Our Approach and Experiments

Our overall approach to solving the problem is to use existing state-of-the-art systems for TTS, and to train them on "text" which is made from a list of musical notes. Our pipeline looks as in Figure 1. We begin with several hours of relatively homogeneous recordings of singing voices. We separate them to 5-second long clips. We then split them to slices of 125 milliseconds and pass them through a pitch-detection program (the Essential audio analysis library (Bogdanov et al., 2013)). We then use a novel approach in which we decide which musical note the pitch represents, and encode it as text. The generated text stream, with each 5 second sample, is the (text, speech) pairs that are used in the training of the Tacotron 2 network.

Data Sets

We have based our work on several data sets. One of the closest works used a professional recording that was made by (Blaauw & Bonada, 2017). They did not share their data with us, but Stanford university's DAMP dataset was in fact shared with us.

DAMP 300x30x2 [1]

The DAMP dataset contains approximately 10^5 audio recordings that were captured by Smule's Sing! Android karaoke application and hosted by the CCRMA. The DAMP 300x30x2 is one of several subsets of those recordings. It contains for each of 30 countries, the top 300 songs that users chose to sing inside the specific country. For each song one recording by a male voice and one by a female voice. Each recording usually contains only the singing voice itself, without the background music. Naturally, this means that a large part of the recording is comprised of silence or background noises. A large portion of the recordings has timed lyrics adjoined to them, some of them are timed by the sentence, some by the word, and some by the syllable. Those timings come from the karaoke application, and the users are generally not precisely on time in their singing. The dataset contains tags for some of the songs, with musical genres (e.g. k-pop, jazz, etc.). There is also data for each of the users, including location, and number of followers. When using the dataset we have noticed that our trained models generate various forms of silence and background noise. This was somewhat solved using the following steps: working only on users with more than 10 followers, cutting the 5-second clips to coincide with the beginning of words, and adding fade-in and fade-out for clips that had sharp transitions.

[1] https://ccrma.stanford.edu/damp/

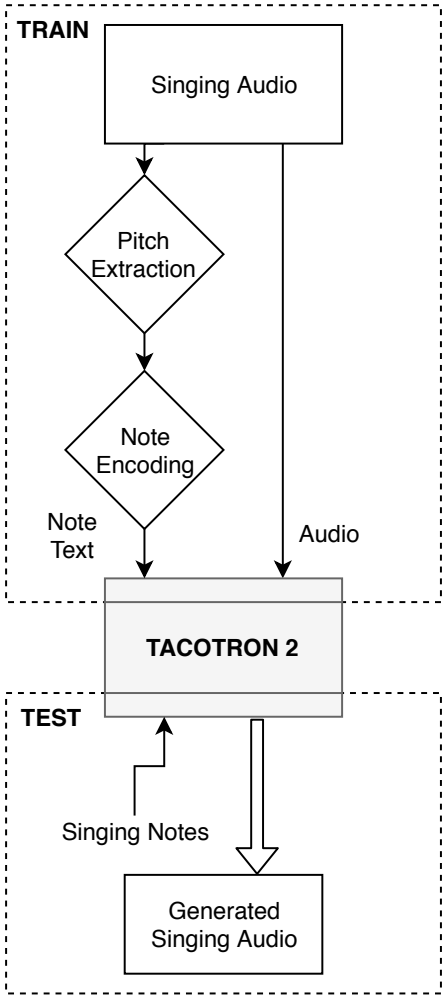

Figure 1. The final Tacotron-2-based architecture of our singing system. At training we feed the network (text, audio) pairs where text is the pitch notes extracted from the audio. For testing we generate audio based on previously unseen singing notes.

A problem with DAMP we couldn't quite solve was the fact that it wasn't homogeneous enough, even after filtering by a specific genre and locale. Many tracks had wrong M/F labels, others contained mostly silence and in some a person was just whispering the words instead of singing. It seems that this dataset could benefit greatly from human cleaning and supervision. These problems, combined with the low quality of the recordings, convinced us to try working on different audio recordings.

Gregorian Chants [2]

In our search for high quality and homogeneous singing data, we have turned to You Tube, where people often post hours-long music collections. There we found a You Tube video

[2] https://www.youtube.com/watch?v=xdroyjKs1Ls

containing six hours of Gregorian Chants (Wawirochi).

Gregorian chant is a form of monophonic, unaccompanied sacred song, which developed mainly in western and central Europe during the 9th and 10th centuries. This dataset proved to be much more homogeneous, and the quality was higher than in DAMP.

LJSPEECH

As a part of the baseline, we have used the public LJ Speech dataset (Ito et al., 2017) that contains several hours of professional voice recordings of text. The recording contains much less noise, and as expected generators trained on them gave much cleaner results.

Extracting the musical notes

Musical notes define the pitch of musical melodies, which is the quality that makes it possible to judge sounds as "higher" and "lower" (Plack et al., 2005). In most cases the pitch will be the fundamental frequency of the signal, which is the frequency of repetition for a periodic waveform. Calculating the fundamental frequency f is complicated, because the harmonics of this frequency ($i \cdot f, \forall i \in \mathbb{N}$) can often be stronger than f.

We have attempted to extract the pitch from the STFT spectrograms of the audio tracks, but had many problems with strong harmonics being detected instead of the weaker fundamental frequency. We then managed to get good results using the MELODIA algorithm (Salamon & Gomez, 2012), which is implemented as part of the Essential audio analysis library (Bogdanov et al., 2013). Figure 2 shows the results

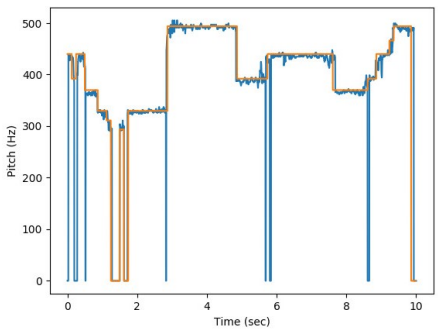

Figure 2. Pitch detection on a recording of a Gregorian Chant.

of pitch extraction on one of the Gregorian Chants tracks. While it is possible to condition networks directly on the estimated pitch vectors, we perform a further simplification of down-sampling the pitch values to 8 values per second, and transforming the continuous frequencies into the discrete keys on a piano. The piano is essentially a logarithmic scale, where the frequency is doubled every 12 keys. Therefore, taking as a reference the A4 note as $f_0 = 440$Hz, we can calculate the frequency of all other notes from their position n relative to the base note:

$$f_n = f_0 \cdot (\sqrt[12]{2})^n$$

After pitch extraction we use the reverse of this formula to get the corresponding note:

$$n = \lfloor \log_{\sqrt[12]{2}} \left(\frac{p}{f_0} \right) + 0.5 \rfloor$$

We then encode each note (from the 4 octaves where most singing occurs) as an uppercase or lowercase English letter. For example, a pitch of 550 Hz will be detected as the note $C_5^{\#}$ (which has a frequency of 554.37 Hz) and will be encoded as "q".

Tacotron 2 TTS

The Tacotron 2 (Shen et al., 2017) is a neural network architecture for synthesizing speech directly from text. We are abusing it, by entering the "text" we got from the musical notes. It's composed of a recurrent sequence to sequence feature prediction network that maps character embeddings to Mel-scale spectrograms, followed by a modified WaveNet model acting as a vocoder to synthesize time-domain waveforms from those spectrograms.

We have trained both Tacotron 2 and all of our baselines on Titan Xp GPUs. Each experiment was ran several hours (at least 12).

Audio outputs can be listened to in here: Singing Voice Generation Samples on Google Drive

Baseline

We have implemented two baseline methods to compare our method with. The first is a simple WaveNet GAN. We have trained it for 12 - 48 hours on various subsets of DAMP and the Gregorian Chant datasets, as well as (for sanity check) LJ speech. Also, We have trained Tacotron, based on the audio samples and their timings from the DAMP dataset.

Conclusion

We presented a singing generator based on Tacotron 2 TTS system that trained over songs' musical notes as "text" and was conditioned over their pitch.

There are two natural ways to judge the quality of the outcome of a system that is designed to generate human sounding voices: comparison to previous state-of-the-art methods, and comparison to real human voices.

Our system generates audio that does not really sound human at all, and comparing it with human voices is hopeless. Also, the existing state-of-the-art systems generate results that are much better than ours. We have reached this conclusion even without the usage of mean opinion score tests, which is a common measuring way in the field of generating human speech.

We have however compared our system to the two baselines described above. We find that the results are slightly better than those generated from the DAMP and Gregorian chants, but sound less human than LJSpeech.

An interesting observation is that the best results (e.g. "damp ha tikva.wav" in the samples folder) that we got in our Notes-to-Singing model were those that were trained on DAMP. But there was a large group of notes tracks that the model did not "sing well", with problems such as singing that faded out to silence in the middle of the song. This could be another testimony to the unbalanced or messy nature of the DAMP dataset, which means this method could achieve better results on a more professional dataset.

It is evident that there is an importance to having a clean, good quality recordings, with correct timings. The better the dataset was in those metrics, the better the results sounded.

Future work

The most obvious way to continue is trying to use a TTS system but adding another text channel, meaning using both correctly timed text and notes extracted in the way we did. This requires having a singing voice dataset which is better than publicly available today.

Another option for the system is to try use tools from the image style transfer world, adapted to the audio domain. There have been several works in this direction, but usually between several musical instruments and not between several tunes of a singing voice.

Another future venue that seems to suggest itself naturally from our representation for the musical notes is using tools from the domain of deep learning over text, to auto generate notes in the same style as those in our data sets, i.e. to compose music that would be in the same style as given audio samples.

Acknowledgments

We gratefully acknowledge Prof. Amir Globerson and the course staff for their support and for letting us use the GPUs we ran on. We also thank Smule Inc. for sharing with us the DAMP dataset, which is hosted at Stanford University's CCRMA. We thank Rayhane Mama and Igor Babuschkin, the developers of Git free versions of Tacotron 2 and WaveNet.

References

Arik, S. O., Chrzanowski, M., Coates, A., Diamos, G., Gibiansky, A., Kang, Y., Li, X., Miller, J., Ng, A., Raiman, J., et al. Deep voice: Real-time neural text-to-speech. *arXiv preprint arXiv:1702.07825*, 2017.

Arık, S. O., Diamos, G., Gibiansky, A., Miller, J., Peng, K., Ping, W., Raiman, J., and Zhou, Y. Deep voice 2: Multi-speaker neural text-to-speech. *arXiv preprint arXiv:1705.08947*, 2017.

Blaauw, M. and Bonada, J. A neural parametric singing synthesizer. *arXiv preprint arXiv:1704.03809*, 2017.

Bogdanov, D., Wack, N., Gómez, E., Gulati, S., Herrera, P., Mayor, O., Roma, G., Salamon, J., Zapata, J. R., and Serra, X. Essentia: an audio analysis library for music information retrieval. In *International Society for Music Information Retrieval Conference (ISMIR'13)*, pp. 493–498, Curitiba, Brazil, 04/11/2013 2013. URL http://hdl.handle.net/10230/32252.

Griffin, D. and Lim, J. Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243, 1984.

Ito, K. et al. The lj speech dataset, 2017.

Kalchbrenner, N., Oord, A. v. d., Simonyan, K., Danihelka, I., Vinyals, O., Graves, A., and Kavukcuoglu, K. Video pixel networks. *arXiv preprint arXiv:1610.00527*, 2016.

Mehri, S., Kumar, K., Gulrajani, I., Kumar, R., Jain, S., Sotelo, J., Courville, A., and Bengio, Y. Samplernn: An unconditional end-to-end neural audio generation model. *arXiv preprint arXiv:1612.07837*, 2016.

Morise, M., Yokomori, F., and Ozawa, K. World: a vocoder-based high-quality speech synthesis system for real-time applications. *IEICE TRANSACTIONS on Information and Systems*, 99(7):1877–1884, 2016.

Nishimura, M., Hashimoto, K., Oura, K., Nankaku, Y., and Tokuda, K. Singing voice synthesis based on deep neural networks. In *INTERSPEECH*, pp. 2478–2482, 2016.

Oura, K., Mase, A., Nankaku, Y., and Tokuda, K. Pitch adaptive training for hmm-based singing voice synthesis. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 5377–5380. IEEE, 2012.

Ping, W., Peng, K., Gibiansky, A., Arik, S., Kannan, A., Narang, S., Raiman, J., and Miller, J. Deep voice 3:

Scaling text-to-speech with convolutional sequence learning. In *Proc. 6th International Conference on Learning Representations*, 2018.

Plack, C., Oxenham, A., Fay, R., and Popper, A. *Pitch: Neural Coding and Perception*. Springer Handbook of Auditory Research. Springer, 2005. ISBN 9780387234724. URL https://books.google.co.il/books?id=n6VdlK3AQykC.

Salamon, J. and Gomez, E. Melody extraction from polyphonic music signals using pitch contour characteristics. *Trans. Audio, Speech and Lang. Proc.*, 20 (6):1759–1770, August 2012. ISSN 1558-7916. doi: 10.1109/TASL.2012.2188515. URL http://dx.doi.org/10.1109/TASL.2012.2188515.

Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Skerry-Ryan, R., et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. *arXiv preprint arXiv:1712.05884*, 2017.

Sotelo, J., Mehri, S., Kumar, K., Santos, J. F., Kastner, K., Courville, A., and Bengio, Y. Char2wav: End-to-end speech synthesis. 2017.

Van Den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*, pp. 4790–4798, 2016.

Wang, Y., Skerry-Ryan, R., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., et al. Tacotron: Towards end-to-end speech synthesis. *arXiv preprint arXiv:1703.10135*, 2017.

Wawirochi. The 99 most essential gregorian chants (complete). URL https://www.youtube.com/watch?v=xdroyjKs1Ls.