Towards an End-to-End Deep Singing Synthesizer

Juncen (Amy) Wang
Advised by Akash Mahajan

Stanford University

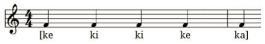
PROBLEM

The goal of this project is to apply deep learning to the application of singing synthesis, which at its most basic involves turning the notation of a song – notes, rhythm, and lyrics (such as in a midi format enhanced with lyrics) into a synthesized waveform. Ideally, this output audio would be indistinguishable from human singing, with the timbre, understandability, and expression of a human vocal performance.

While there have been a large number of positive results in applying machine learning to portions of singing synthesis, there has not yet been a successful end-to-end deep singing synthesizer published. This particular project focuses on the first steps towards such an end-to-end synthesizer: generating sound given pitch and lyrics, neglecting rhythm and expression for now in order to perfect a working model.

DATA

We used data freely available online recorded for use with an existing (non-machine-learning-based) singing synthesizer. This data consists of long strings of monotone sung syllables, made for the Japanese language and recorded by one female singer; furthermore, it comes with time labels for each of the "lyrics". One of these files may be be musically transcribed like this:



Each recording lasts around four seconds, and the entire training set contains about six hours of data. One of the architectures applies a mel-spectrogram transformation and uses that as a feature instead of raw audio:

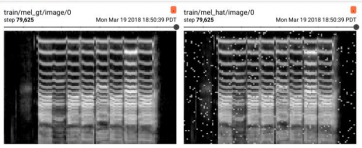


Fig. 1: ground truth and predicted mel-spectrograms (Model 4 at 16000 epochs)

MODELS

We experimented on two different model architectures:

WaveNet

WaveNet is a convolutional neural network that learns a conditioning of audio on previous samples, with optional additional local conditions. It doesn't use any pooling layers, and consists of a series of stacks - sequences of layers with increasing dilations (holes between convolved values in the input sequence):

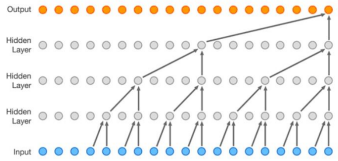


Fig. 2: A WaveNet stack. (Source: DeepMind)

The loss function takes both into account the accuracy of the predicted sample, and its predicted classification based on the input local condition.

DC-TTS

Deep Convolutional Text-to-Speech was designed as a convolutional speedup to RNN text-to-speech systems. It learns encodings of the input features and of the audio mel-spectrogram using convolutional networks. Like a sequence-to-sequence RNN, it also learns an attention module before a decoding function to produce a mel-spectrogram and audio (the SSRN module) given those encodings. The loss function sums a least-absolute-deviation and a logistic loss between the predicted and the ground truth waveforms:

$$L_{text2mel} = \sum |Y - S| - \sum S\log(y) + (1 - S)\log(1 - Y)$$

It also summed the loss from the attention, using a modified linear mapping as ground truth:

$$\text{Expected attention}_{nt} = 1 - e^{-\left(\frac{n}{N} - \frac{t}{T}\right)^2 / 2g^2}$$

RESULTS

The WaveNet models were extremely costly in training time, synthesis time, and memory; this limited the hyperparameter space and the iteration speed. DCTTS fared much better; We also experimented with speeding up training by processing the audio differently, along with passing in the lyric sequence as opposed to the labeled time series of lyrics as input.

Model	Training loss (97% of data)	Test loss (3% of data)
1 - WaveNet (4 stacks of 6, batch size 2, 160 epochs)	75.42	74.67
2 - WaveNet (3 stacks of 5, batch size 6, 420 epochs)	77.61	77.89
3 - DCTTS sequence (fft shift=0.0125, embedding=256, hidden units=128)	0.4462	0.45599
4 - DCTTS series (fft shift=0.025, embedding=400, hidden units=128)	0.47735	0.51338
5 - DCTTS series (fft shift=0.0125, embedding=256, hidden units=320)	0.43098	0.49903
6 - DCTTS sequence (fft shift=0.0125, embedding=256, hidden units=320)	0.44074	0.47415

Table 1: Models and losses

As the WaveNet batch size was forced to be so small, the loss graph resembled that of stochastic gradient descent, which might explain why the test loss was less than the training loss for the first model:

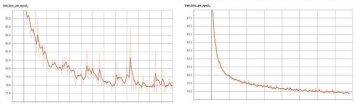


Fig. 3: Test and training loss, Model 1 at 160 epochs

The attention module learned by the DC-TTS models was noteworthy:

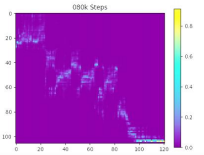


Fig. 4: Attention (output vs. input timesteps), Model 4 at 16000 epochs

DISCUSSION

Training time was an obstacle when working with these architectures - with the first experimental models we allowed the model to train for as long as it needed until it sounded alright, but afterwards had to limit training time to eight hours for experimentation.

Interestingly enough, even though the train and test losses decreased quite quickly and stayed relatively stable, the outputted audio had a huge difference in subjective quality between, for example, 8000 epochs and 16000 epochs for DC-TTS (e.g. unrecognizable/wrong lyrics, unexpected noises, incorrect pitch at 8000 that had been trained out by 16000). This suggests that the cost function can be modified to give greater weight to similarity to the ground truth. We would like to also train these models for longer to find the balance between overfitting and accuracy.

When testing on a smaller dataset, we found that the models often overfit by treating each time series label as a discrete label, and without more data, we would not expect the current models to generalize to unseen data very well (such as to sequences with varying pitch.)

FUTURE

In the future, for this particular problem of generating from pitch and lyric alone, we would like to experiment on different cost functions, and perhaps try another architecture which may be less expensive. For the general application of singing synthesis, we would like to find and transfer train on data with varying pitch and rhythm, so as to create a fully functional (if extremely basic) synthesizer. From this, extensions such as conditioning on features of singing style or voice timbre as well as pitch, lyric, and duration could also be worth exploring.

REFERENCES

[1] Aaron van den Oord, et. al. "WaveNet: A Generative Model for Raw Audio," arXiV:1609.03499 [cs.SD], Sep. 2016

[2] Hideyuki Tachibana, et. al. "Efficiently Trainable Text-to-Speech System Based on Deep Convolutional Networks with Guided Attention," arXiV:1710.08969 [cs.SD], Oct. 2017