VocalCycleGAN - Executive Summary

Chutian Ma, Greg Taylor, Jaspar Wiart

Overview

- Goal - Train a CycleGAN model to transform speech into singing and vice versa
- Data - Speech data from the LibriSpeech ASR Corpus. Song data from the MUSDB18 dataset. Take Fourier transform of audio to reduce size of tensors.
- Architecture - CycleGAN
- Results - We achieved good training behavior from the model. While the generator outputs do not resemble human singing, they do create an interesting “vocoder” effect that can be used musically.
- Created a Streamlit app to allow user to input audio and listen to the output.

Data

For the speech data, we used the “clean” training set available at the LibriSpeech website. This dataset contains different speakers reading from audio books. Song data comes from the MUSDB dataset which conveniently separates the different instruments. This allowed us to give our model a vocal with no music and music with no vocals.

Audio data is difficult to handle in the time domain, as sample rates are typically high (e.g. 44100 Hz). To mitigate this, we compute log mel spectrograms of the time series data. That is, we take small chunks of audio, compute the Fourier transform followed by a logarithm. The result is a (lossy) transformation to a time series of frequency information. A spectrogram of size 128x256 gives about 2.5 seconds of audio.

Model Architecture & Training

CycleGAN trains generators, G_AB and G_BA, to transform data between domains A and B. For us, those domains are human speech and human singing. In the architecture, there are additionally two discriminators, D_A and D_B, which learn to distinguish between real and fake data. The model trains by optimizing the following loss functions:

1. *Binary cross entropy* of the discriminators predictions
2. *Adversarial loss*, which is the mean-squared error of the difference between the real label and the discriminator’s decision on generated data. This trains the generator to trick the discriminator
3. *Cycle loss,* this measures the distance between a data point x and G_AB(G_BA(x)). Ideally, this cycle would result in minimal change
4. *Identity loss,* this measures the distance between G_AB(x) where x is a datapoint in domain B. Ideally, the generator would not change the data in this case.

Generators are implemented with Wave-U-Net. It is a convolutional neural network which features an encoder-decoder architecture. Each encoder layer reduces the resolution. This is especially powerful for audio tasks because it allows the model to capture patterns at different time frequency scales.

Discriminators are implemented with the MiniRocket classifier. It is designed for time series classification tasks. It transforms the input with a set of pre-defined, well selected kernels, and feeds the extracted features into a linear classifier. Its advantage is that the training does not attempt to learn the kernels, only the linear classifier. This allows it to achieve very fast processing speed while maintaining good performance.

The CycleGAN model trained well. We achieved healthy gradients even through many epochs of training. CycleGAN can often become unstable, but we achieved better results even after 500 epochs.

Results

One can hear the results of our model in our project presentation. The output of the generator certainly would not trick any human into thinking it was real singing. However, the result is an interesting and usable vocoder effect for vocals.

We included a Streamlit app in the github