华南理工大学

South China University of Technology

The Experiment Report of Machine Learning

School: School of Software Engineering

Subject: Electronic and Information

Author:
Weiwen Hu

Supervisor:
Mingkui Tan

Student ID:
202021045611

Grade:
Graduate

January 3, 2021

Speech Synthesis Based on Neural Network

Abstract—In this experiment train and run a speech synthesis model named Tacotron2. We train it on open dataset LJSpeech, and test it with any text we came up with.

I. Introduction

WE are conducting this experiment in hope of: 1. Understand the basic theory of speech signal processing; 2. Understand the application of sequence modeling in speech synthesis; 3. Understand the processes of Tacotron2 and use it in practice.

After the experiment, we get a speech synthesis model. It can transform any English sentence into speech, also known as text-to-speech (TTS) task.

II. Methods and Theory

A. Spectrogram

A spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time.

A common format is a graph with two geometric dimensions: one axis represents time, and the other axis represents frequency; a third dimension indicating the amplitude of a particular frequency at a particular time is represented by the intensity or color of each point in the image.

Spectrograms may be created from a time-domain signal in one of two ways: approximated as a filterbank that results from a series of band-pass filters (this was the only way before the advent of modern digital signal processing), or calculated from the time signal using the Fourier transform. These two methods actually form two different time–frequency representations, but are equivalent under some conditions.

Creating a spectrogram using the FFT is a digital process. Digitally sampled data, in the time domain, is broken up into chunks, which usually overlap, and Fourier transformed to calculate the magnitude of the frequency spectrum for each chunk. Each chunk then corresponds to a vertical line in the image; a measurement of magnitude versus frequency for a specific moment in time (the midpoint of the chunk). These spectrums or time plots are then "laid side by side" to form the image or a three-dimensional surface,[4] or slightly overlapped in various ways, i.e. windowing. This process essentially corresponds to computing the squared magnitude of the short-time Fourier transform (STFT) of the signal $s(t)$ — that is, for a window width ω, $\text{spectrogram}(t, \omega) = |\text{STFT}(t, \omega)|^2$

B. Mel Scale

The mel scale, named by Stevens, Volkmann, and Newman in 1937, is a perceptual scale of pitches judged by listeners to be equal in distance from one another. The reference point between this scale and normal frequency measurement is defined by assigning a perceptual pitch of 1000 mels to a 1000 Hz tone, 40 dB above the listener's threshold. Above about 500 Hz, increasingly large intervals are judged by listeners to produce equal pitch increments. As a result, four octaves on the hertz scale above 500 Hz are judged to comprise about two octaves on the mel scale. The name mel comes from the word melody to indicate that the scale is based on pitch comparisons.

A popular formula to convert f hertz into m mels is:

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{1}$$

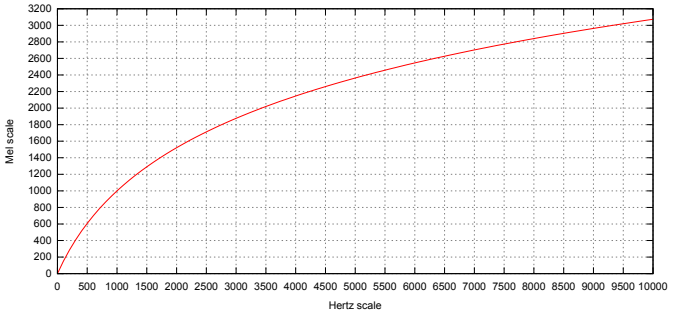

Fig. 1. Plots of pitch mel scale versus Hertz scale

With mel scale applied on frequency axis of spectrogram, we get mel-spectrogram, which is used as the training target in this experiment.

C. Tacotron 2

Tacotron 2 is a neural network architecture for speech synthesis directly from text. The system is 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. The architecture is illustrated in figure 2

III. Experiments

A. Dataset

we use the public speech synthesis dataset LJSpeech[1]. It consists of 13,100 short audio clips. A transcription is provided for each clip. Clips have a total length of approximately 24 hours.

[1]https://data.keithito.com/data/speech/LJSpeech-1.1.tar.bz2

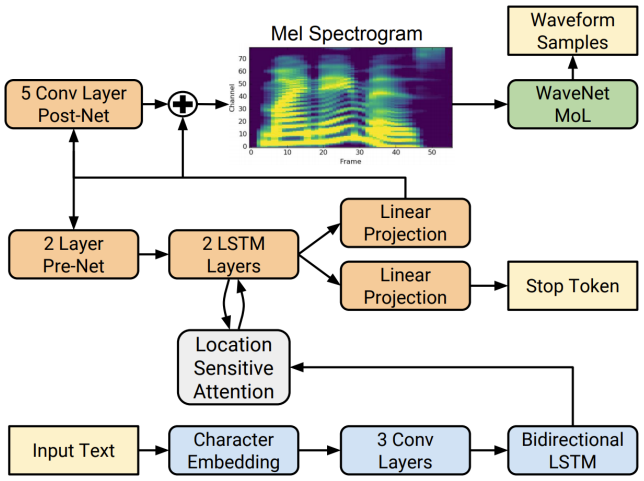

Fig. 2. Block diagram of the Tacotron 2 system rchitecture

B. Implementation

The whole pipeline is Implemented by Python, the model is based on PyTorch, librosa and some other packages are used to deal with audio signals.

I directly used the provided code. But that code is not well written, and contains some bugs. I made some small changes, and the synthesis improved dramatically. Due to the limited time and computation resources, I didn't perform the training process. But instead, I just used the provided checkpoint to run inference.

C. Results

Due to the limited time, I don't have many thing to report here. The synthesis quality is acceptable, when I listen to it, I can understand the content. But it contains many noise. So I suspect there are more bugs in the pipeline. I believe Tacotron 2 can perform better if Implemented correctly.

IV. Conclusion

This experiment gives barely acceptable results. The well-known Tacotron 2 model can synthesis speech from any English sentences. In this experiment, I'm familiarised with the speech signal processing pipeline and the Tacotron 2 model. And my engineering skills are improved.