IP Interim Report

Harry Twigg 30748119

Contents

Table of Figures

Introduction

This brief article introduces my work so far in the subject 'Comparing audio decompositions for sound synthesis with deep neural networks'. The project setup shows in more depth than the project plan the detailed network architecture and specifically how I plan to encode music data into a format that can be used in a deep learning based model. The literature review highlights the background research into what I've found to be a very broad field of research.

Literature Review

There are a variety of methods of audio signal encoding for machine learning. Different methods have different levels of resolution and levels of abstraction, they each bring their own strengths and weaknesses.

Symbolic Representation

The first research into using machine learning to generate music goes back to the 1980s (Todd, 1989) (Lewis, 1988) with the very first research on symbolic data. These higher-level representations often took the form MIDI or other musical notation. Symbolic based models provide a high-level abstraction of the musical piece, meaning that they are easier to train as the model does not have to worry the physical process to produce sound. However, they are significantly limited to music that can be described in terms of midi notation, i.e. vocals and other instruments with unique modes of being played. Todd proposed the use of an autoregressive recurrent neural network to generate symbolic music. This is a technique that many later works built upon.

I recent paper looked at applies the modern transformer machine learning architecture to symbolically generate music with long term structure. A challenge experienced in many previous pieces of work (Eck, 2018). Transformer based architectures do not rely on recurrent or convolutional based mechanisms and instead use what is called an attention-based mechanism to generate long term structure (Ashish Vaswani, 2017). In (Eck, 2018) long term music structure was generated successful.

Fourier Methods and Spectograms

In 2002 a new method of encoding audio was proposed introducing the concept of spectrograms (Matija Marolt, 2003). A spectrogram is a graphical plot of the decomposition of sound. It consists of 2 plots frequency against time and phase against time. Each point of the plots is coloured in amplitude/intensity of the decomposed audio signal at that point in time. Normal spectrograms are linear though Mel-Spectrograms are more useful for music sound synthesis. This is because Mel-Spectrograms have amplitude/sound intensity adjusted along a logarithmic scale like human hearing is. This enables machine learning models to learn how to produce audio sequences that sound more natural to a human listener. The time-domain based audio signal was divided into equal length shorter periods. A Fast Fourier Transform (FFT) is then applied to each segment, decomposing the signal at each of the timestep periods into their individual frequencies and corresponding amplitude. The complex values from the Fast Fourier Transform are complex values, giving spectrograms of frequency and phase.

Success at interpreting the phase spectrogram to date has been limited. Most models simply choose to discard the phase part of the signal and make models purely off the frequency spectrogram. This is a problem as phase is a key part of audio making the image representation fully convertible back to audio, though it is difficult to work with for several reasons: Firstly, phase appears to be random making it difficult to distinguish meaningful information from noise, secondly phase is a cyclic quality. One paper called GANSynth (Jesse Engel K. K., 2019) which attempts to overcome this problem. In this paper the difference in phase between individual timesteps of the spectrogram is calculated and 2π adjustments are made for when the phase wraps around. This difference in phase is called the instantaneous frequency and provides far more informative information than just the phase. Instantaneous frequency over harmonics for instance is expected to be constant.

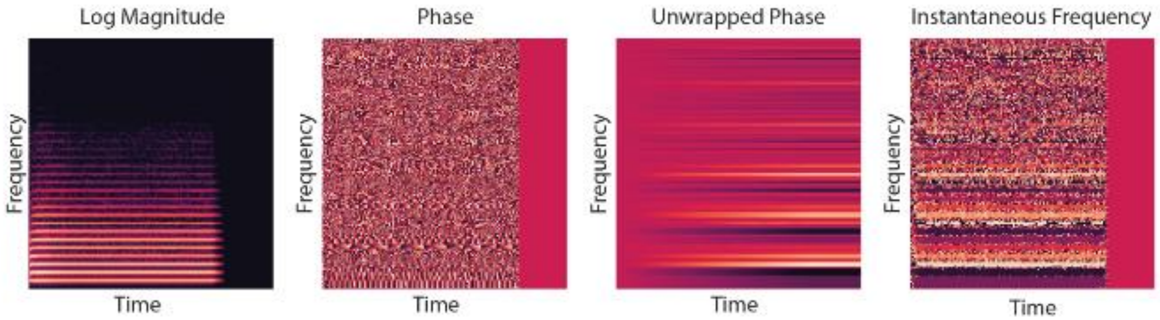

Figure 1. Mel Spectrogram of Frequency and the adjusted Phase Spectrogram

Waveform Based Models

One of the most recent ways to encode music audio is to directly present it as raw timeseries audio data. This has been recently done by OpenAI (Prafulla Dhariwal, 2020). An autoencoder compresses 44kHz input audio to a discrete space using a technique called Vector-Quantized Variational Autoencoders (VQ-VAE). Several independent VQ-VAE Levels are used retaining different levels of resolution up to 128x encoding. Lower levels capture local music structures e.g. timbre and local pitch, whereas higher levels capture higher level long range music structure features. Each level is then trained. The autoregressive sparsed transformer-based models learn the probability distribution of the VQ-VAE levels that unsampled each previous level. The model can be conditioned on lyrics genre, artist. After training, new songs can be unsampled through each level to give new raw music audio.

Although local musical coherence, timbre among other things is good, longer term musical structure is not present. The upsampling process also introduces significant noise into the final audio. Another big disadvantage is the lack of parallel sampling and autoregressive nature of the model meaning that it takes multiple hours to produce one minute of music. The model also functions much like a black box, preventing us from gaining any important information into how it is synthesising its audio.

Differentiable Digital Signal Processing

Differentiable Digital Signal Processing (DDSP) are differentiable versions of traditional digital signal processing elements that can be integrated into machine learning models. They were first released a year ago (Jesse Engel H. H., 2020). DDSP elements are based off traditional synthesiser-based components. The DDSP network is modular in design and consists of feedforward components as opposed to recurrent based networks, this allows parallel training and generation of samples. The model aims to take advantage of what the authors call inductive biases of music instead of making the model figure out all the features of music itself. For example, instrument have sinusoidal vibrations at multiples of the fundamental frequency, this is where the link with traditional digital signal processing elements come in.

The Initial Model

The initial DDSP model by (Jesse Engel H. H., 2020) employed a modified autoencoder decoder setup where the autoencoder was trained to minimise reconstruction loss in outputted synthesised audio. Training data is fed into the autoencoder in the form of Mel-spectrogram images. Loudness is statistically determined outside of the autoencoder. The autoencoder then attempts to extract the fundamental frequency and what is called latent coding representing residual information. The decoder than uses the F0, Z, and loudness as control values for the filtered noise and oscillator synthesisers.

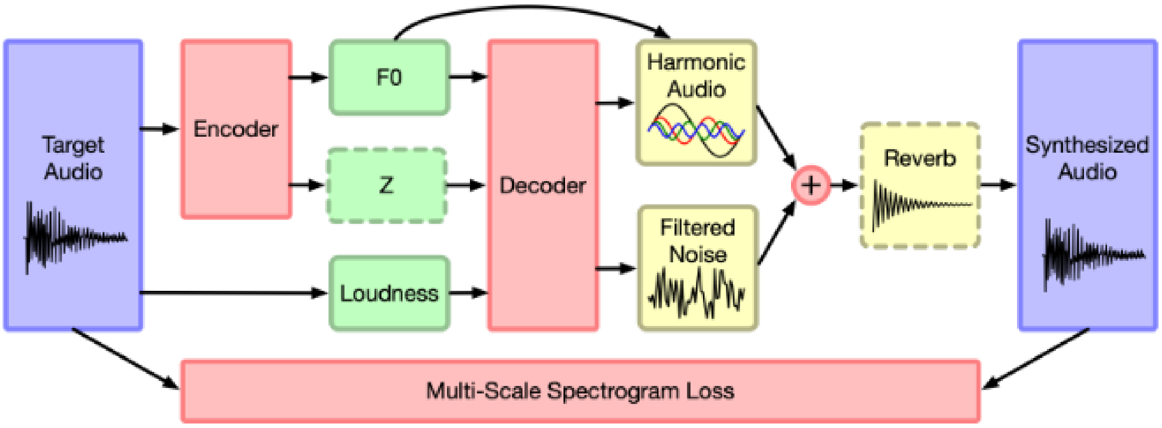

Figure 2. The DDSP Network Architecture from the original DDSP paper (Jesse Engel H. H., 2020). Red components make up the neural network architecture

Due to the modular design, certain features can be explicitly controlled for example room acoustics. Some networks would implicitly pick up room acoustics. This method may introduce unnecessary mode covering, increasing potential training time. The DDSP model explicitly defines room acoustics using a reverberation synthesiser. Finally, the target and synthesised audio spectrograms are compared using a multi-scale spectral loss process.

The DDSP decoder very quickly learned how to resynthesize datasets for a single instrument that sounded like the original audio sample. Due to the interpretable and modular design of the mode, individual factors such as timbre, pitch or loudness could be varied whilst keeping the others constant. This was because the model was only conditioning on pitch, loudness, and residual z relates to information.

Singing Voice Synthesis Using DDSP

Further work on top of the initial paper has been conducted by a new team to produce singing voice synthesis (Erkut, 2021). This paper builds on existing DDSP work to make a more complex output of singing like output.

Even on a small dataset of just fifteen minutes of audio, promising results were obtained. Timbre transfer was possible. Unfortunately, when the model was forced to recreate sung lyrics, it was unintelligible, appearing to make stuttering noises. The model managed to obtain correct loudness and pitch of sound (except for the occasional pitch artifacts) but had no understanding of phonetics (i.e. the certain words that make up speech). The paper suggests several steps for further work:

- Phonetic condition of the model to model the nuances of human singing and to model the lyrics.
- Using synthesizers more suitable to modelling the human voice
- Pre-processing of fundamental frequency to remove pitch artefacts.

Speech Synthesis

The only other recent paper on using DDSP for vocals relates to speech synthesis, although not directly designed for music, the paper introduces a slightly different system that modifies the DDSP architecture to more closely match the human voice (Giorgio Fabbro, 2020). Instead of using an autoencoder, the network uses a series of convolutional neural networks. The method yields highly accurate results (although it is still noticeable the output has been synthesised). Timbre was accurately measured, though consonants still sounded off. They trained the network so that the filtered noise generator generates consonants, and the harmonic oscillator gives vowels. The convolutional/recurrent structure of the model allows it to pick-up long-term patterns in human speech, for example how certain words are said. The convolutional neural networks are used to train the harmonic and filtered noise generators respectively. What's interesting about it is the teacher forcing through the feeding of pitch directly into the harmonic oscillator as a source of ground truth, this has not gone through the neural network and has been obtained straight from the spectrogram. The forcing of same output pitch enabled the network to focus on picking up the subtleties in speech, for example how different sounds are produced, as opposed to determining how to extract pitch information.

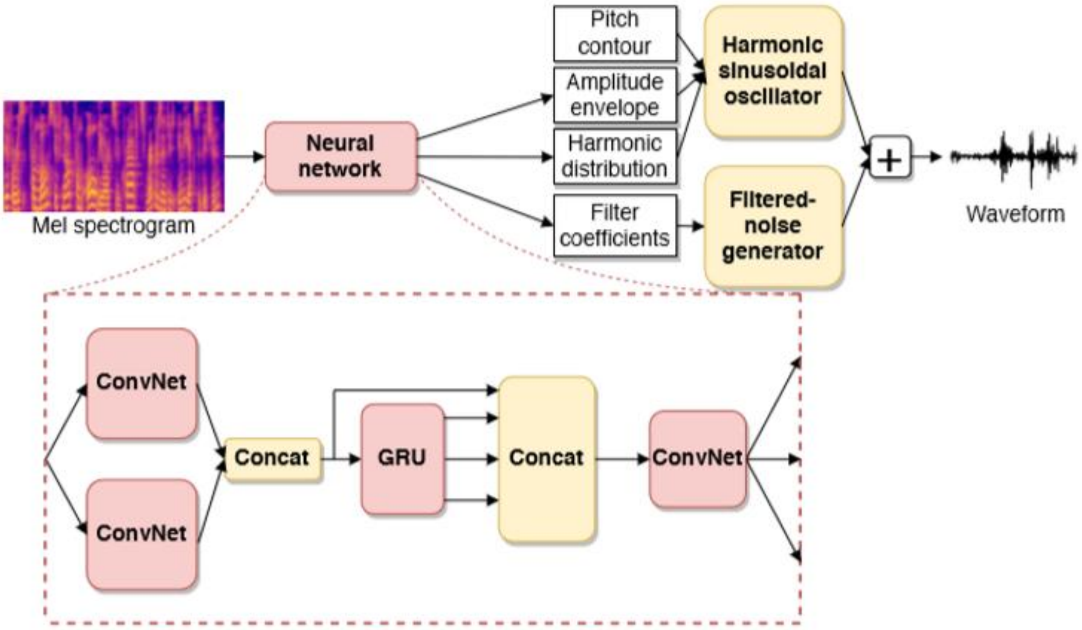

Figure 3. The modified network architecture of the DDSP neural vocoder

Project Plan

My plan is to try and build a new system for encoding and processing singing using Mel-Spectrograms for singing synthesis. Specific objectives related to this goal include the following:

- Successfully encode audio spectrum data to a neural network
- Incorporating the phase data into the input spectrogram in some way, perhaps by using the instantaneous phase diagram (further goal)
- Set up the convolutional neural network based off the model provided by Giorgio Fabbro et al. that then feeds data to harmonic and filtered noise oscillators.
- Implement a loss function to train the network
- Train the network on a test dataset and produce output waveforms using the DDSP framework

I am building on the work of the Differential Digital Signal Processing models for singing and voice further (Erkut, 2021) (Giorgio Fabbro, 2020). It could detect timbre and pitch of human voice accurately. However, it lacked the ability to make intelligible lyrics. This paper in its conclusion made several recommendations for future work that I am going to take on board. A lot of information is present in sung lyrics besides pitch and loudness that the model failed to pick up. I need to develop some way of conditioning and encoding data at the language level so that the model can learn the nuances of phonetics. The paper by Giorgio Fabbro et al. is explicitly referenced as an example of what they plan on doing in further research. I am therefore going to attempt to make a model that decomposes and analyses the input audio in an architecture that shall be like the diagram shown in Figure 3, unfortunately they did not publish their code, so I am going to have to try and reverse engineer the kind of network design from scratch.

I want the model that I produced to be able to be modular so that individual parameters e.g. pitch, timbre or amplitude of the speaker can be modified. The encoder is going to use a recurrent neural network to decompose the input Mel-Spectrogram into a set of control variables for each timestep n, specifically (credits for eqn. go to (Giorgio Fabbro, 2020)):

- Pitch contour $f_1(n) \in \mathbb{R}^+$ for each time step
- Amplitude envelope $A(n) \in \mathbb{R}^+$
- A distribution over harmonics $c_k(n)$ which gives amplitudes for each harmonic $A_k(n) = A(n)c_k(n)$

The harmonic oscillator shall superpose sinusoidal signals for H integer harmonics where the output of the harmonic oscillator is:

$$y(n) = \sum_{k=1}^{H} A_k(n) \sin(\phi_k(n))$$

Phi is known as the instantaneous frequency and it must be chosen. This is best calculated so that the highest harmonic i.e. H * fundamental frequency is at the Nyquist Frequency.

$$\phi_k(n) = 2\pi \sum_{i=0}^{n} f_k(i)$$

Fabbro et al. use a time varying filter bank for the filtered noise generator and let the neural network output parameters for it.

I have yet to find a dataset to use, but I will probably choose to pick one based purely on singing samples of one artist. This is so that the model will not get confused between different artists.

Incorporating instantaneous phase into the network is a further goal that I may implement if I manage to get a functioning network, this will involve the use of a concat operation of convolutional layers for the 2 separate spectrograms. This is a standard feature of the TensorFlow library.

References

Ashish Vaswani, N. S. (2017). *Attention Is All You Need*. Retrieved from https://arxiv.org/abs/1706.03762

Eck, C.-Z. A. (2018). *MUSIC TRANSFORMER: GENERATING MUSIC WITH LONG-TERM STRUCTURE*. Retrieved from https://arxiv.org/pdf/1809.04281.pdf

Erkut, J. A. (2021). *LATENT SPACE EXPLORATIONS OF SINGING VOICE SYNTHESIS*. Retrieved from https://arxiv.org/pdf/2103.07197.pdf

Giorgio Fabbro, V. G. (2020). *SPEECH SYNTHESIS AND CONTROL USING DIFFERENTIABLE DSP*. Retrieved from https://arxiv.org/pdf/2010.15084.pdf

Jesse Engel, H. H. (2020). *DDSP: Differentiable Digital Signal Processing*. Retrieved from https://openreview.net/attachment?id=B1x1ma4tDr&name=original_pdf

Jesse Engel, K. K. (2019). *GANSYNTH: ADVERSARIAL NEURAL AUDIO SYNTHESIS*. Retrieved from https://openreview.net/pdf?id=H1xQVn09FX

Lewis, J. (1988). *Creation by refinement: a creativity paradigm for gradient descent learning networks*. Retrieved from https://ieeexplore.ieee.org/document/23933

Matija Marolt, A. K. (2003). *Neural Networks for Note Onset Detection in Piano Music*. Retrieved from https://www.researchgate.net/publication/2473938_Neural_Networks_for_Note_Onset_Detection_in_Piano_Music

Prafulla Dhariwal, H. J. (2020). *Jukebox: A Generative Model for Music*. Retrieved from https://arxiv.org/abs/2005.00341

Todd, P. M. (1989). *A Connectionist Approach To Algorithmic Composition*. Retrieved from https://www.jstor.org/stable/3679551