Week 3 Report

What I've done this week

- Finished the *Audio Signal Processing for Machine Learning* course.
    - Good overview of some of the key features used in ML with audio such as (Mel-)spectrograms, MFCCs and dealing with audio chunks in frames.
    - Terms frequently come up in the literature, so it has been useful to get some hands on experience with extracting these features.
- Read and summarised *AudioGen: Textually Guided Audio Generation (2023)*.
    - Main takeaways given below.
- Re-read *Differentiable Signal Processing With Black-Box Audio Effects (2021)* with a focus on how the gradient estimation for effect parameters work.
    - Can go into detail about this at Monday meeting.
- Didn't have time to read DALL-E paper. However, Computerphile released a video this week which gives a high level overview of how it works.
    - https://www.youtube.com/watch?v=1CIpzeNxIhU
- Found an AudioCommons model which extracts various timbral attributes (*hardness, depth, brightness, roughness, warmth, sharpness, booming, and reverberation*) from an audio signal.
    - Can play with the model here: https://www.audiocommons.org/2018/09/05/timbre-sound.html
    - Code available here: https://github.com/AudioCommons/timbral_models
- Downloaded and played around with some IDMT datasets - Bass, Guitar, Piano, Drums and Audio Effects. The audio effects dataset may be particularly useful for project.
    - Contains sounds produced from 10 different effects using both guitar and bass.
    - Other information such as parameter values and pitch information also provided.
    - Could be potentially used alongside the AudioCommons timbral model and Black-Box DDSP architecture with some sort of word embedding?
    - NSynth dataset also has metadata about note qualities directly included. However, there isn't a dry/wet signal comparison like the IDMT dataset and sample rate is low (16kHz).
        - https://magenta.tensorflow.org/datasets/nsynth#note-qualities

Questions

- No specific questions this week as I'm still just gathering information. Any feedback on what I've done, or anything I've missed would be very welcome though!

Plan for next week

- Same creator of the *Audio Signal Processing for Machine Learning* course also has another course called *Generating Sounds with Neural Networks*. This course seems more hands on with actually building a neural network pipeline using features from the previous course. Culminates in a model which produces spoken digits (a bit like MNIST for computer vision). Planning to make a start on this this week.
    - https://www.youtube.com/playlist?list=PL-wATfeyAMNpEyENTc-tVH5tfLGKtSWPp

- More in-depth reading into how word embeddings are actually implemented in models like DALL-E and AudioGen for conditioning and which language model might be appropriate.
- Spend time learning how adversarial losses, LSTMs and Transformers are implemented.
- I had started looking into how non-technical descriptive language about musical timbre can be quantified. I've added a couple of interesting looking papers to Zotero which I hope to have a look at this week.

Current state of project

- Narrowed down project idea to text-to-audio-effect synthesis.
- Still getting to grips with handling audio data for deep learning.
- I imagine it might be at least another 2-3 weeks before I start any proper implementation of a model.
- First need to make sure I have a suitable dataset for training.

AudioGen paper summary

Why is text-to-audio a difficult problem?

- There is a lack of large datasets compared to computer vision.
- Audio signals have very long sequencies/dependencies even with signals that are a few seconds long due to high sampling rates.
- Generating novel sounds requires composing sounds which may not appear together in the dataset.
- Example given in the paper: “a dog barks while somebody plays trumpet in a busy street”.
    - Three different categories of acoustic content. How do we weigh them?

Model architecture:

- First an autoencoder is trained to reproduce a raw 1D audio signal. A variety of losses are used including temporal/frequency L1/L2 distances and complex-valued spectrograms.
- The encoder part of the trained network is used to create audio tokens. These are concatenated with word embeddings obtained from the T5 language model. Cross attention is applied, then the decoder part reconstructs the signal.
- Classifier-free guidance is used to amplify the association between the word embeddings and reconstructed audio.

Dataset

- Made up of multiple audio dataset, some of which have tags while others have full audio captions.
    - In the former case, the tags are concatenated into a single sentence.
        - *Example*: [“dog”, “bark”, “park”] -> “dog bark park”.
    - In the latter case, stopwords are removed and lemmatization applied to produce a similar single sentence output.
        - *Example*: “A dog is barking at the park” -> “dog bark park”.
- To help improve the model's ability to create novel sounds data augmentation is performed. Audio signals are overlapped and their text descriptions concatenated.
- In total the dataset contains over 4k hours of audio.

Thoughts/takeaways from paper

- Sample rate is quite low (16kHz) and would not really be usable in the music audio domain which generally requires a rate >= 44.1kHz.
- The dataset is extremely large and I don't think that there will be nearly as much usable audio data for my project (IDMT Audio Effects dataset is ~30 hours @ 44.1kHz).
- The concatenation of tags is an interesting way to create word embeddings from descriptions. However, the authors mention that this results in issues with creating temporal ordering in generating audio (can't do things like "dog barks **then** birds chirp").
- I think the data augmentation method might be an interesting thing to try with whatever dataset I end up using.
- Unfortunately the code isn't open source yet, but I think I have a decent understanding of the high-level architecture of the model.