A Deep Learning Approach to Musical Effects

Kieran McCool

2142393m

Proposal

Motivation

Most industries began to move away from analogue equipment long ago, however the music industry remains faithful to a great number of aging technologies. Many professionals feel that digital modelling of these result in less pleasing audio. Some would go as far to say that modelling fails to pick up on subtle details of the effect, for instance the decay effect you get from a tape delay, or the non-linear response to volume from an analogue distortion.

Deep Learning has proven itself proficient in modelling similar non-linear systems and is beginning to make progress in the field of speech synthesis and audio classification. Perhaps it can be the key to modelling these analogue effects accurately.

Aims

The aim of the project is to apply Deep Learning to the field of musical effect. This will take the form of training a neural network to reproduce an effect by providing it with a clean non-processed signal and having it predict the transformation needed to apply the effect. Likely effects that will be modelled are distortions, chorus, reverb, delay and filters/noise gates. For simplicity, the project will model VST effects rather than analogue ones, however success in this domain would suggest real potential for use on analogue effects.

Progress

- Created a pipeline for generating test data automatically.
- Python chosen as language and PyTorch for machine learning framework.
- Used PyTorch to create a neural network consisting of both convolutional and fully-connected layers which can accept audio data.
- Produced some promising results with distortion effects with the convolutional network.
- Added support for GPU acceleration which significantly improves the training rate on my desktop PC.
- Provided a user-friendly, well documented command line interface for all components of the project to ensure it will be useful to others when complete.

Problems and risks

Problems

- Automating the VST process was difficult as there are very few tools available which can do this via command line.

    - Had to use a poorly documented feature of Reaper (a digital audio workstation) which allowed command line invocation of the batch mode.
- Poor support for VST on Linux and PyTorch doesn't support Windows, so until recently much of the development had to happen on a MacBook which is significantly slower than on my Desktop
    - Found that Reaper works flawlessly under Wine and the pipeline only needed a few changes to support this, allowing me to use GPU accelerated desktop with significantly more RAM than MacBook.
- Audio needs to be kept uncompressed to eliminate compression artifacts as a variable in output.

Risks

- Many different effects to test, a comprehensive yes or no to the question of how good deep learning is at modelling musical effects would be impossible. **Mitigation:** Will choose a few effects, probably 5 to 10, ranging from distortions, chorus, reverb, delay, and noise gates and validate success with these.
- Difficult to quantify how good an effect sounds or how well it models the test data. **Mitigation:** A combination of Melspectograms and ABX testing will give as close to a quantitative answer as possible.
- Not very many similar projects exist, difficult to tell what network architecture would be useful, or good input sizes etc. **Mitigation:** A lot of experimentation and making use of what little there is in the way of similar projects (DeepSound, WaveNet, etc)

Plan

- Week 1-3: Train the network on selected effects, archiving the melspectogram output and wav files for analysis and experimenting with network architectures.
    - **Deliverable:** A range of files representing the networks performance at a given effect. This will be used in the future for ABX testing and analysis. It is likely this will be an ongoing activity but the bulk of it should be done in this period.
- Week 3-4: Research methods of conducting evaluation properly
    - **Deliverable:** A process for conducting ABX testing as well as some way of comparing melspectogram (possibly through the use of GIFs or similar)
- Week 4-5: Finalise implementation
    - **Deliverable:** A fit for purpose set of command line tools which are easy to use and understand, and could be picked up as a starting point or example for similar projects in the future. It would also be good useful if I could provide some model checkpoints which can be used to create specific effects to provide practical application to the project.
- Week 6-8: Start evaluation projects, consisting of ABX testing, spectogram anaylsis and graphing loss over time.
    - **Deliverable:** A dataset which documents the effects of network architecture as well as the success of the project in modelling effects.
- Week 8-10: Write up
    - **Deliverable:** A first draft of the dissertation to be submitted to supervisor before final deadline.