Title: Emulating Guitar Pedal Effects Digitally using Machine Learning with Pedal Knob Control
Name: Kyle Antipuesto
Supervisor: Dr Ben McCanna
Module Code: COM3610

One pastime that I've enjoyed throughout most of my life is playing the guitar, but only recently have I begun investing in modifying the tone of an electric guitar and learning how to achieve a specific sound with certain effects. In particular, the market for guitar effect pedals has started to shift from analogue electrical circuitry to digital signal processing. Over the years, the digital reproduction of analogue effects has evolved. This evolution has resulted in impressive replication with insignificant differences. Examples of these include Neural DSP's Quad Cortex and BOSS's Plugout FX Pedal (Both being able to recreate multiple pedal effects due to DSP).

I personally wanted to delve deeper into the possibilities of modelling guitar effects digitally, so I decided to use machine learning to help accurately model these pedal effects digitally by having it train on data based on the real pedals themselves. One issue with current digital models of effects is that, due to their complete dependence on a digital circuit, they disregard replicating certain sonic nuances that come with physical, electrical analogue circuits, which many veterans in the hobby have come to adore. With training a model based on the real physical circuits themselves, those nuances can be captured within the data and be replicated in the model.

Scientific papers that I have read (Paper on Amp FX Emulation using Machine Learning) have investigated similar ideas. The approach is, however, completely black-box. Their focus was on exploring how well machine learning can replicate analogue effects. Instead, I want to focus on developing software that is usable and versatile for users. To further build upon this paper, I would like to include dynamic control over the knobs that you find on a pedal itself. For the example pedal that I'm going to use for this project (the Ibanez Tube Screamer Overdrive Pedal), The main knobs that control the unique characteristics of the pedal's effect are the overdrive knob (The knob that controls the clipping of the audio signal and therefore how heavily distorted the sounds comes out) and the tone knob (The knob that controls the high frequencies of the signal)

In terms of feasibility, the paper mentioned in the previous section proves that the use of machine learning for tone replication is definitely viable. My task is to provide a more user-friendly and versatile approach and implementation to the current work.

One current issue is related to my method of recording parallel test samples. My current approach to gathering data is to record some guitar playing through a loop pedal I own and have it recorded through my audio interface twice (once dry and the second time through the FX pedal). However, the quality of audio varies between the guitar plugged directly into the audio interface and a recording sample played from the loop pedal into the audio interface, so the training data may not be as accurate as it could be. I can solve this by plugging my guitar into an instrument cable splitter into an audio interface with two inputs. However, my current audio interface only has one input and purchasing another one may not be realistic.

Another issue that may arise is the learning curve of new frameworks. Pytorch and JUCE are both noted to be used within the project, but they are both fairly new to me. Pytorch may not be too difficult to get to grips with because I'm comfortable with Python. I wanted to use JUCE as it is the industry standard for developing audio plugins; however, it is based on C++, which I have no prior experience with. I do plan to get a sufficient grasp of however much C++ I need to accomplish the task relating to my project (if worse comes to worst, there is also a Python alternative for JUCE that I can use: Popsicle).

For evaluation, I wanted to use both subjective human testing and objective accuracy testing. The subjective test involves using ABX testing. This is accomplished by having experienced guitarists try to guess which of the digital model and physical pedal is which by switching between the two and having them compare at different knob settings. The objective test involved comparing the visual and numerical properties of the physical pedal signal and digital model signal (decibel and spectral differences).

A high-level structure of my plan:

- Produce parallel multiple recordings of guitar playing: one with no effects applied (dry) and one run through the FX pedal I'm attempting to recreate (wet), and also label with their knob settings.
- Edit the recording pairs to suit as training data for the neural network.
- Develop the neural network to obtain a working emulation model.
- Create a VST audio plugin with knob control that can implement the ONNX model.
- Evaluation
-

Setup Requirements

- Audio Interface
- FX Pedal
- Guitar
- Instrument Cables
- DAW Software
-

Literature in the process of reading

- Past Dissertation on Bass DSP
- Paper on Amp FX Emulation using Machine Learning
- Grey Box Modelling of a Phaser Effect