# Music Generation

Christie Gan, Andreia Nichols, Carl Parker, Hasnah Said, Preston Stringham

# Background

Current music generation packages are difficult to interface with for users unfamiliar with music theory or model training. These packages usually provide extensive functionality and with that also a lot of complexity.

We aim to simplify the training process by abstracting away technical details that are necessary for the models to be properly trained. Our deliverable will be a simple Python package for people to generate music similar to their music taste.

# Use Case

Recently, content creators have had issues with finding copyright-free music to use in their content (e.g., background music, jingles).. By using our package, content creators could easily generate unique music that they can use in their own content.

For example, Carl wants to start a podcast and have opening music, but doesn't want to go through the hassle of buying or creating his own music. He could use this package to create his opening song.

# Libraries: Pre-processing

Mido

- Can only parse MIDI files
- Lightweight and easy to interface with
- Well-documented
- Only built for parsing MIDI
- Created by a small team of developers

Music21

- Can load music from multiple file formats (e.g MIDI and Musicxml)
- Powerful and easy-to-use
- Well-documented
- Lots of pre-built analysis tools
- Managed and developed by MIT

# Example: Pre-processing

```
import mido
for file in self.files:
    messages = []
    for message in mido.MidiFile(self.path + file).tracks[0]:
        messages.append(message)
```

```
import music21 as mus
music21objects = []
for file in files:
    s = mus.converter.parse(file)
    music21objects.append(s)
```

# Libraries: Models

Magenta

- Contains various pre-trained models that follow different styles of music generation
- Written using Tensorflow
- Uses LSTM layers in basic RNNs
- Exports MIDI or musicxml that can be converted to MIDI

Folk-RNN

- Contains pre-trained models but only generates folk music
- Written using Theano
- Theano is no longer being supported by its original creators
- Uses LSTM layers in a RNN algorithm
- Exports MIDI files

# Example: Models

Magenta

```
BUNDLE_PATH=<absolute path of .mag file>
CONFIG=<one of 'basic_improv', 'attention_improv' or 'chord_pitches_improv', matching the bundle>

improv_rnn_generate \
--config=${CONFIG} \
--bundle_file=${BUNDLE_PATH} \
--output_dir=/tmp/improv_rnn/generated \
--num_outputs=10 \
--primer_melody="[60]" \
--backing_chords="C G Am F C G Am F" \
--render_chords
```

Folk-RNN

```
python sample_rnn.py --terminal metadata/folkrnn_v2.pkl
```

# Our Choice

Pre-Processing

- Music21
  - Large library with lots of pre-built analysis tools
  - Easily parses both MIDI and MusicXML

Modeling

- Magenta
  - Well built and trained models with additional parameters to make generated music sound better

# Questions?