Music Generation with Machine Learning

By Matt Ng
General Assembly DSI 25

Capstone Project

Contents

- Problem
- Background
- Data Source
- Data Preprocessing
- Models and Metrics
- Generated Music

Problem

Difficulty in finding thematic music

Making original music involves high skill cap

Original music may also be too similar to current works

May also be able to reconstruct styles of famous composers

Background

Asymmetry in time-frame of several famous composers

Lack of further works by composer

Incomplete works by composers

Increasing levels of copyright claims to specific melody

Mastery of musical composition may takes years

Data Source

Using Beethoven’s Piano Sonatas (Early to Mid-period works)

Taken from

https://www.classicalarchives.com/midi/composer/2156.html

Data Cleaning not required as MIDI data is an actual recorded performance

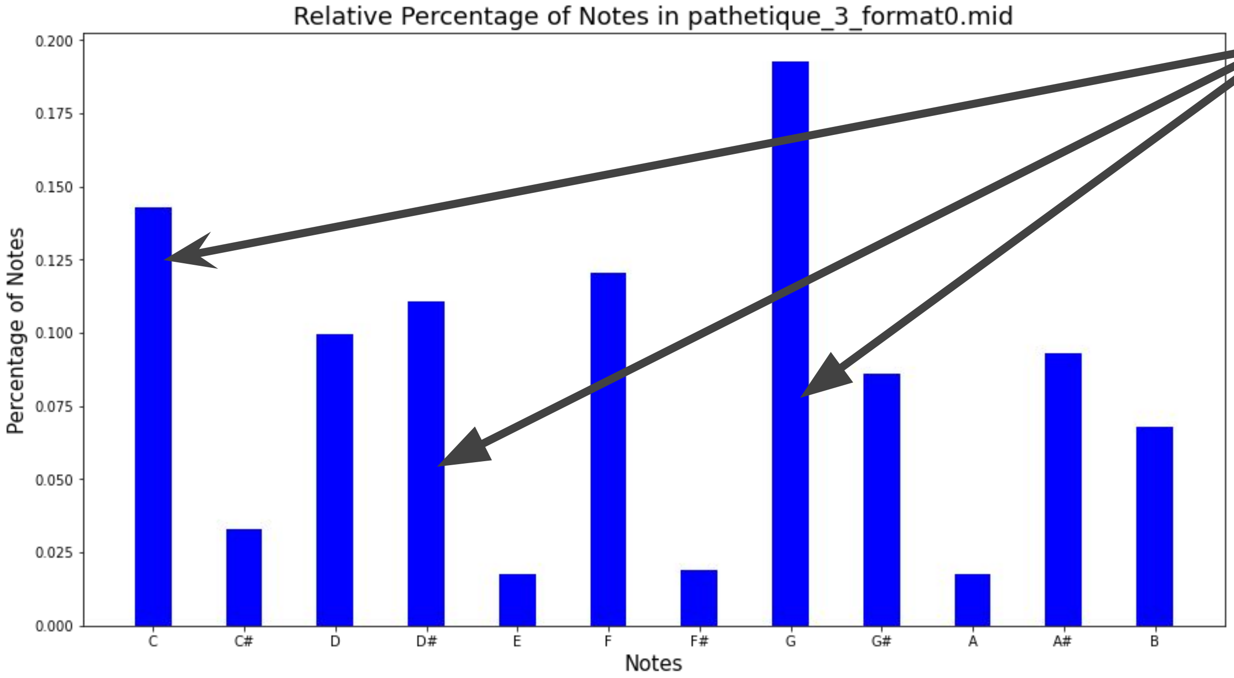
Initial Exploration of Data

3 highest spikes in notes chart typically indicates the key signature of the piece

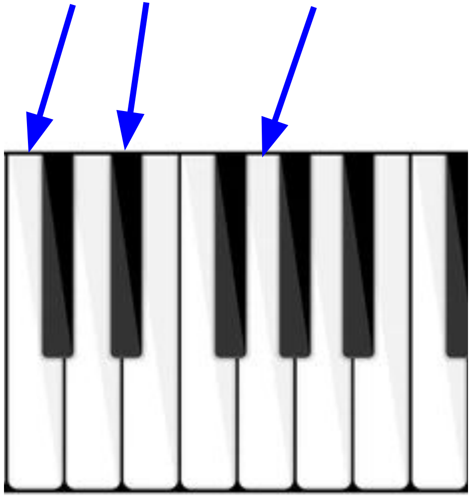
Different for every MIDI data

Taken into account with our MIDI library

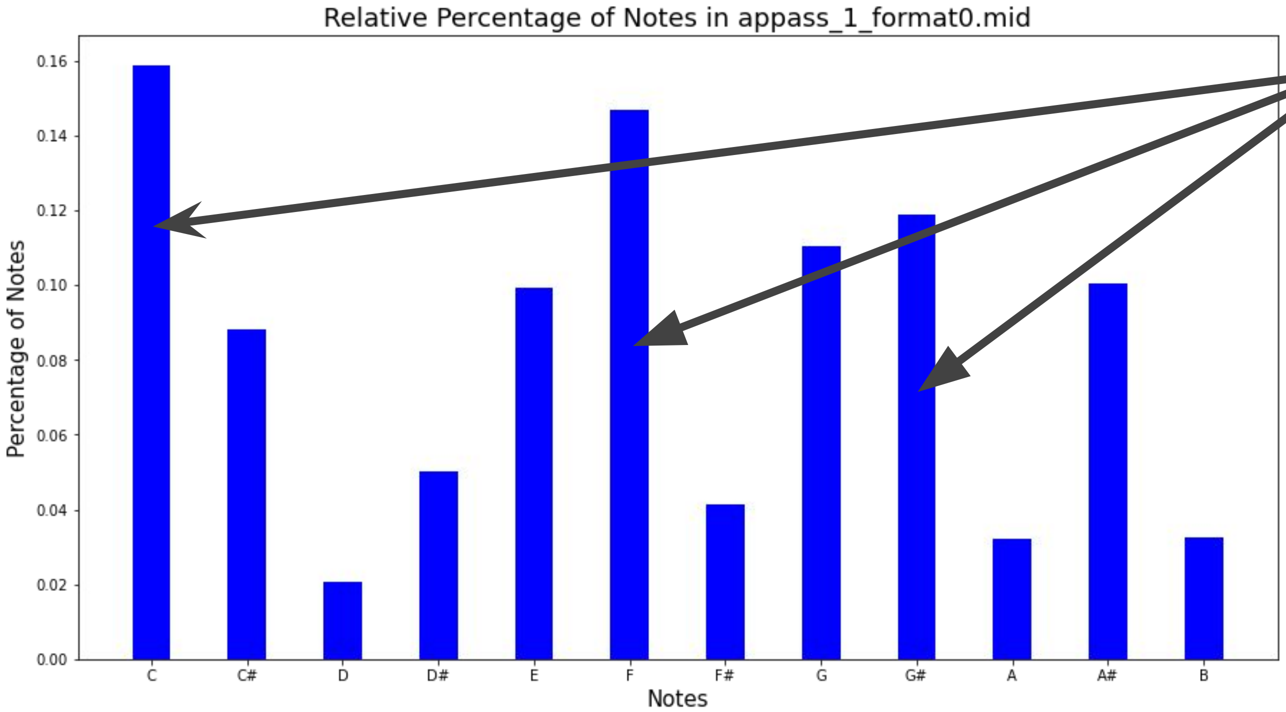
Piano Sonata No. 8 in C minor, Op. 13 ‘*Pathetique*’
3rd Movement



Forms C minor Chord
C, D#(E♭) and G

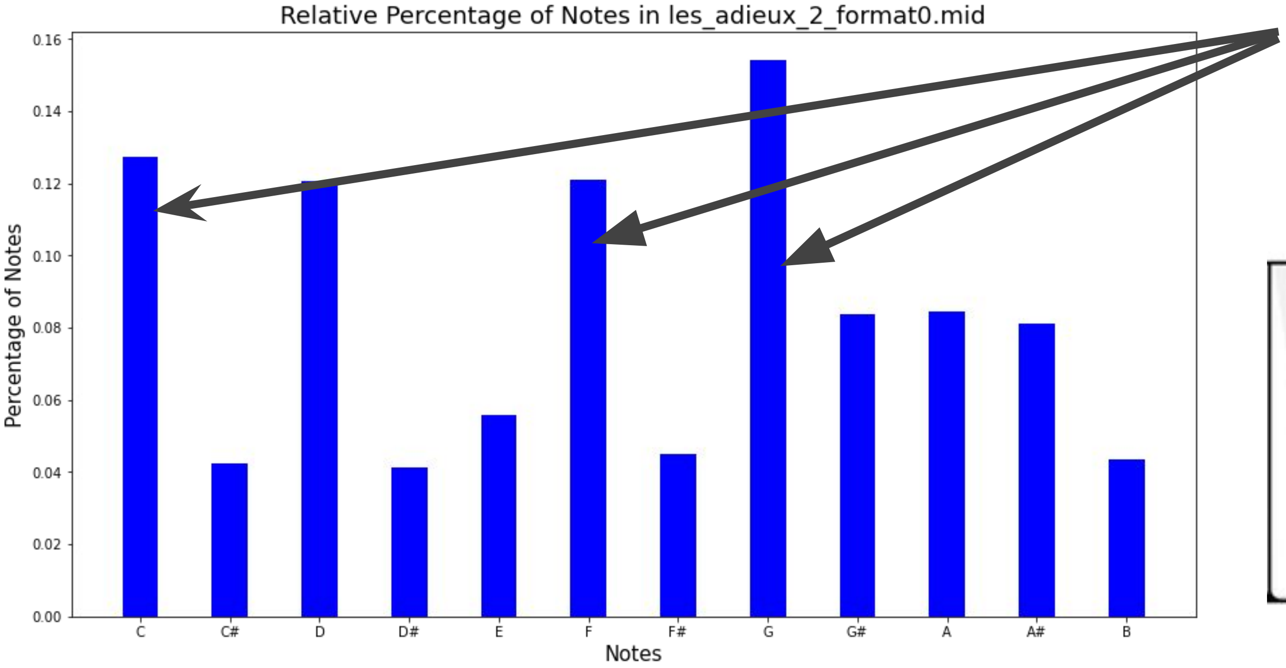


Piano Sonata No. 23 in F minor, Op. 57 ‘*Appassionata*’ 1st Movement

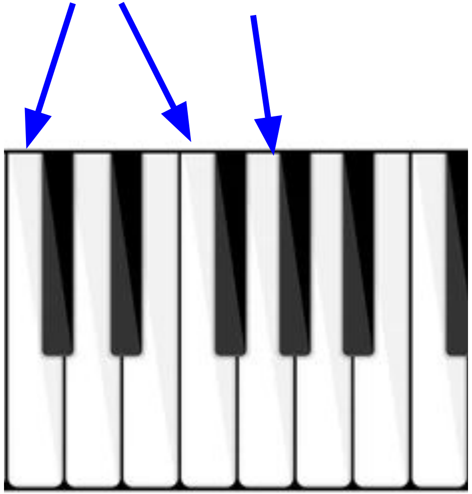


Forms F minor Chord
C, F and G# (A♭)

Piano Sonata No. 26 in E ♭ major, Op. 81a 'Les Adieux' 2nd movement



Does not form a chord
C, F and G



Data Preprocessing

Raw MIDI Data

Direct from web archive

PrettyMIDI Library

Handles the musicological component of the music*

One-hot encoder

Takes our processed data and arrange into a readable array

Machine Learning Model

Model will be trained based on input MIDI data

**other libraries, such as Music21, are available too*

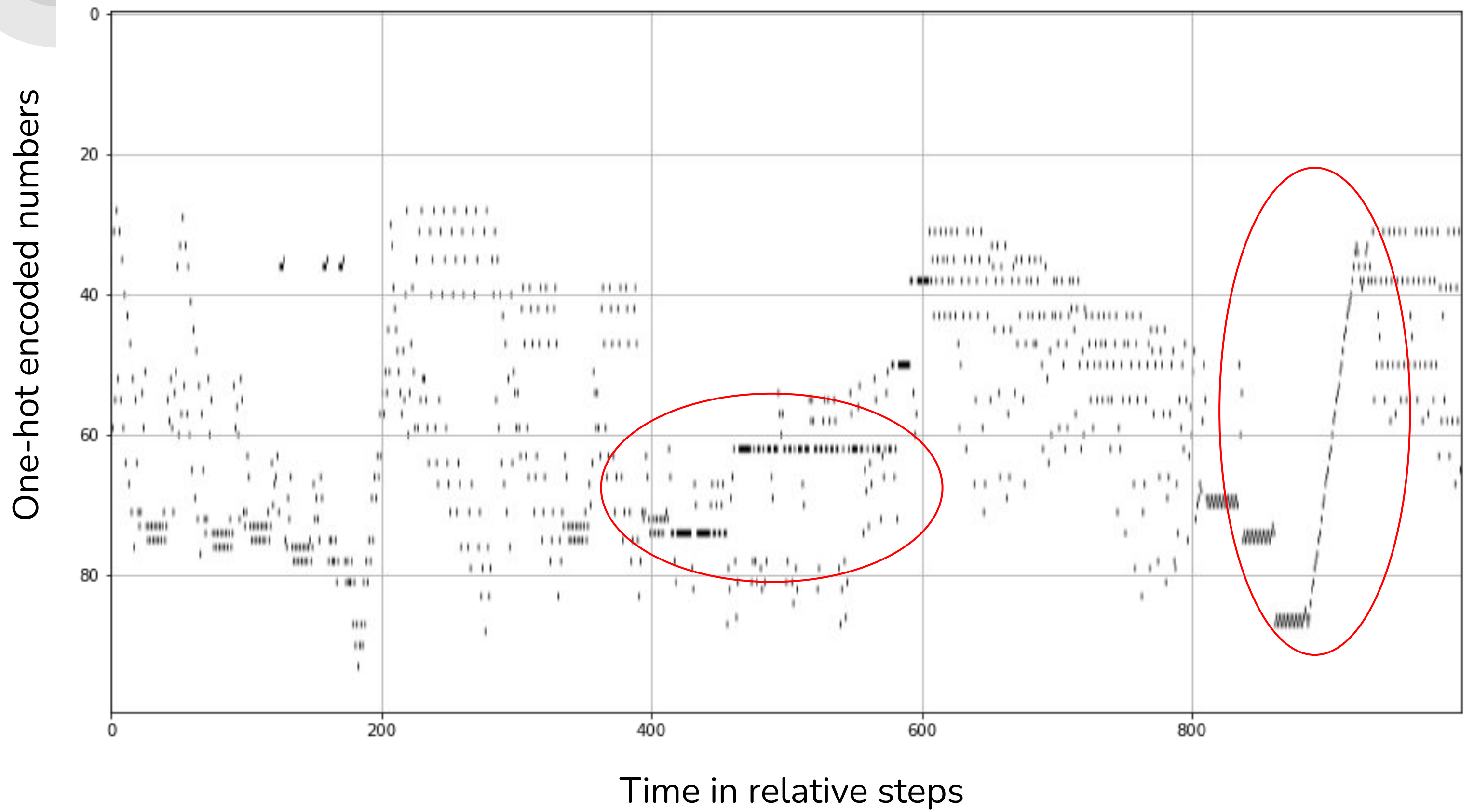
Observations from Data

Highly ordered and clear display of rules

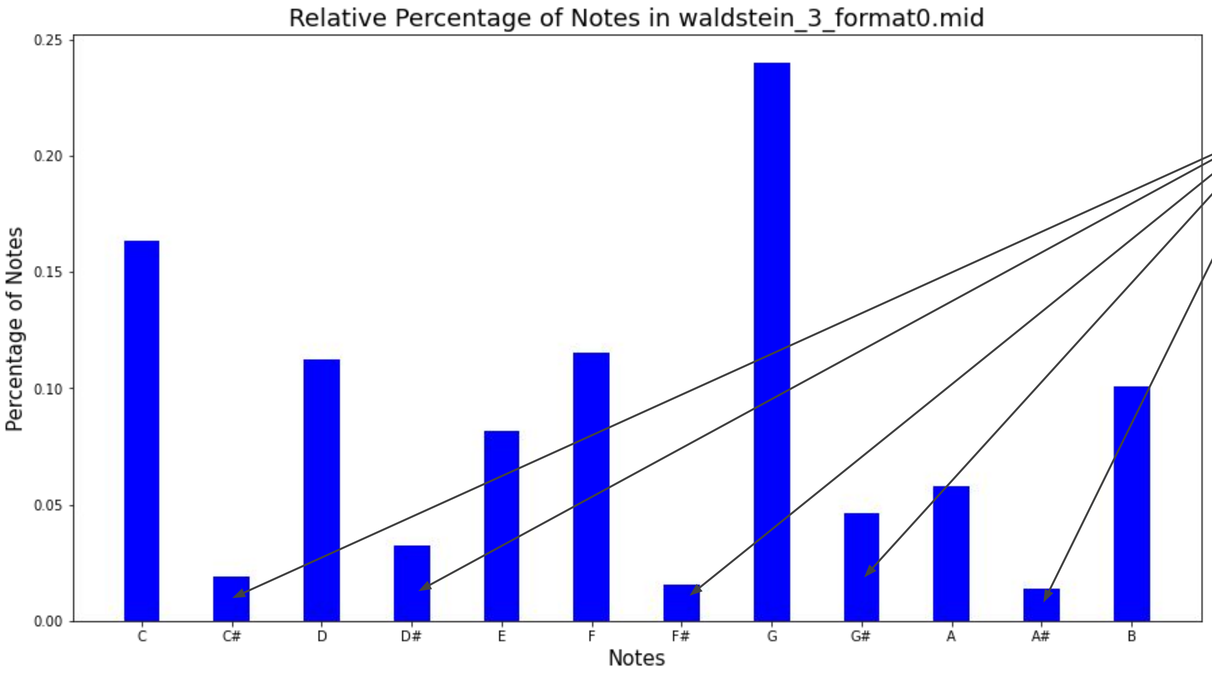
Variable movement in notes

Polyphonic data

One-Hot Encoded Map of a Training Data



Piano Sonata No. 21 in C major, Op. 53 'Waldstein' 3rd movement



Low note occurrence
Out of key notes

Process and insert demo track of appassionata 1 for baseline listening

Modelling

Raw MIDI Data

Direct from web archive

PrettyMIDI Library

Handles the musicological component of the music*

One-hot encoder

Takes our processed data and arrange into a readable array

Machine Learning Model

Model will be trained based on input MIDI data

**other libraries, such as Music21, are available too*

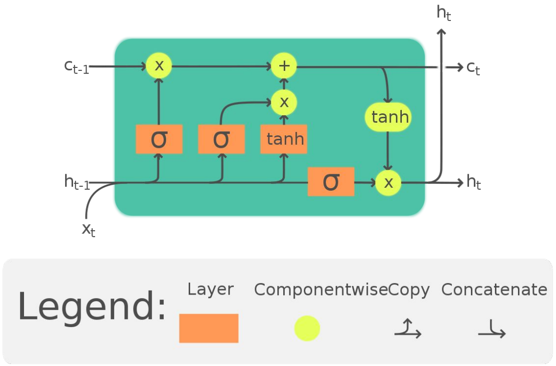
Long or Short Term Memory Recurrent Neural Network (LSTM-RNN)

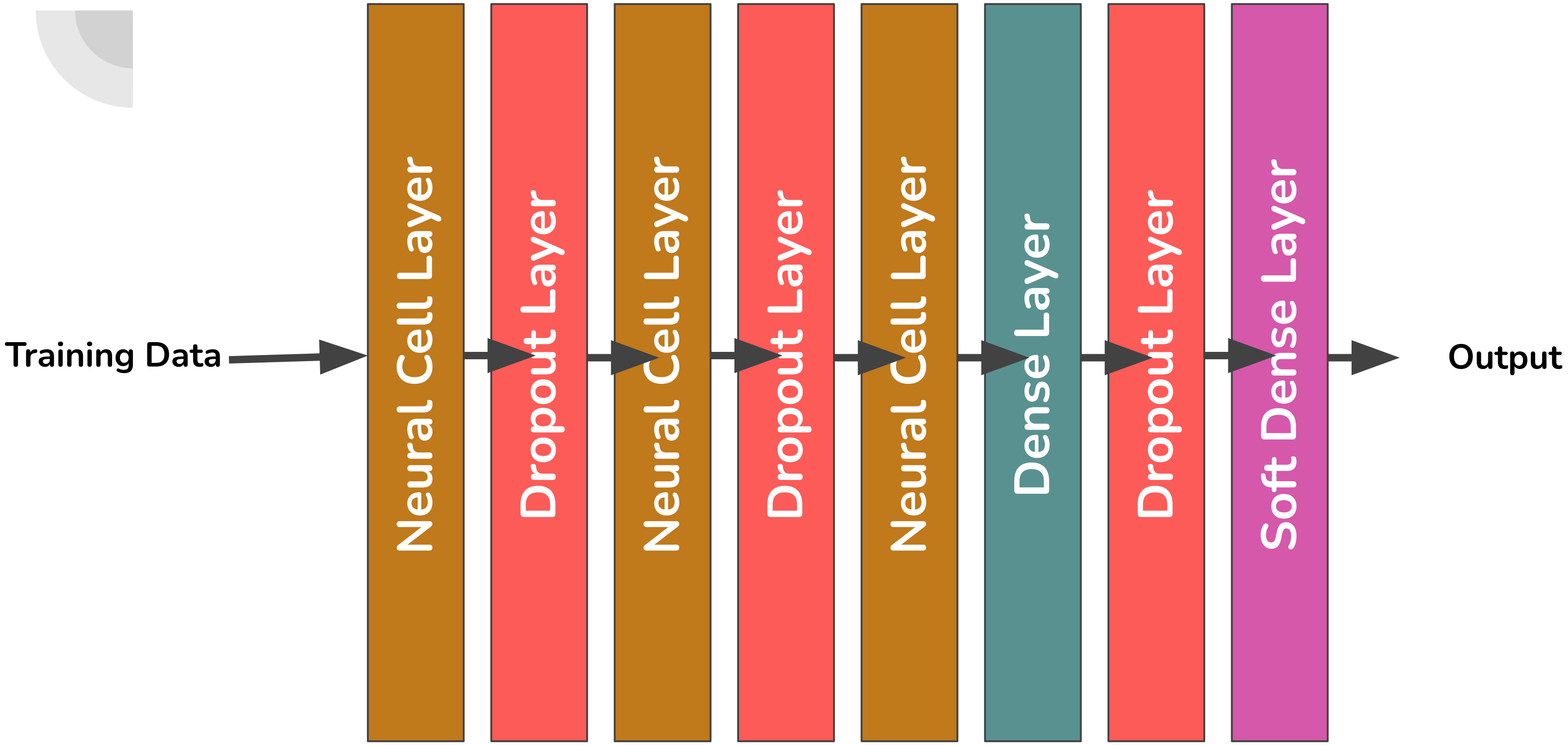
Unsupervised Machine Learning Model

Data points are fed into a recurring loop back into the model

Useful in cases where learnt rules need to be stored in our model

Well-suited to processing and making prediction to our time based data



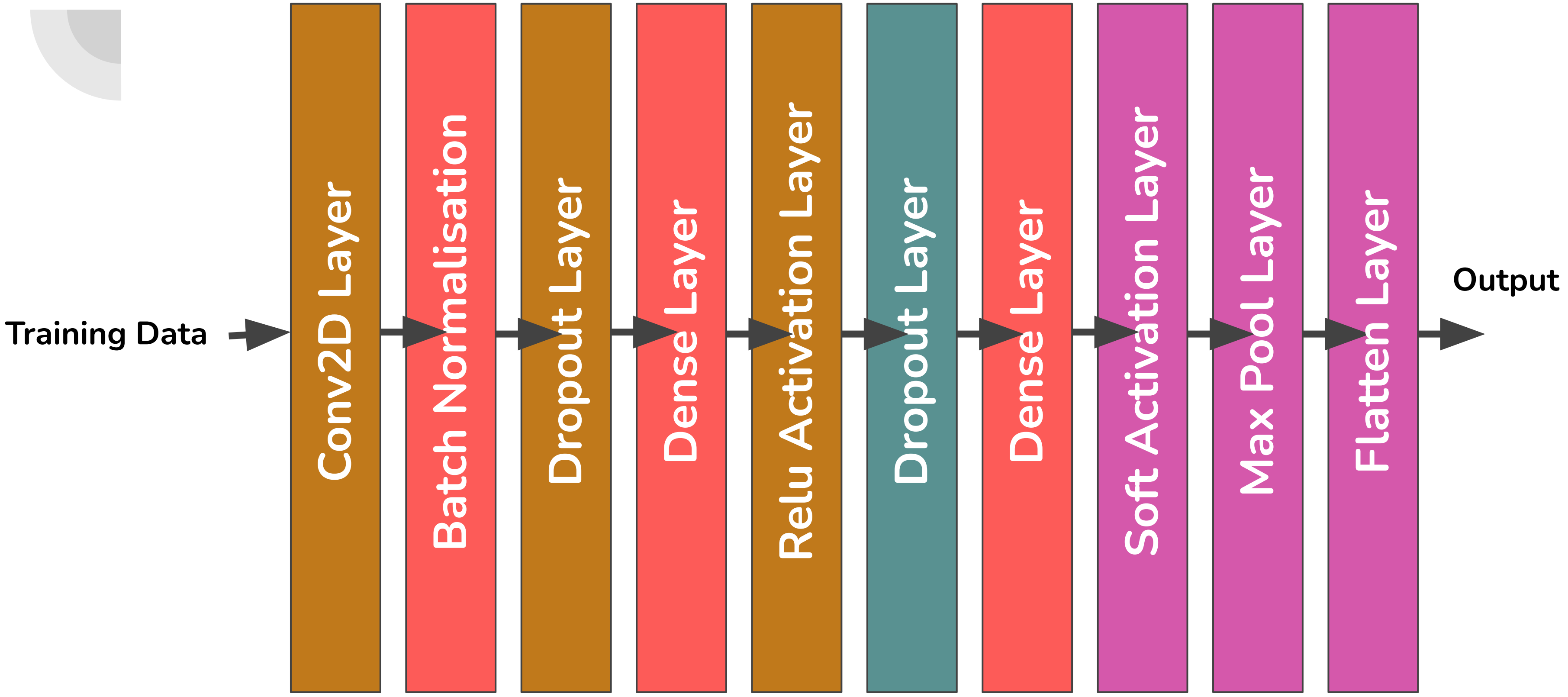

Training Data
Neural Cell Layer
Dropout Layer
Neural Cell Layer
Dropout Layer
Neural Cell Layer
Dense Layer
Dropout Layer
Soft Dense Layer
Output

Convolutional Neural Network (CNN)

Useful for classification of images

May be able to learn the patterns of the training music

One-hot encoded data may be used for CNN to create convolution matrices


Training Data
Conv2D Layer
Batch Normalisation
Dropout Layer
Dense Layer
Relu Activation Layer
Dropout Layer
Dense Layer
Soft Activation Layer
Max Pool Layer
Flatten Layer
Output

Evaluation Metrics

Accuracy - Comparison of output data with our actual music

Loss - Cross entropy loss from our output

Musical Evaluation by ear

Caveats

Overfitting

- Danger of regurgitating training material instead of generating original music based on the training data.
- Want it to follow the data generally for musical rules and stylistic approach

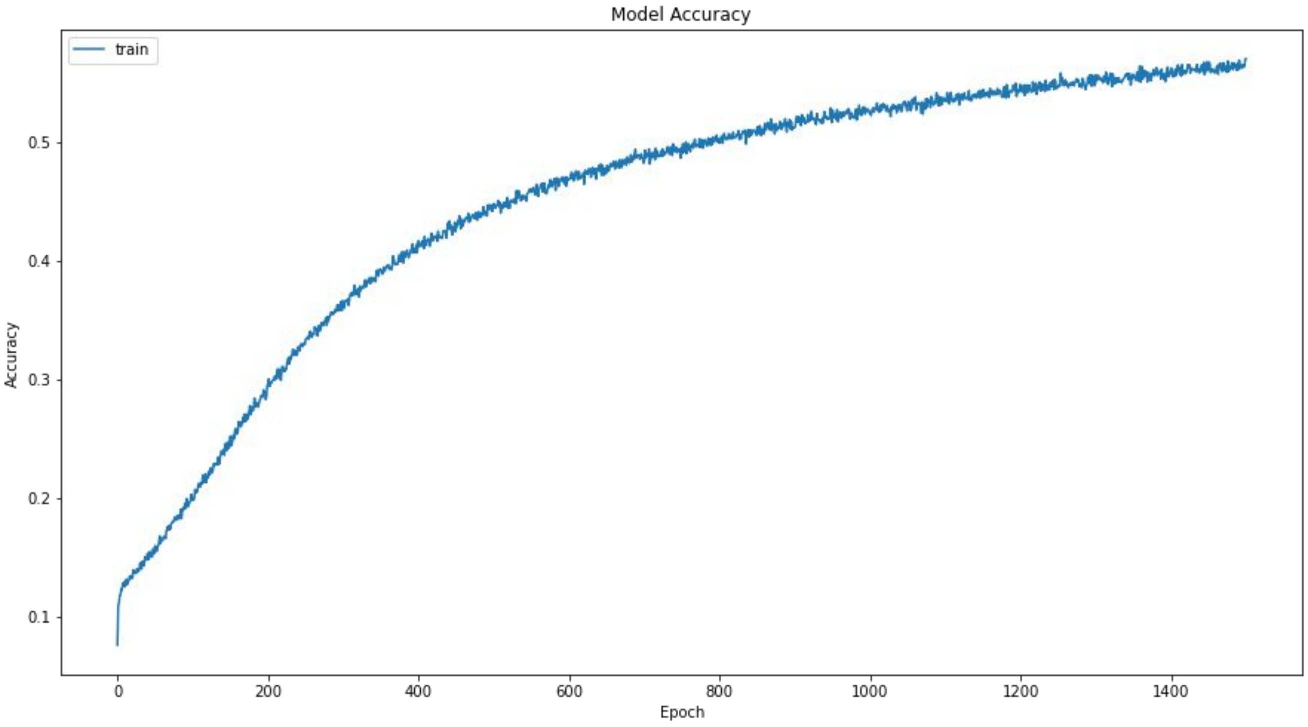
Underfitting

- Too much variability in data
- May result in very funky or erroneous sounding notes

LSTM Model Accuracy-Epoch Graph

Increasing Accuracy as training epochs increases

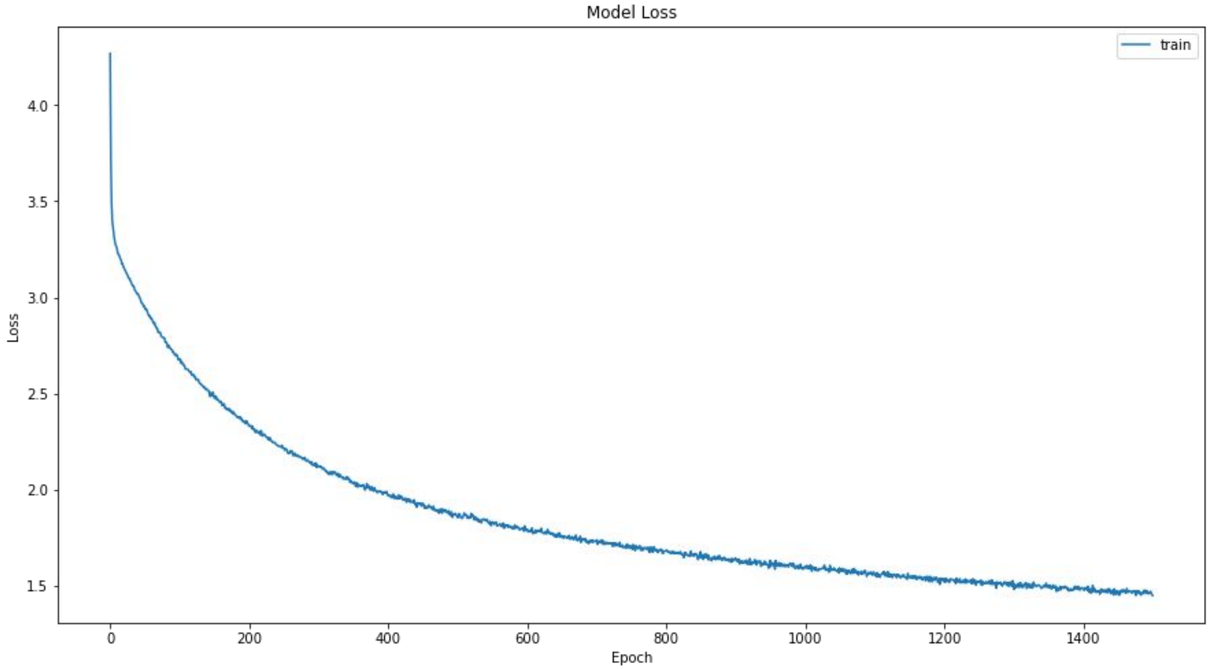
Indicative that musical rules may be assimilated into the model



LSTM Model Loss-Epoch Graph

Lower loss as training epochs increases

May indicate better training for generated music



CNN Model Accuracy-Epoch Graph

Similar trend with
LSTM

CNN Model Loss-Epoch Graph

Similar trend with
LSTM

Generate our Music

Generate our music by sending a random one-hot encoded array

Parse the output music into a synthesizer module

Evaluate our music based on our ears

LSTM-RNN Generated Music

100 Epoch

1000 Epoch

1000 Epoch

CNN Generated Music

20 Epoch

200 Epoch

300 Epoch

Future Improvements

Lack of GPU acceleration (Ran mostly on CPU due to Tensorflow issues)

Velocity Data discarded

Pre-train models on common harmonic patterns and rules

Possible Explorations

MusicGAN (Generatively Adversarial Network) may yield better results

Instead of using CNN (2D), we can apply CNN LSTM

High computational power required thus not explored

Conclusion

LSTM-RNN is a suitable model for generating music

Long-Short Term Memory is crucial in learning music

Balance of computational power vs output

Q & A

Thank you!

General Assembly DSI 25

Capstone Project