Technical Documentation

There are four models split into three different folders Classification, Interval & Duration and Regression. They share functions which provide similar use – they are not copies of each other and do something unique each. In this documentation, I will briefly explain the general purpose of each file but note that they all slightly vary.

Main.py

This is the runnable file to build the model. It finds the files listen in the /tabs/ directory which is the input for the network and iterates through them and extracts the relevant data into a list. This is done through the convertToText.py. It then has to save original data e.g. the notes for later use in the validation because it needs a note-to-int dictionary and vice versa.

convertToText.py

This uses stacked for loops to get the individual data using pyGuitarPro. There is a keyword list which is checked to make sure the song's track does not contain that word first. The valueToNote function can return a note depending on the MIDI note number. It does this by modulus 12 and then the remainder dictates the note. For example 55 % 12 = 7, and then using the valueToNote function it would return that it is a G. This has varied use throughout the utils function but does not provide any critical information required for the program.

Rnntwo.py (and other RNN)

This starts by sorting the pitch names for use by the dictionary to minimise the input variety. It is then transformed into rolling list of the sequence length described in listLength. This is then normalised for more effective calculation and send through a sequential LSTM model. The metrics are then plotted using matplotlib.

Transpose.py

This split into three different functions. Firstly, the tranposeBy function receives the song.key.name from convertToText and then returns how many semitones difference is required to get the key signature of the original song to the key of C Major – which is the shared input key across all songs.

Process is called in convertToText which contains the semitone value that the note needs to be added to convert it to the key of C Major.

convertToC is not used but I chose to keep it in because it may be an alternate way to convert all the songs to C, this may prove useful for further validation.

Utils.py

Utils allows the user to create full passages of song of any length. The length can be changed in the for loop. It reads the input of a song and then creates the rest of the song for you. This shows a practical use for the application further than just simply validation. It follows the same preprocessing rules as the input for the network and then predicts based off the saved model. This is then saved to a MIDI files and can be accessed in the main directory as output.mid.

Validate.py

This is used to validate the model against a series of test - 'sameNotes.gp5', 'CScale.gp5', 'outOfScaleSameNotes.gp5', 'AB.gp5', 'ABC.gp5', 'ABCD.gp5', 'pentatonic.gp5', 'inputtedSong.gp4', 'newSong.gp5'. It loads the model and predicts the output each time, when the actual output is also stored so it can be compared and analysed to against the predictions. Then matplotlib for each test will display a graph with the actual and prediction lines.

Validate – New Songs.py

This validation file is purely for new songs. It allows the user to input Guitar Pro files into /validation/newSongs and then it the new songs will be predicted. This will also display a graph.

Output.mid

This is the output MIDI file which is the outcome from using the utils.py functions. This can be viewed using a sequencer which can also be found online.

/input/

For use by utils which the user can input the start of a song and get the network to predict the rest.

/models/

This is where the model is saved alongside any other required files such as the scaler and note lists.

/tabs/

This is the input for the network build, all files must be Guitar Pro. If the key of a song is not detected then the file will be automatically deleted from the folder.

/validate/

Validation files used by validate.py and validate – new songs.py