Generating Music Using Deep Learning Neural Networks

Low Zhe Wei
DSIF 3

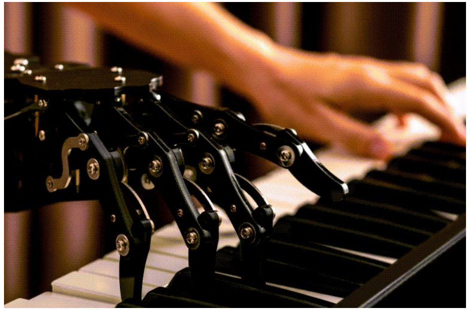
Table of Contents

Problem Statement

It can be difficult at times for musicians to come up with melodies and their accompanying chords. This can often lead to undue frustrations and stress on the musician, producer and record labels, as well as incurring monetary costs due to the additional time spent in the recording studio or delays in song releases.

In view of this, I plan to develop and model that can generate melodies with a random or manual input.

With this model that is able to generate its own music, I hope to save costs in the music industry by reducing the time taken to write catchy songs.



Reading the Dataset

Songs in MIDI format

Use Music21 Library to read the MIDI files

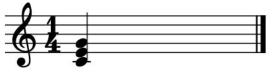
```
for file in glob.glob("../midi_songs/*.mid"):
```

```
notes_to_parse = midi.flat.notes
```

Notes
note.Note(‘note’)

Chord
chord.Chord(‘chord’)

```
c1 = chord.Chord(['C', 'E', 'G'])
c1.show()
```



```
c1.normalOrder
```

```
[0, 4, 7]
```

Durations
quarterlength

Exploratory Data Analysis

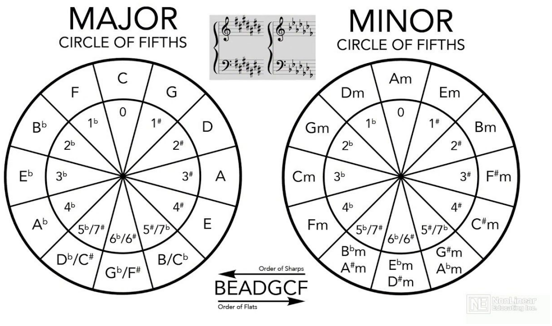
	Train Dataset	Test Dataset
Source	GitHub	Kaggle
Genre	JRPG	Anime
No. of songs	92	130
No. of elements	62,367	97,424
Average song length	677.9	749.42
Unique notes/chords	382	437
Unique durations	173	70

	Train Dataset		Test Dataset	
Top 5 notes/chords	notes/chords		notes/chords	
	A2	1592	D3	1912
	D3	1376	9	1886
	A3	1351	2	1829
	E4	1350	A2	1826
	G4	1341	7	1811
Top 5 durations	durations		durations	
	0.5	25729	0.25	36024
	0.25	9640	0.5	32512
	0.75	7990	1.0	10751
	1.0	4190	1/3	4860
	1/3	3541	0.75	3307

Dividing Notes/Chords & Durations

A song can be split into 2 basic elements:

Notes/Chords - this determines which notes usually go together



Durations - this determines the beat of the song or sections of the song

We will construct a 2 models! 1 for Notes/Chords, 1 for durations!

Preprocessing

Separate the **notes/chords** and **durations** of individual songs.

Map each note/chord and duration(key) to an integer(value).

{‘C4’:0,‘C5’:1…}

{‘0.25’:0,‘0.5’:1…}

Add unknown variable as ‘**unkw**’ to the integer value of 0.

{‘unkw’:0,‘C4’:1…}

{‘unkw’:0,‘0.25’:1…}

Convert the notes/chords and durations sequences into integers.

<u>Input:</u>

Divide into lists of 100 elements each, where each new list contains the 1st - 101st element relative to previous list.

<u>Output:</u>

Each output will be the 101st element of the 100 note input

<u>Input:</u>

[0,1,2…100]

[1,2,3…101]

<u>Output:</u>

[101]

[102]

Model Architecture

Embedding

By passing the input through an Embedding layer, the model can learn which notes/chords or durations are closely related to each other, which will ultimately help with predictions.

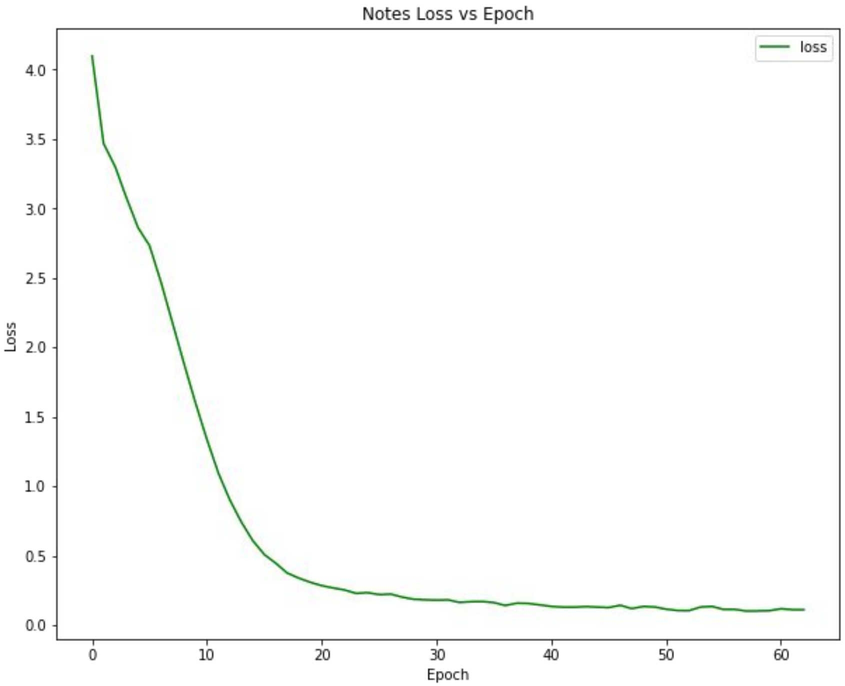
Bidirectional LSTM

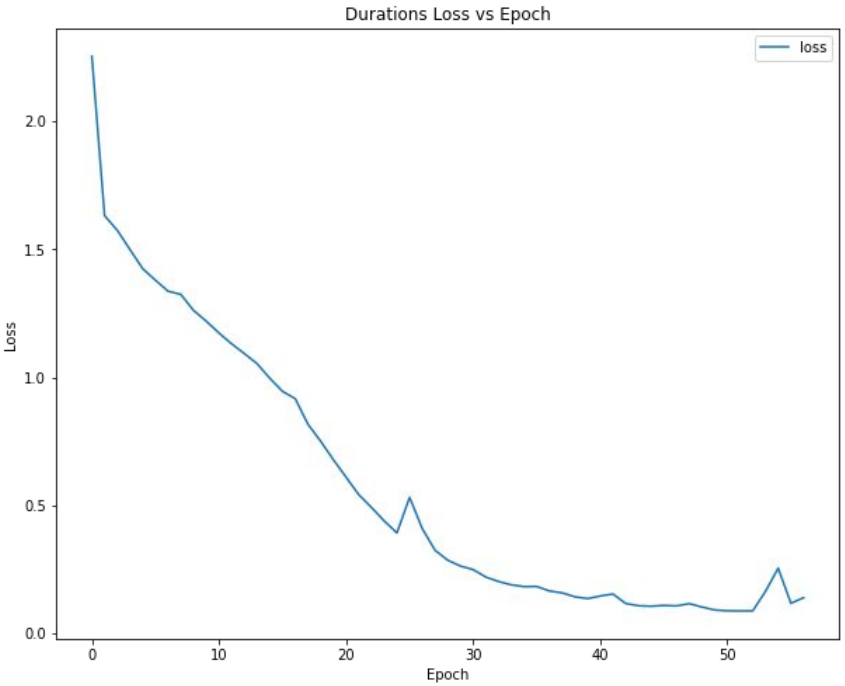
Bidirectional LSTM is the process of making any neural network have the sequence information in both directions backwards (future to past) or forward(past to future).

Dense

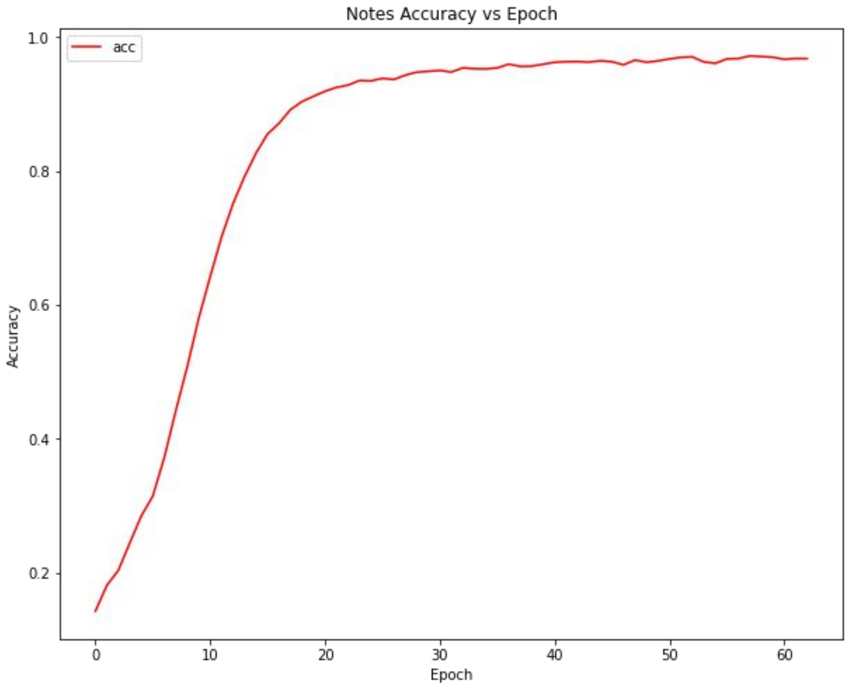
Includes activation functions like ReLU (hidden layer) and Softmax(output).

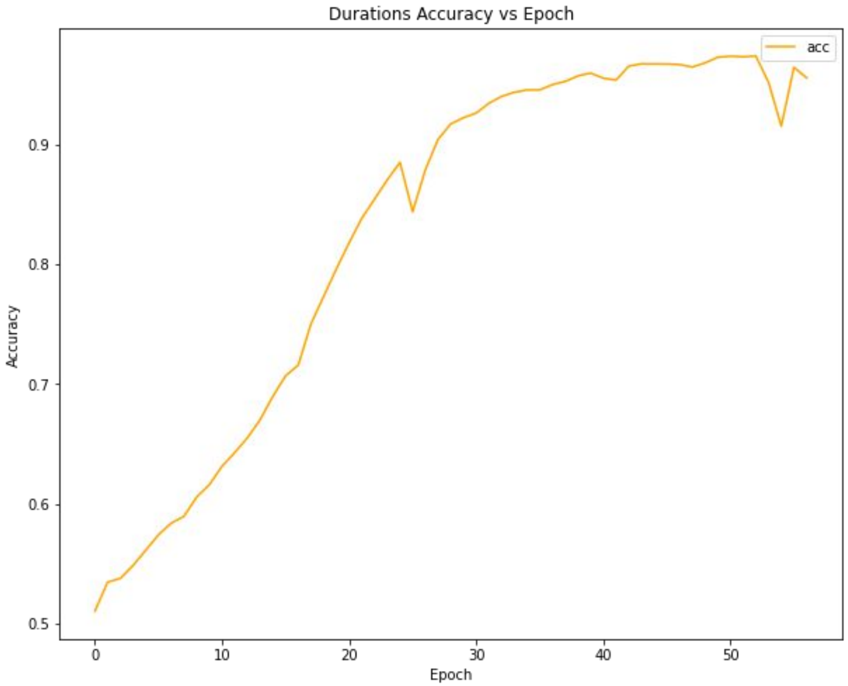
Model Loss vs Epoch





Model Loss vs Accuracy



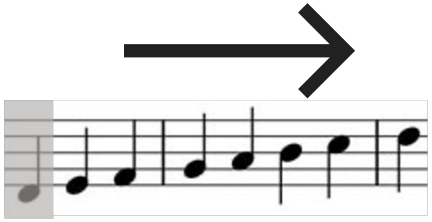


Getting Predictions

<u>Method 1 - Generation with insertion</u>

- Generate a note/chord or duration.
- **Insert this generated note back into the input sequence.**
- Predict the next note.
- Repeat the process as many times as you want.

Works best for notes/chords!



<u>Method 2 - Generation with shifting</u>

- Generate a note/chord or duration.
- **Shift the input sequence one note up.**
- Predict the next note.
- Repeat the process as many times as you want.

Works best for durations!

Constructing the Song

---

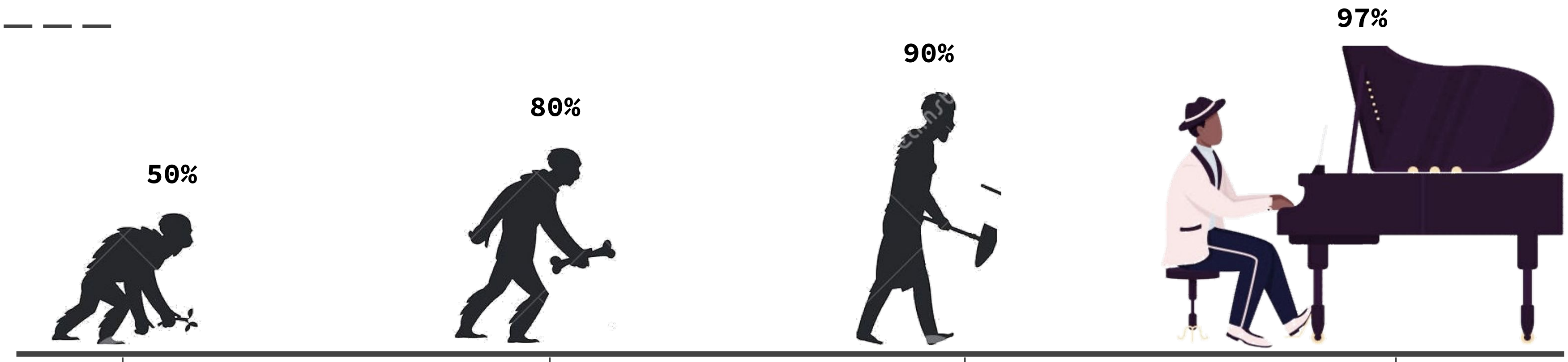
1. Convert integers back to notes/chords or durations
2. Melody generation

```
n1 = note.Note('C', quarterLength=1)
n2 = note.Note('E', quarterLength=.5)
n3 = note.Note('G', quarterLength=.5)
n4 = note.Note('C', quarterLength=2)

s = stream.Stream()
s.append([n1, n2, n3, n4])

s.show()
```

Music Generation by Accuracy



https://soundcloud.com/zhe-wei-3/50a?utm_source=clipboard&utm_medium=text&utm_campaign=social_sharing

https://soundcloud.com/zhe-wei-3/80a?utm_source=clipboard&utm_medium=text&utm_campaign=social_sharing

https://soundcloud.com/zhe-wei-3/90a?utm_source=clipboard&utm_medium=text&utm_campaign=social_sharing

https://soundcloud.com/zhe-wei-3/97a?utm_source=clipboard&utm_medium=text&utm_campaign=social_sharing

Did it learn Music Theory?

‘A2’ Note

	Cosine Similarity
'A', 'B', 'C#'	0.400902
'G', 'Bb', 'C#', 'Eb'	0.39942
'B'	0.369358

'A', 'B', 'C#' make up the Aadd(2) chord, which is a variation of the A chord. Furthermore, all 3 notes are in the key of A major.

'G', 'Bb', 'C#', 'Eb' make up the Eb7/G. This combination of notes is rarely seen in western music, but is more common in eastern music, and is a likely blend of the D harmonic minor and A harmonic minor, both of which feature the A note.

'B' is a prominent note in both the key of A major and A minor.

Future Improvements

Multiple Parts

More Instrument Types

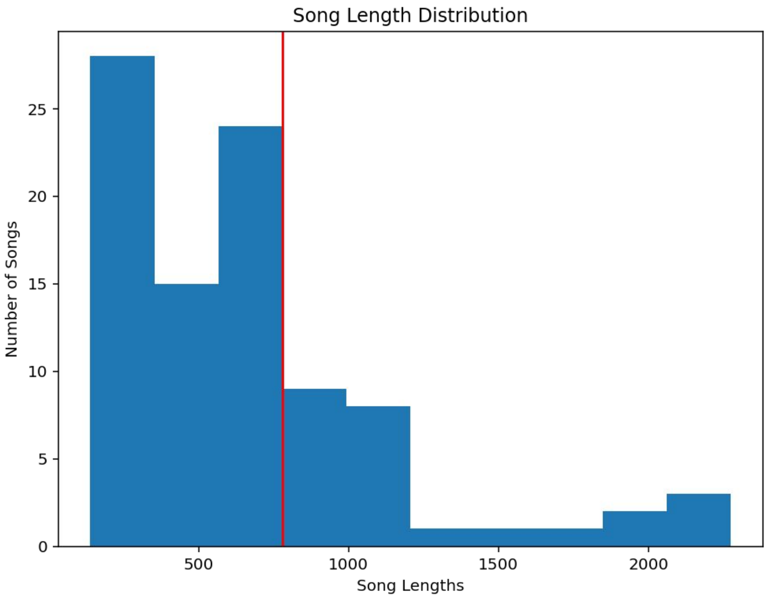
Use of Transformers

Seq2Seq Architecture for chord input and melody output

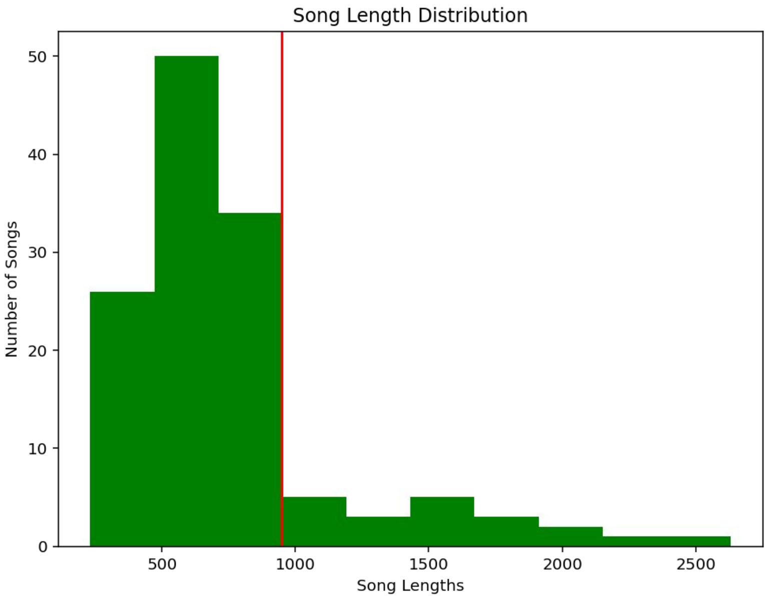
Appendix

Song Length Distributions

Train



Test



Full Model

```
def create_network(network_input, n_vocab):
#     create the structure of the neural network
    model = Sequential()
    model.add(Embedding(
        n_vocab,
        512,
        input_length=100,
    ))
    model.add(Bidirectional(LSTM(
        512,
        recurrent_dropout=0,
        return_sequences=True
    )))
    model.add(Bidirectional(LSTM(512, return_sequences=True, recurrent_dropout=0,)))
    model.add(Bidirectional(LSTM(512)))
    model.add(BatchNorm())
    model.add(Dropout(0.3))
    model.add(Dense(256))
    model.add(Activation('relu'))
    model.add(BatchNorm())
    model.add(Dropout(0.3))
    model.add(Dense(n_vocab))
    model.add(Activation('softmax'))
    opt = Adam(learning_rate=0.001)
    model.compile(loss='categorical_crossentropy', optimizer=opt, metrics=['acc'])

    return model
```

My SoundCloud

https://soundcloud.com/zhe-wei-3?utm_source=clipboard&utm_medium=text&utm_campaign=social_sharing