Theodore Glavas
Wassim Jabbour

MAIS 202 - Deliverable 2

1- The goal of the project is to make a model that can generate new classical music based on samples we pass it

2- We are using the same dataset from deliverable 1, and are using Chopin and Beethoven midi files on kaggle. The Kaggle dataset had an example data preprocessing template which used the Music21 library to extract only the pitch of notes and chords. Starting from that base code, we expanded it to include pauses, and different note durations to introduce rhythm to the training set. Additionally we used a MIDI edit software, MuseScore, to isolate a single track from each piece (the right hand piano track). This allowed the machine learning algorithm to focus on a single musical pattern at a time.
After removing infrequently used notes, our final dataset contains 31490 notes/rest/chords, with 752 unique combinations of pitch and duration.

3- We are using a neural network with LSTM layers to predict the next note in a given sequence.

A. We used Keras to implement our long short term neural network with dropout layers to avoid overfitting. The current architecture we are using is based on the dataset's example code:

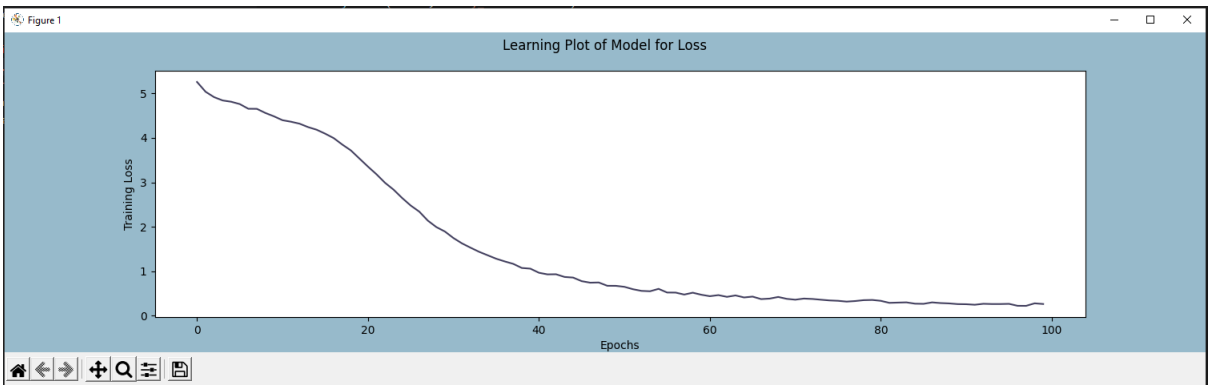
Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 40, 1024)	4202496
dropout (Dropout)	(None, 40, 1024)	0
lstm_1 (LSTM)	(None, 512)	3147776
dense (Dense)	(None, 512)	262656
dropout_1 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 752)	385776

B. We used Adamax for the learning curve (Learning rate hyperparameter), with an 80-20 train-test split.
C. We tested our model by generating sequences of 400 notes based on a random seed and listened to the output.
D. We faced many challenges since neural networks were not covered in class before the deadline. However, the hardest part by far was preprocessing the data because a bad dataset leads to bad results.

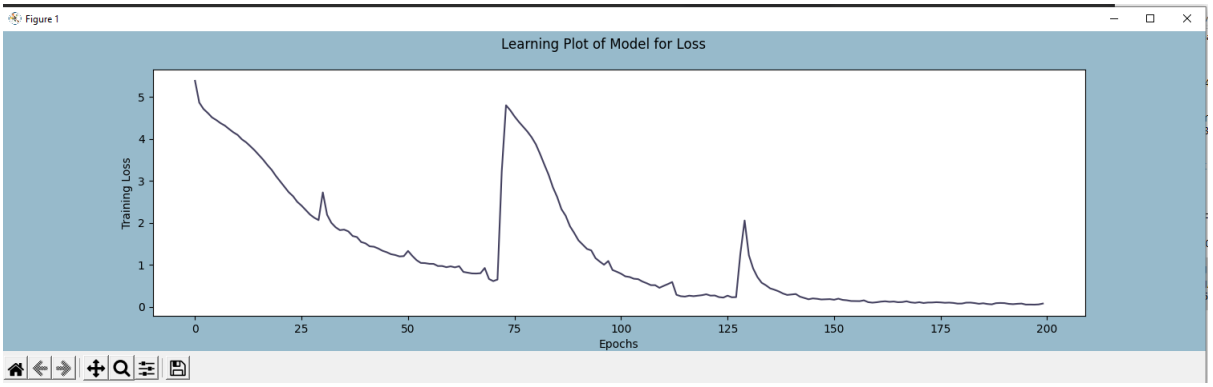
4- We used the cross-entropy loss function to train the model

The first model was half the size of the architecture described in part A and was trained for 100 epochs. It arrived at a loss of 0.2662.

Theodore Glavas
Wassim Jabbour



The second model with the architecture described in part A had a few abnormalities in training. It eventually reached a loss of 0.0480 after 200 epochs.



We suspect that the second model is somewhat overfitted, because a few segments of the output are very similar to some of the input pieces. On the bright side, this at least assures us that our model is indeed working!

To listen to some of the generated pieces, go to
https://github.com/WassimJabz/MelodifAI/tree/main/Music%20Generated
And download the MIDI files. Windows Media player can play these files automatically.

The preliminary results are mixed. There is definitely some form of rhythm and musicality in some areas, but most of the sections still seem random.

5- We will be trying to improve our preprocessing of the data while also tuning the parameters to get better results. Specifically, we will try to fix weird bugs that completely distort a lot of our dataset files (we had to manually remove those instances), and experiment with different architectures and epoch counts.