GrooveNet: AI for Music Generation

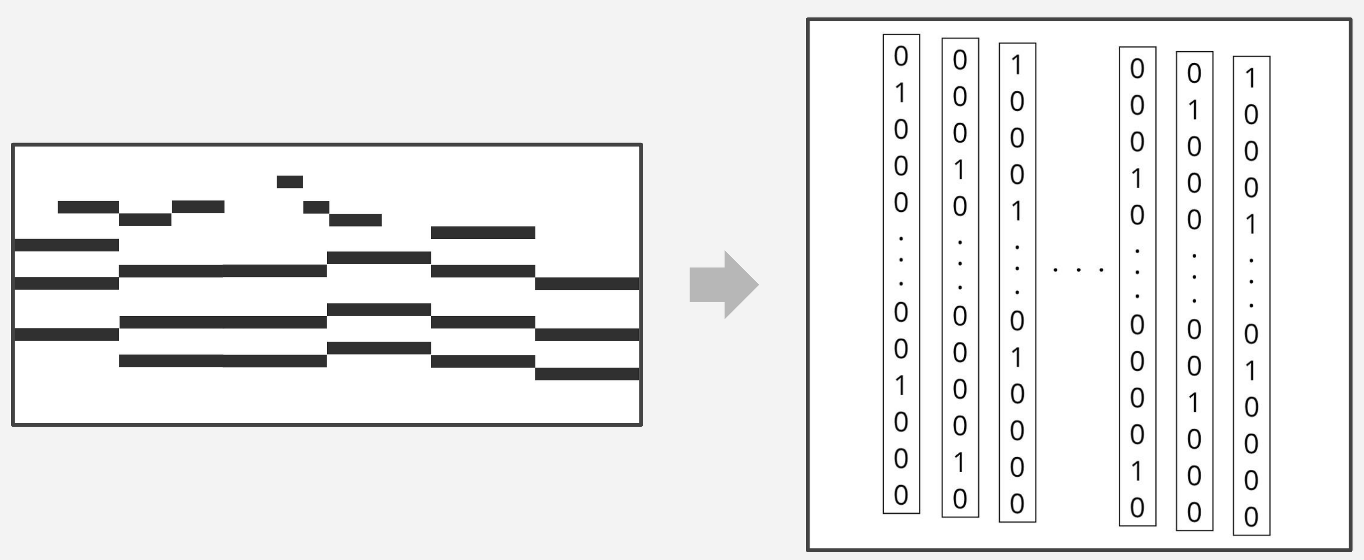
Luciano Gonzalez and Dhara Yu

Overview

For our project, we are exploring the use of different AI techniques to create novel classical piano pieces. A particular challenge is the inherent artistry required of composing music - how can we capture musical concepts like rhythm and reposition?

Data and Features

- Trained on corpus of Beethoven piano compositions in MIDI file format
- Processed each file into a sequence of discretized musical moments, where each moment represents which notes are being played
- Discretized into 64th note segments
- Moments embedded as vectors of length 127



Baseline

Markov Chain

- States are represented as a binary array, indicating which notes are played at the given moment
- State **A** will transition to state **B** with the same probability that state **B** appears after state **A** in the training data
- Music is generated through a random-walk method, where a moment of music is generated by a weighted sample of the previous moment's possible transitions and their probabilities

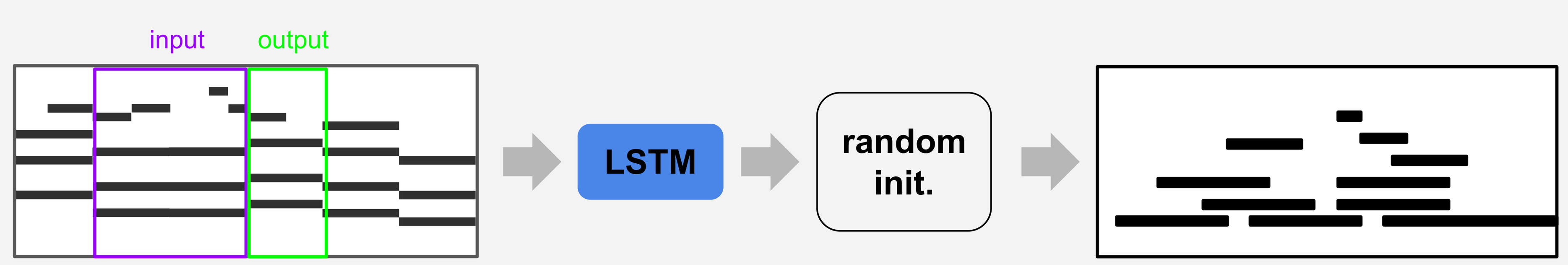
Models

Logistic Regression

- Logistic regression performed for each note, generating a probability for each note being played
- As features we use the binary array of note indicators, along with rhythm features
- Trained to minimize absolute loss
- To generate sequence:
  - Initialize with an empty moment of music
  - Use previous moment of music + rhythm features to generate probability of notes being played in next moment
  - Use conditional probabilities to choose the number of notes to play k_i based off the number of notes played in the previous moment k_{i-1}
  - Perform a weighted sample for k_i notes to select which notes to play in the given moment.

LSTM network

- Implemented in Keras with Tensorflow backend
- 2-layer LSTM, batch size of 128, for 100 epochs
- Cross entropy loss for multi-class classification
- Trained on input sequence of length *n* = 1, 3
- To generate sequence:
  - Initialize with random sequence of length *n*
  - For each moment, feed most recent sequence of length *n* into model, which returns probabilities of each note being played
  - Choose the *k* notes with the highest probability to be played in that moment, where *k* is calculated based on distributions from training data



Evaluation

- Survey participants (*n* = 29) listened to 30 second music sample and assessed for belief that the sample was produced by a human composer
- 1-5 scale, where 1 represents certainly machine-generated and 5 represents certainly human generated

	Beethoven composition	Markov model	Logistic regression	LSTM network
Mean	4.55	2.98	2.03	2.50
Std. Dev.	0.92	0.68	1.14	1.14
Mode	5	3	2	2

Discussion

- Overall, moderately effective in capturing the structure of chords and producing polyphonic notes
- But failed to output pieces with long-term structure, rhythm or repetition
- Markov model output was considered the best of the generative models

Next Steps

- Larger and more varied dataset
- Deeper LSTM network
- Handling for notes not seen in dataset
- More sophisticated embedding to capture long-term dependencies and structure

Acknowledgements: We used piano-midi.de for training data. Thanks to Sigorur Skuli for his article on LSTM for music generation, Roger Dannenburg for his course notes from Intro to Computer Music, and Andries Van Der Merwe for his article on Markov models.