Music Generation Using Deep Learning

By: Marwan Harajli

Outline

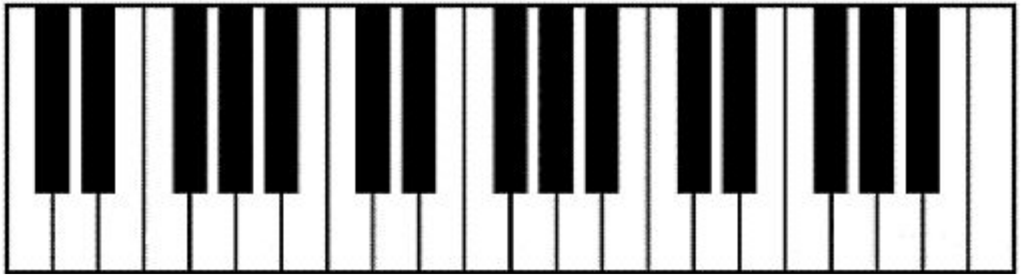
1. Introduction: What are we trying to do?
2. What is music?
3. The Data: Midi files, Classical Archives, and Music21
4. Exploratory Data Analysis
5. Encoding Music: Music as a time series
6. Building and Training an LSTM Neural Net
7. Results and Discussion

Introduction: What are we trying to do

- Can a machine learn to generate music given a data set of “good” music.
- Can we use deep learning methods to do the above (LSTMs, GANs, etc.)

What is Music?

- Sound Frequencies/Notes
- Scales
- Tempo
- Pronunciation
- Chords

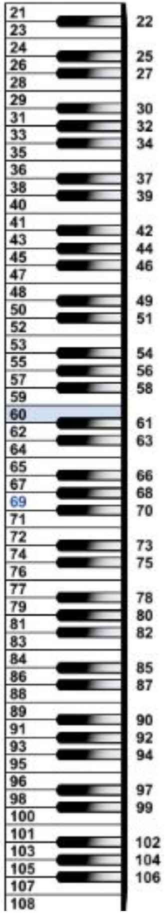


The Data

- The Source: http://www.piano-midi.de/
- 266 Midi files, separated by artist (17 Artists).
- All files are classical piano pieces (piano solos).
- For this presentation we only use the 15 songs by Mendelssohn.

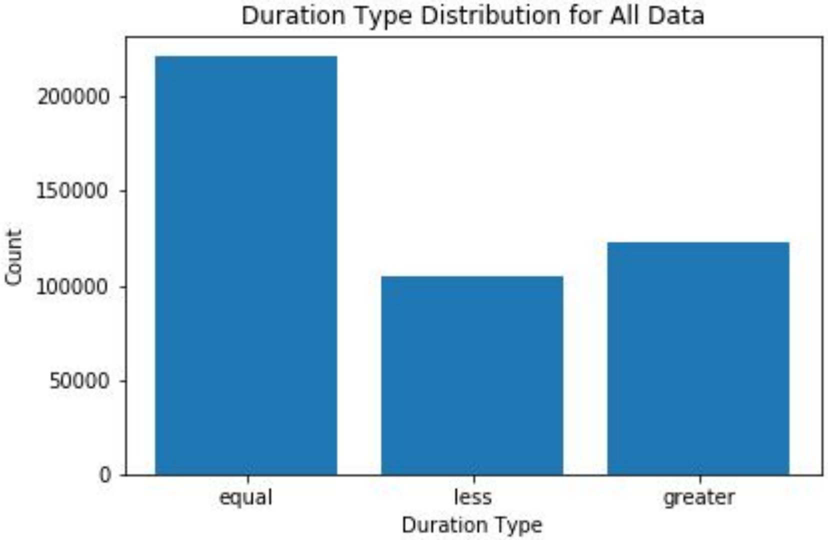
The Data

- Music21: a Toolkit for Computer-Aided Musicology.
- This library contains methods that takes a midi file as input and parses it. It returns a list of Music21 elements. These Elements include:
  - Note Events
  - Tempo Events
  - Instrument Change Events
- Each note events has the following attributes:
  - Pitch: a number representing which key on the piano
  - Offset: The time of the event relative to the start of the midi stream. (In quarter lengths)
  - Duration: how long to play the note (in quarter note lengths)
  - Velocity: how hard the key was hit



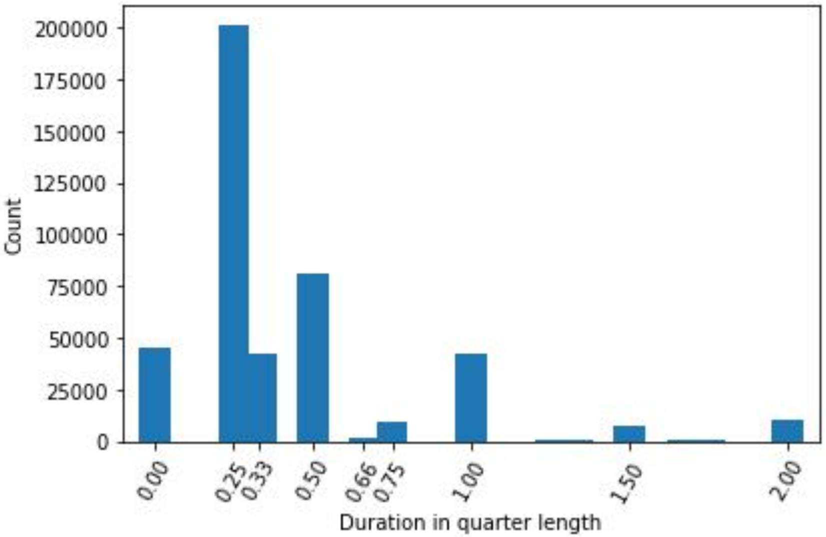
Exploratory Data Analysis

Do most notes have durations equal to the difference between the offset of the next note event minus its offset?



Exploratory Data Analysis

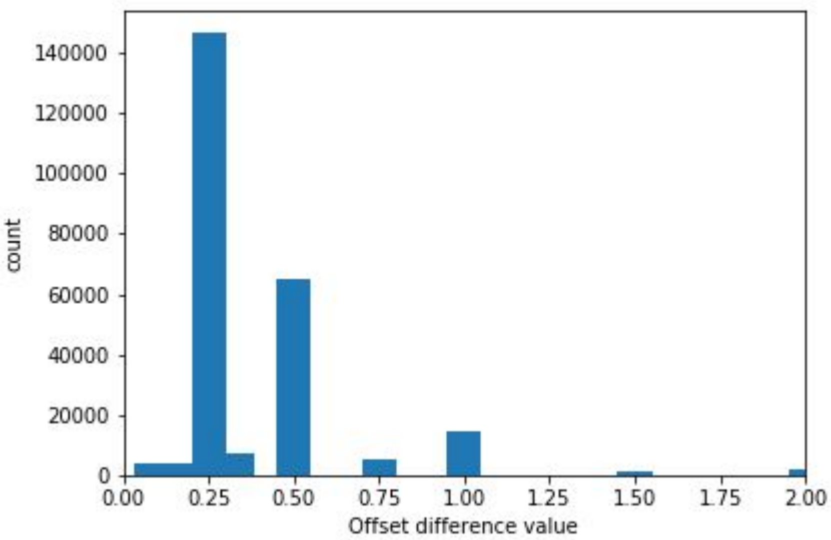
What is the distribution for the duration of note event?



We see that a quarter of a quarter note (called a sixteenth note) is the most common note duration.

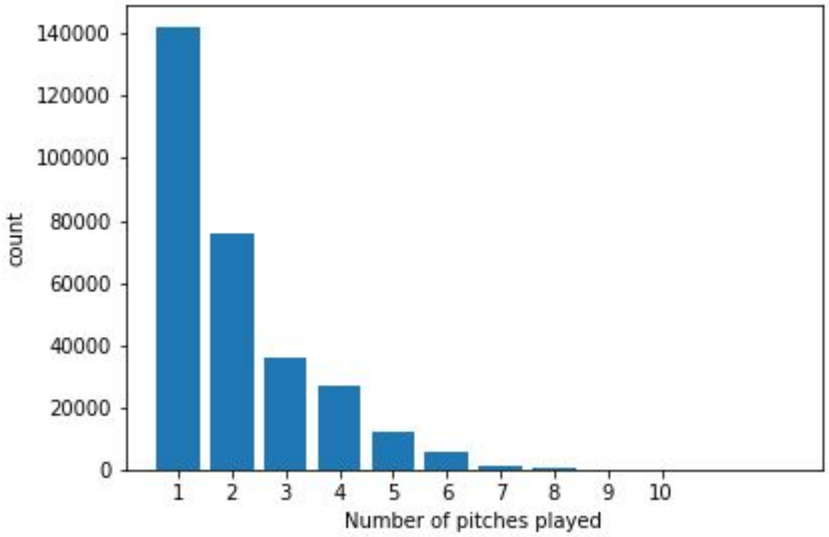
Exploratory Data Analysis

What is the distribution of Offset Differences?



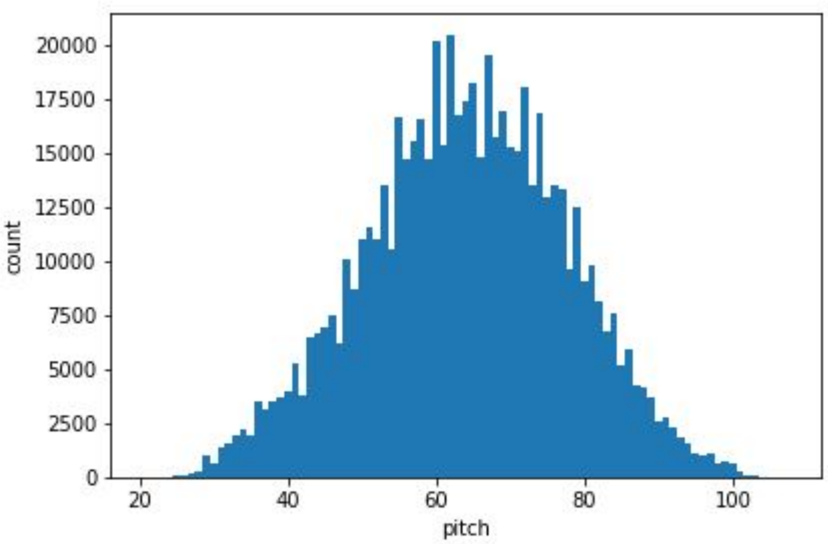
Exploratory Data Analysis

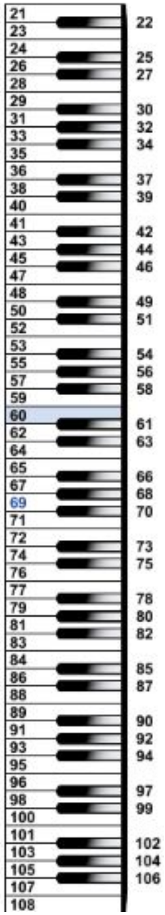
Knowing humans have only 10 fingers, what is the distribution of number of notes played at a given offset?



Exploratory Data Analysis

What is the distribution of played pitches?





Encoding Music:

- We assume offset differences and note durations are the same
- A piano piece becomes a sequence of note events and goforward (gf) events.
- Note events are represented by numbers representing the pitch
- gf events are represented as gf+(time jump in quarter notes): example gf0.25
- Fur elise thus becomes as shown below:
- Discussion: Uniqueness of representation

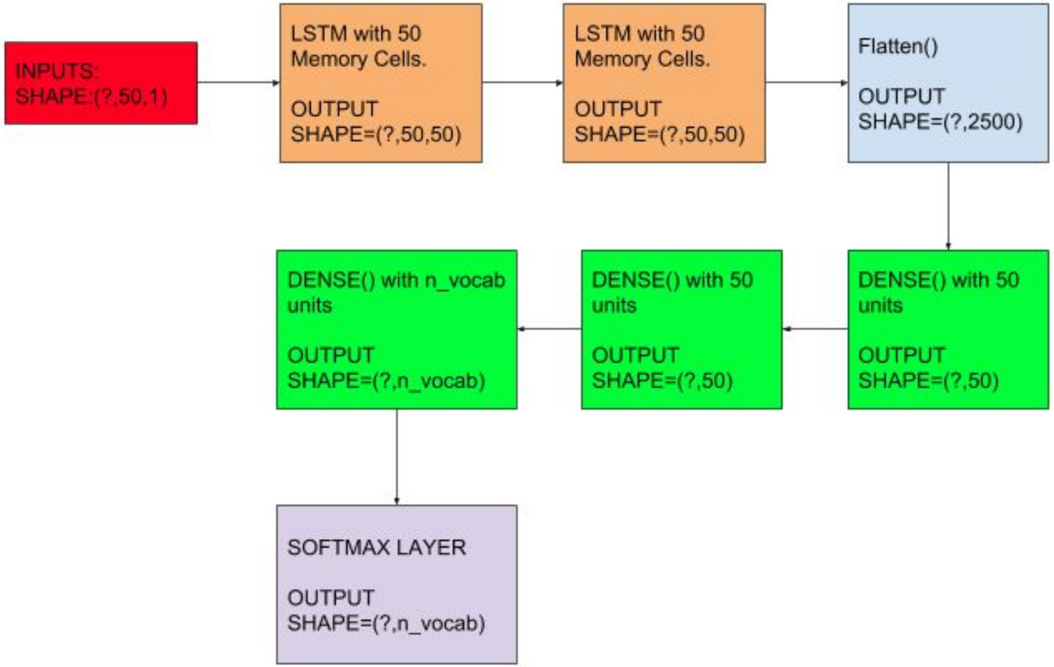
[‘76’, 'gf0.25', ’75’, 'gf0.25', ‘76’, 'gf0.25', ‘75’, 'gf0.25', ‘76’, 'gf0.25', ‘71’, 'gf0.25', ‘74’, 'gf0.25', ‘72’, 'gf0.25', ‘45’, ‘69’, gf0.25’, ‘52’, ...]

Building and Training an LSTM Neural Net

- Given a sequence of 50 note and gf events, we train a model to predict the 51 event. (Discussion how to decide input sequence length?)
- Given this trained model, we then feed it 50 events, and let it generate a 51st and so on till we have a sequence of desired length.

Building and Training an LSTM Neural Net

Neural Network Architecture:



INPUTS:
SHAPE:(?,50,1)

LSTM with 50 Memory Cells.

OUTPUT SHAPE=(?,50,50)

LSTM with 50 Memory Cells.

OUTPUT SHAPE=(?,50,50)

Flatten()

OUTPUT SHAPE=(?,2500)

DENSE() with 50 units

OUTPUT SHAPE=(?,50)

DENSE() with 50 units

OUTPUT SHAPE=(?,50)

DENSE() with n_vocab units

OUTPUT SHAPE=(?,n_vocab)

SOFTMAX LAYER

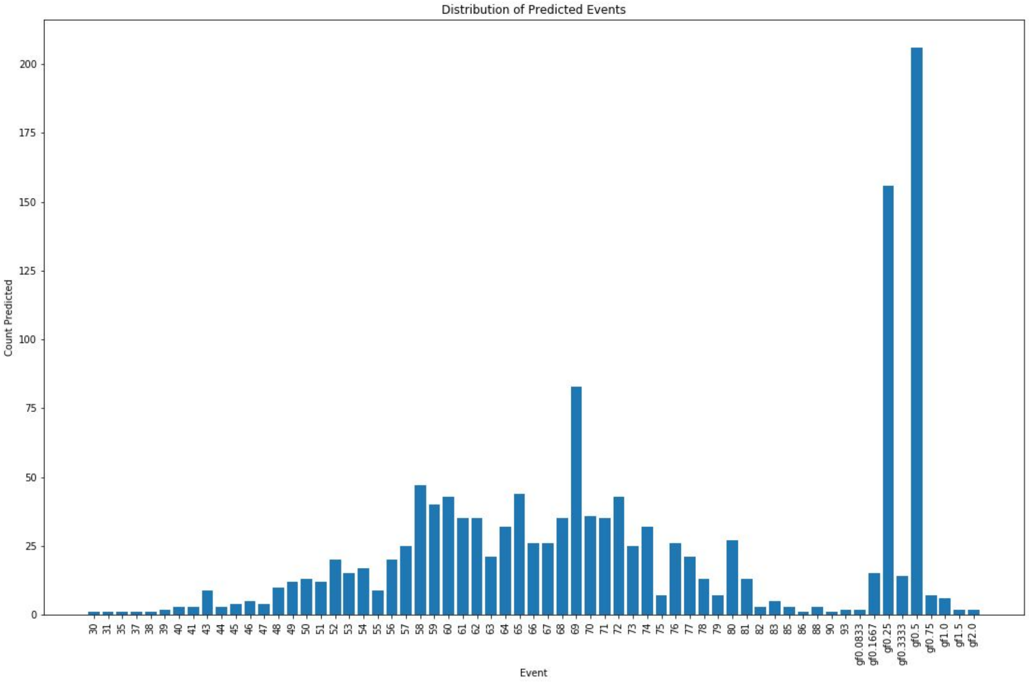
OUTPUT SHAPE=(?,n_vocab)

Building and Training an LSTM Neural Net

- The Mendelssohn midi files translate to 30,000 sequences.
- We use CuDNN LSTM implementation on AWS p2.xlarge
- Trained for 500 epochs with a batch size of 1000
- Training takes 12.5 minutes, loss=0.0788; accuracy=98%
- Accuracy on held out set= 16.25%

Building and Training an LSTM Neural Net

On testing data, what do the prediction look like? (Discussion)



Generating Music

- Music Generated from training data vs. testing data
- See TrainingDataSample23 vs. TestingDataSample9 vs. TestingDataSample10

Future Work

- Building a metric to compare two sequences to each other.
- Evaluate if the generated pieces are simply replays of snippets from the training data (is the model just memorizing/overfitting).
- GAN? Maybe with midi as an image? And maybe ConvNets rather than LSTM?