Music Composition Using Classification Algorithms

Peter Wang (pwang01@stanford.edu)

Predicting

- Want to compose music by using classification algorithms on past observed note values
- This results in a multiclass classification problem used for prediction
- Each note (label) is assumed to be exactly the result of the k preceding notes (features)

Data

- Raw musical data was obtained from a subset of the Lakh MIDI Dataset, available at http://colinraffel.com/projects/lmd/.
- Over 40,000 MIDI files were available; however, only ~150 were used due to time and computation constraints (340,000 examples).
- Each MIDI was sampled at 10 samples / second to determine which notes were held
- A sliding window was applied (similar to n-grams) to obtain training examples.

$$\begin{bmatrix} 47 & 47 & 51 & 51 & 51 & 51 & 51 & 59 \\ 51 & 51 & 59 & 59 & 59 & 59 & 59 & 54 \\ 59 & 59 & 54 & 54 & 54 & 54 & 54 & 35 \\ 54 & 54 & 35 & 35 & 35 & 35 & 35 & 0 \end{bmatrix} \xrightarrow{t+1} \begin{bmatrix} 47 & 47 & 51 & 51 & 51 & 51 & 51 & 59 \\ 51 & 51 & 59 & 59 & 59 & 59 & 59 & 54 \\ 59 & 59 & 54 & 54 & 54 & 54 & 54 & 35 \\ 54 & 54 & 35 & 35 & 35 & 35 & 35 & 0 \end{bmatrix}$$

Features

- Strong assumption: each note depends only on a finite number of preceding notes
- Chords are represented as vectors of numbers (0 = silence).
- Feature set is a list of vectors.
- Labels are vectors as well.

Models

Linear Regression

- Attempts to predict first number of the label vector from the flattened previous vectors.
- Average squared training loss of 181 and test loss of 180 (MIDI values are 0-127)
- Extremely high bias (no better than always outputting the average)
- Illustrates non-linearity of musical creativity

Support Vector Machine

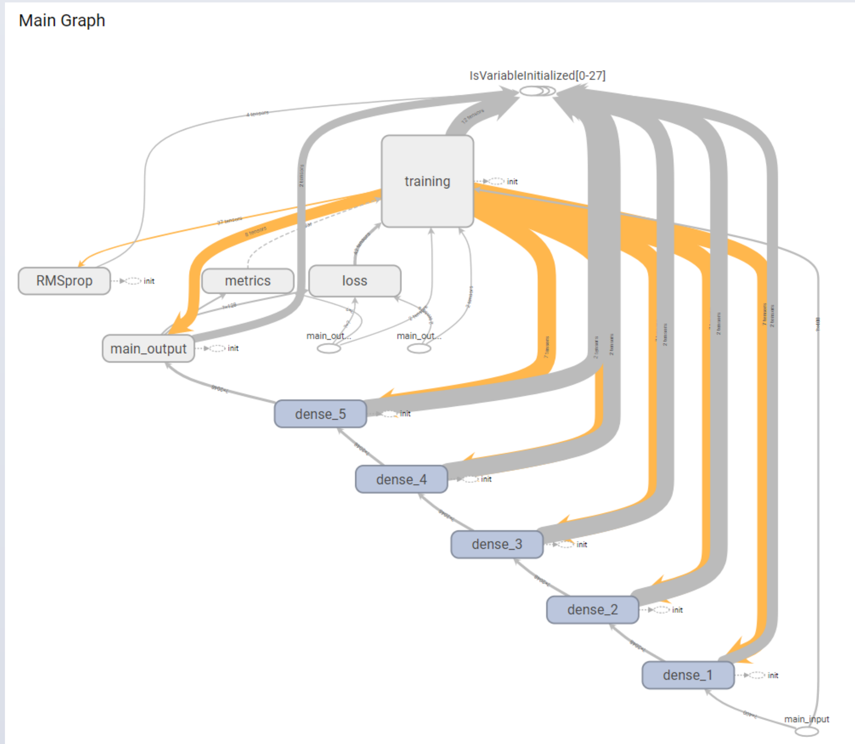
- Kernel used: RBF (γ=1)
- Tried in response to extremely high training error of the Linear Regression model
- 2000 examples in train, 2000 in test (too slow for more examples)
- Train accuracy: 1.0 (perfect)
- Test accuracy: 0.014

Multiclass Neural Network

- Same prediction / classification problem as the linear models
- 3 fully-connected hidden layers of 2048 neurons each (ReLU)
- Softmax activation for output (128 neurons, one-hot)

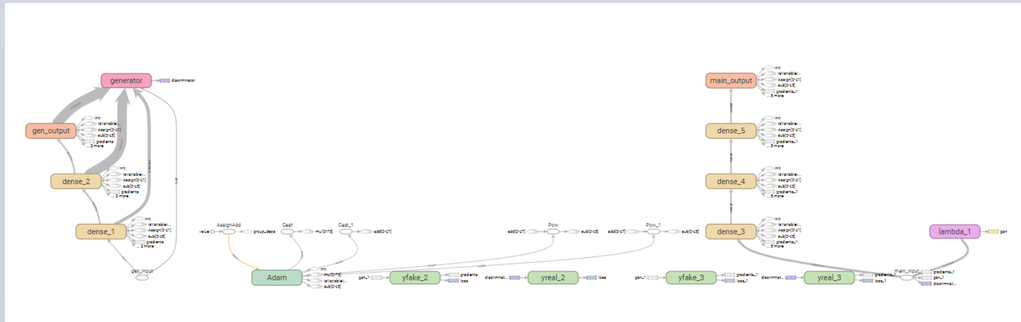
Improved Multiclass Neural Network

- Instead of a softmax layer I use a sigmoid output layer where multiple neurons can be activated, representing depression of a chord
- This eliminates the noise introduced by the arbitrary ordering of the notes of a chord since in previous models, one note had to be selected arbitrarily.
- Increased number of hidden layers because of continued training error problems



Generative Adversarial Networks

- Discriminator input: concatenated and flattened feature and label vectors
- Generator input: 32-dimensional vector of normally distributed noise



Results

- No model was accurate enough to produce any sort of intelligible music.
- All neural network models became stuck outputting the same set of values as if the input data was randomly generated.

Model	Test Error
Linear	181 (mean sq)
SVM	98.6%
Multiclass NN	92.3%
Improved NN	1.5%*
GAN	-

Music composition / prediction was much harder than anticipated. The highly unpredictable art of composing music and the inability for any machine learning algorithm to read minds means that the noise is overwhelming. Even multilayer feedforward neural networks have trouble predicting this time series because so much has to be held in memory. RNNs such as the LSTM have been used in existing research to perform music composition, and likely for good reason. The simplistic approach given here generates a massive number of difficult-to-learn example vectors.

Future Research

Music composition shares a lot with image processing; applying the GAN with an LSTM seems like a good way to improve performance, particularly with the memory aspect

References

Chollet, F. (2017). *Keras Documentation*. [online] Keras.io. Available at: https://keras.io/ [Accessed 11 Dec. 2017].

Raffel, C. (2016). *Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching*. Ph.D. Columbia University.