Anish Shah
Nick Torba

Machine Learning MS#0

Project Idea

We aim to classify the genre of music using a Recursive Neural Network (NN) Restricted Boltzman Machine (RBM) with LSTM architecture schemas (Long Short Term Memory). The inspiration for this architecture stems from many previous projects that successfully generate original works of music. We wish to generalize the principles of this methodology to allow for it to distinguish it between 7 different genres (classical, country, rock, hip-hop/r&b, metal, edm, and pop). Furthermore we want to be able to generate music that is as popular, or human-like, as pre-existing music such as the one used to train our algorithm.

Dataset Description

The dataset will consist of a collection of MIDI files of the aforementioned genres which are an aggregation of hexadecimal commands that will determine the appearance of a note, the note itself, the velocity of the note, etc. From our research it can be seen that the position temporally, the separation of a note into pitch steps, the temporal positioning of a note in respect to the surrounding temporal position of the notes, the actual note itself in relation to other surrounding notes, and the beat seem to be all important features that can be extracted from the MIDI files. Furthermore, we wish to uniquely distinguish all features independently for each instrument within the MIDI file to allow the trained net to look at not only the totality of the song instructions, but also the important features of a genre per instrument. Through our resources we saw that on average every genre should be trained with 50 songs on conventional feed forward neural networks. However using the RNN-RBM schema as we have it to deal with the time and conditional note distribution, we hypothesize that the payoff of such a complex network that would take longer to train would be of equal or more accuracy. As such to deal with the time of training we will be training on 20 MIDI files per genre with 5 validation MIDI files per genre and 10 test MIDI files per genre (which can be increased seeing how long testing takes). These have already been collected from a repository found on Reddit and other free midi websites.
https://www.reddit.com/r/WeAreTheMusicMakers/comments/3ajwe4/the_largest_midi_collection_on_the_internet/
http://www.midiworld.com/
https://freemidi.org/

Planned code/libraries

- The midi package, mido package, or the TonesInTunes MIDI to excel software can be used in conjunction with a csv reader to read the MIDI files

- Extract key features from whatever format the package places it into and clean it into a workable format.
- Numpy to store the cleaned features.
- Tensorflow with packages for all the layers, optimization methods, RNN, LSTM, and RBM to be able to train the collected MIDI data from the stored numpy.
- Use it to classify the model
- Take the trained model and use it to produce music which requires conversion of whatever format python put our “predicted midi” in and turn it back into its hexadecimal byte form (CPickle)

Eval/Baseline

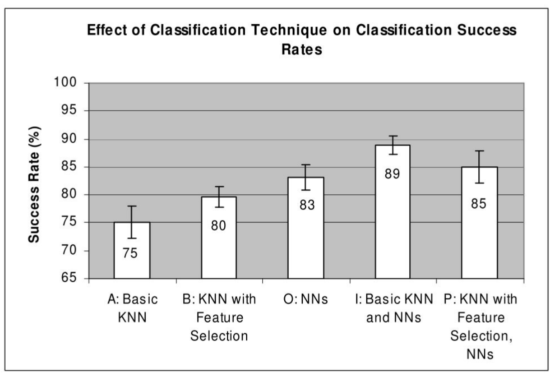

Figure 16: Effect of different classification methodologies within a classifier ensemble on the classification of Taxonomy T-9. Values are averages over all folds and error bars correspond to standard error. Letter codes identify the experiments (see Table 2).

In our research we have not found any research documentation or credible source that uses a Recursive Neural Network for a music genre classification task. As such, we have found a comparable baseline for the model we intend to train. Other projects have been done using feedforward neural networks, KNN, and other variants as seen in the figure above. The figure above shows the accuracy of said trained models on a simple classification task for a reduced taxonomy classification set of 3 genres. Of course, there is an inverse relationship between the number of genres predicted and accuracy. For the DeepSound project listed in the references below they used a CRNN to classify a spectrogram of music with 67% accuracy for 10 genres. As such we can see that the baseline reasonably falls within 60-70% with the well working trained models falling in between 70-80%.

Planned work for MS#1

In order to ensure a minimum viable product by December 7th, we will need to have our input data ready to be used by our architecture. To achieve this, we need to clean our dataset and extract the features we have chosen to create our music. Referring back to the dataset description, we must collect a sufficient amount of MIDI files to convert into the format which will contain the position, note specified by pitch, relative temporal positioning, relative pitch, and beat (and any other features we may deem more necessary).

Collaboration Plan

- Anish: Collection of 50 MIDI samples of 4 genres.
- Torba: Collection of 50 MIDI samples of 3 genres.
- Torba: Will use MIDI packages to make data into workable format i.e appending genre of said file.
- Anish: Set up a learning environment to be able to train a large scaled dataset be it Amazon Web Services, GPU of personal GTX950M computer, or your cluster
- Anish: Feature extraction of half of instruments
- Torba: Feature extraction of other half of instruments
- Anish: Feature extraction of totality of song
- Torba: Creation of Neural Network to compose music on small samples of music
- Anish: Take small sample code, generalize to a larger sample, and place into the Cluster/AWS environment
- Anish & Torba: Tweak net for generation of music

http://www.hexahedria.com/2015/08/03/composing-music-with-recurrent-neural-networks/
https://cs224d.stanford.edu/reports/allenh.pdf
https://cs224d.stanford.edu/reports/NayebiAran.pdf
http://jmir.sourceforge.net/publications/MA_Thesis_2004_Bodhidharma.pdf
http://web.itu.edu.tr/~cataltepe/pdf/2007_Journal_JASP_cataltepeJASP36409.pdf
http://deepsound.io/music_genre_recognition.html