Hip Hop Beats Side Project Notes

1. [RCharlie Radiohead](https://www.rcharlie.com/post/fitter-happier/)
   1. Spotify’s Web API provides detailed audio statistics for each song in their library. One of these metrics, “valence,” measures a song’s positivity. (So sounds like he didn’t scrape the track himself in anyway)
      1. Another key component of a song’s sentiment is its lyrics, and it just so happens that Genius Lyrics also has an API to pull track-level data. (So this of course would be used if I wanted to get at the lyrics and not just the beats)
      2. To determine a song’s sadness, I calculated a weighted average of valence and lyrical sentiment.
   2. When we use spotifyr with the API, and look up an artist, for example, we can get:
      1. Artist, Artist ID
      2. Album id, album type, album image (List form with images in hyperlink form – like 3 each usually – so I’m guessing front, back, and maybe insert or side), release date, release year, release date precision, album name
      3. Cool metrics – danceability, energy, loudness, speechiness, acousticness, instrumentalness, liveness, valence, tempo
      4. Track\_id, track name, track preview, track number, type, track\_uri, external\_urls
      5. analysis\_url
      6. Mode, mode\_name
      7. Key, time Signature, key\_name, key\_mode
      8. Artist – Each row is a 6 column, 1 row df
      9. Available markets (countries)
      10. Disk Number
      11. Duration
   3. Lots of cool analysis that can be done here, but not exactly what I’m looking for
      1. I think I’d want to loop through the track urls and apply the equivalent of pixilation for audio
   4. My goal would be to loop through the external\_urls.spotify column, which links to open spotify online and immediately plays the song.
      1. From there, I’d want to employ some kind of crawler that can scrape the audio.
      2. Would need to control for time if number of streams is what you’re interested in.
      3. Also not sure that we have genre from the spotify API
      4. Can I download Spotify songs as wav files? That would be key and I’m not sure that’s possible.
2. [Spotify Web API](https://developer.spotify.com/documentation/web-api/)
   1. Based on simple REST principles, the Spotify Web API endpoints return JSON metadata about music artists, albums, and tracks, directly from the Spotify Data Catalogue.
   2. The base address of Web API is https://api.spotify.com. The API provides a set of endpoints, each with its own unique path. To access private data through the Web API, such as user profiles and playlists, an application must get the user’s permission to access the data. Authorization is via the Spotify Accounts service.
   3. Note: If Web API returns status code 429, it means that you have sent too many requests. When this happens, check the Retry-After header, where you will see a number displayed. This is the number of seconds that you need to wait, before you try your request again.
3. How to actually analyze
   1. <https://www.r-bloggers.com/intro-to-sound-analysis-with-r/>
      1. In R, there is a library called tuneR that enables us to work with sound clips.
   2. <https://www.analyticsvidhya.com/blog/2018/08/chorrrds-r-package-analyzing-working-music-data/>
      1. chorrrds is a R package that helps you extract, analyze, and organize music chords
      2. Called chorrrds, the package scrapes data from the Cifraclub site and extracts it to your machine. The aim of chorrrds is to help R users analyze and organize music chords and it can be considered a music information retrieval (MIR) package
   3. <https://www.analyticsvidhya.com/blog/2018/02/audio-beat-tracking-for-music-information-retrieval/>
      1. Audio beat tracking is commonly defined as determining the time instances in an audio recording, where a human listener is likely to tap his/her foot to the music.
         1. Audio beat tracking enables the “beat-synchronous” analysis of music.
         2. The goal of the automatic beat tracking task is to track all beat locations in a collection of sound files and output these beat onset times for each file
         3. On a more technical note, beat tracking could be used to for music information retrieval tasks such as music transcription. For example, you can annotate the beat of drums – so that it can then be shared with other music creators and enthusiasts
      2. In python (and prob same in R), wav files are the preferred way to deal with the data
      3. Can use the sample from here to actually try things out.
      4. Approach 1: Onset Detection and Dynamic Programming
         1. What we can do is, find out the locations where there is a sudden burst of sound (aka onset) and mark those moments of time
         2. This is most likely to be the representations of the beat. But it would contain many false positives, such as vocal sound of a person or background noise. So to minimize these false positives, we can find the longest common subsequence of these onsets to identify the beats.
      5. Approach 2: Ensemble of Recurrent Neural Networks coupled with Dynamic Bayesian Network
         1. The gist of the approach is this – we preprocess the audio signal, and then use recurrent neural network to find out the most probable values of these beat times. The researchers additionally use a committee of Recurrent Neural Networks (RNN), and then ensemble their outputs using a bayesian network.
         2. So to boost the performance, what we can do is to train multiple RNN models on a single genre of music files, so that it can catch the patterns in that genre itself. This helps to mitigate some of the data insufficiency problems.
      6. madmom library contains the implementation of various state of the art algorithms in the field of beat tracking and is available on github
         1. It incorporates low-level feature extraction and high-level feature analysis based on machine learning methods.
         2. Further supplement: <https://www.analyticsvidhya.com/blog/2017/08/audio-voice-processing-deep-learning/>
            1. <https://www.analyticsvidhya.com/blog/2016/05/ase-studies-10x-faster-using-dynamic-programming/>
4. <https://www.topbots.com/dope-learning-art-using-ai-generate-deepbeats-rap-songs/>
   1. After mapping lines to high-dimensional vector space, DeepBeat leverages a Ranking SVM to pick the most relevant next lyric. (Can I do this for sound bytes instead of lyrics)
      1. Document ranking algorithms like PageRank use a single static ranking, but accuracy can be dramatically improved by combining multiple features via machine learning algorithms.
      2. Recurrent neural networks (RNNs) are typically used for textual predictions, but since rap lines are relatively short and equal in length, a feedforward network is a simpler and still suitable architecture for this problem.
      3. Lyrics creation was then modeled as an “information retrieval” problem, where the query is the first x number of lines of a song, and the answer is the most relevant follow up lyrics.
         1. A generative model that constructs new lyrics word by word would yield more creative output, but also drive the complexity up significantly. Perhaps an aspiring academic should hustle on this front and produce a Dope Learning 2 paper.
5. <https://medium.com/@snikolov/neuralbeats-generative-techno-with-recurrent-neural-networks-3824d7ba7972> (#1 Guide so far!)
   1. Recurrent neural networks have recently been used with great success to model sequences. Their generative capability has great potential for artificial and AI-assisted creativity.
   2. Generating music is an obvious area of application, and many have successfully used neural networks such as LSTMs to generate melodies and harmonies, even rhythm.
   3. Some have tried modeling sequences closer to the raw waveform, either taking raw samples, or spectral features over short windows.
   4. But roughly, we train the network to predict which note (or combination of notes, or silence) comes next given the notes that just happened.
   5. So, I downloaded drum patterns from various genres — from hip-hop, breakbeat, and R&B to jazz, punk, metal, and more — and started training. (Seems exactly what I want)
   6. After generating automatically, this would be the next step:
      1. Instead of autonomously generating music, an AI can assist people in creating music.
      2. I looped short sections of the generated rhythm and triggered them in Ableton Live. I then added some bass and chord elements to quickly create a house track.
      3. Essentially, I’m only making the choice of how to arrange the track and letting the AI figure out the fine structure of the rhythm. (I’d get someone else to do the ableton part probably though)
6. <http://www.hexahedria.com/2015/08/03/composing-music-with-recurrent-neural-networks/>
   1. Feed Forward Neural Networks
      1. A single node in a simple neural network takes some number of inputs, and then performs a weighted sum of those inputs, multiplying them each by some weight before adding them all together
         1. Then, some constant (called “bias”) is added, and the overall sum is then squashed into a range (usually -1 to 1 or 0 to 1) using a nonlinear activation function, such as a sigmoid function
   2. Notice that in the basic feedforward network, there is a single direction in which the information flows: from input to output.
      1. But in a recurrent neural network, this direction constraint does not exist.
      2. Basically, what we can do is take the output of each hidden layer, and feed it back to itself as an additional input.
         1. Each node of the hidden layer receives both the list of inputs from the previous layer and the list of outputs of the current layer in the last time step
         2. So if the input layer has 5 values, and the hidden layer has 3 nodes, each hidden node receives as input a total of 5+3=8 values.
      3. One problem with this is that the memory is very short-term. Any value that is output in one time step becomes input in the next, but unless that same value is output again, it is lost at the next tick.
         1. To solve this, we can use a Long Short-Term Memory (LSTM) node instead of a normal node. This introduces a “memory cell” value that is passed down for multiple time steps, and which can be added to or subtracted from at each tick.
      4. First, we need to define how good or bad any given output is, given the input. This value is called the cost.
         1. Once we have this cost value, we can use backpropagation
         2. This boils down to calculating the gradient of the cost with respect to the weights (i.e the derivative of cost with respect to each weight for each node in each layer), and then using some optimization method to adjust the weights to reduce the cost.
      5. Hypothetically, what would happen if we replaced the convolution kernel with something else? Say, a recurrent neural network? Then each pixel would have its own neural network, which would take input from an area around the pixel. Each neural network would in turn have its own memory cells and recurrent connections across time.
         1. Now replace pixels with notes, and we have an idea for what we can do.
         2. This is exactly how I’ve been thinking about it!
   3. The model is implemented in Theano, a Python library that makes it easy to generate fast neural networks by compiling the network to GPU-optimized code and by automatically calculating gradients for you.
7. <https://keunwoochoi.wordpress.com/2016/02/23/lstmetallica/>
   1. Word-RNN (LSTM) on Keras with wordified text representations of Metallica’s drumming midi files, which came from midiatabase.com. (So not entire songs)
   2. LSTM model comes from Keras.
   3. Read Midi files with python-midi.
   4. Convert them to a text file (corpus) by my rules, which are
      1. (Temporal) Quantisation
      2. Simplification/Omitting some notes
      3. ‘Word’ with binary numbers
   5. Learn an LSTM model with the corpus and generate by prediction of words.
8. <https://deepjazz.io/>
   1. I built deepjazz in 36 hours at a hackathon.
      1. It uses Keras & Theano, two deep learning libraries, to generate jazz music.
      2. Specifically, it builds a two-layer LSTM, learning from the given MIDI file. It uses deep learning, the AI tech that powers Google's AlphaGo and IBM's Watson, to make music -- something that's considered as deeply human.
9. <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
   1. Recurrent neural networks are networks with loops in them, allowing information to persist
   2. Long Short Term Memory networks – usually just called “LSTMs” – are a special kind of RNN, capable of learning long-term dependencies
   3. All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.
      1. LSTMs also have this chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way.
10. <https://www.newyorker.com/magazine/2006/10/16/the-formula>