James Kaufman, Ahmed Mohamed

Jonathan Gorel, Sebastian Cortes

Eamon Kostopulos

12/4/18

Music Classification: A Study of Genres and Different Network Types

Discerning one genre of music from another can be a complicated task, dependant on the track and related genre in question. When it comes to a broad set of categories however, this job is considerable simpler, to the point where most human-level intelligences are able to tell if a song is more folk than metal. With our final assignment in our Deep Learning class, our topic of choice was to see if a neural network, when fed raw data from a series of tracks, could tell the difference of genre between them. With this project we would use the resources allocated to us and available in the Python language as modules: Tensorflow, a machine learning module, Keras, a neural network module that works with Tensorflow, and Librosa, a module that supports the use and import of MP3 files. In this, we were able to create and develop a neural network that was able to train off of a large database of differently categorized songs, and accurately train and sort them based on a snippet voting system. These are our findings.

Starting our project, our idea was to use some kind of neural network, either a recurrent network, a convolutional network, or an LSTM, to take in raw data from music files to determine the genre of said music. As can be seen in the sources we use in this paper, this idea has seen some application in the field of neural networks already; however our use of a program to break up MP3 files into snippets and convolutional neural networks may be a fairly new development into a reasonably explored field thus far. As far as our raw data is concerned, we used a database called the Free Music Archive that allowed us to use a large amount of unlicensed music in order to train our network.

As it happens, other projects similar to ours have been dreamed up by teams that used the Mel-Frequency Cepstrum Coefficient or MFCC, which use a convolutional network alongside a mel spectrogram to help train the network. When uploading songs that we would use to train our neural network we at first looked at the Mel-Frequency Cepstrum Coefficient (MFCC). This is a form of data derived from what is known as a Mel-Frequency Cepstrum (MFC). An MFC is a representation of spectrum-power based on a short period of time. Upon further progression it was realized that MFCCs’ did not provide enough information for our neural network to be strong enough. Moving from here we then decided that a Mel-Spectrogram would be sufficient in providing a strong and reliable neural network. A Mel-Spectrogram represents an acoustic time-frequency representation of sound: . It is than sampled into a number of data points based around equally spaced times and frequencies on a Mel-Frequency scale (the Mel-Frequency scale is defined a Notably, mel-spectrograms mimic the way a human ear works in terms of separating frequency and magnitude, which ultimately lends itself to the simulation of a more genuine trained neural network. In one such example, a neural network created by Mingwen Dong used a convolutional framework as well as mel-spectrograms while utilizing the Librosa module in Python, similar to ours. Her network used a series of layers, including two convolutional ones. The network architecture she used is as follows:

“1. Input layer: 64 \* 256 neurons, corresponds to 64 mel scales and 256 time windows(23ms, 50% overlap).

2. Convolution layer: 64 different 3 \* 3 filters with a stride of 1.

3. Max pooling layer: 2 \* 4.

4. Convolution layer: 64 different 3 \* 5 filters with a stride of 1

5. Max pooling layer: 2 \* 4.

6. Fully connected layer: 32 neurons that are fully connected to the neurons in the previous layer.

7. Output layer: 10 neurons that are fully connected to neurons in the previous layer.”[1], (pg. 2) Compared to this project, we use two dense layers and two activation layers in our network architecture. Overall I would say that our knowledge of different types of layers, be that concolutional, activation layers, dense layers of other types is rather limited. In addition to this, it’s uncertain if we have they best combination and alignment of these layers for our intents and purposes, so it may be that some experimentation can yield positive results sometime down the road.

In addition to the type of neural network and our combination of layers used to process the raw data, we also need to consider the different choices related to how we want our network to see and understand the data we are giving it. The first choice to make in this case, as well as the one that most other code would be based off of is the type of neural network to be used. Looking at many of the other different coding projects done in the school database, we’re able to conclude that due to the massive successes of the convolutional neural network in terms of image recognition, in addition to the inherently complex and accurate representation of music with a mel-spectrogram, the use of a convolutional network is the most popular by far. Other potential networks either employ some kind of variant of a convolutional network, such as the network designed by Feng Lin and her colleagues while employs a convolutional network and a recurrent network in parallel[3], or an entirely different design such as the conditional neural network designed by Fady Medhat and their co-workers[2], which uses a conditional neural network and masked conditional neural network to temporally recognize patterns in music genres. While either of these two latter networks boast an above average accuracy in relation to the more popular convolutional network, we elected to use a convolutional network as it was more intuitive to work with, as well as a testing ground for other ideas that will be discussed later on in the paper.

According to Mingwen, “Humans’ classification accuracy plateaus at 3 seconds and good results were obtained using 3-second segments to train convolutional deep belief network.”(pg. 2) Based on this as well as the fact that an entire song file may be difficult for the network to digest, we heeded this advice by scraping the data into a series of five second snippets. While around the range of three seconds, our final choice of five seconds was due to the fact that our network would be able to better gauge the each snippet’s genre with this added accuracy. This greatly helped us in speeding up the runtime of the network as well as presenting the data in a way the network could understand it better. Our next step for the project was to have the network guess and check as accurately as it could throughout the different epochs of the program, which boiled down to a question of methodology.

For this same type of problem, we must consider what other computer scientists have done in the past for such a situation. In a fascinating study by Lin Feng, Shenlan Lui and Jianing Yao[3], a hybrid neural network was theorized and created for experimentation in sorting different music tracks into their corresponding genre. In this experiment, a convolutional network and a Bi-Recurrent network are set up to run in parallel with one another, and accept a mel-spectrogram of each three second snippet of each track, similar to Mingwen’s study. What both of these studies have in common, that also differ from our project is that these networks base the final genre sorting algorithm in terms of percents. In such a methodology, one snippet equates to being 85% folk, 10% bluegrass, and 5% miscellaneous, with the final choice of genre depending on these different probabilities. However, our algorithm decides each full track’s genre based on a voting system. With each of the different five second snippets, they are flagged based on their highest probability of being a certain genre, based on a base confidence threshold. Then, at the end of this flagging process for each individual track, each of the different snippets flagged are counted as a vote towards a genre. After the votes are counted, the majority of the votes counted toward a certain genre was used to label the entire track as a certain genre of music, usually positively affecting the overall accuracy of the network. This allows for us to have far higher accuracy on full length songs than we did on snippets - although our neural network capped out between 60-70% validation accuracy on five second snippets, we were able to reach 85% accuracy on full length songs.

Another variant of the typical convolutional model network is the use of a Long Short Term Model (LSTM) recurrent network, which uses a gradient method between layers called Resilient Propagation (RProp) to train and set the weights for the network. This variant, explored by I-Ting Liu[5] in 2014, highlighted the inherent promise and detriments of using a recurrent neural network in generative music composition. While music sorting and music composition sound like dramatically different tasks for a network to tackle, many of the core concepts remain the same; first and foremost being the network’s capacity to understand a song, either in terms of an image generated from it or a temporally understood pattern that uses time and pitch to map out musical phrases. As can be seen in the study, “In our system, we use 88 binary visible units that span the whole range of a piano from A0 to C8 as was done by Boulanger-Lewandowski et al. (2012). The reason why we avoided psychologically distributed encodings or any other dimension reduction techniques but instead represent the data in this simple form is that we believe that a good network should be able to identify harmonically correlated pattern between notes by learning bias. Besides, such representation is flexible in representing both monophonic and polyphonic data.”(pg. 3) We can see here that the way a recurrent neural network understands is quite different than our more traditional convolutional network. However, this difference makes sense in practice, as understanding the patterns in each different pitch would be theoretically more useful in music composition than in music classification. Based on this example however, it’s good to know that music can interact with different neural networks in a variety of ways, which may pave the way to bigger and better uses of deep learning trying to understand music.

In terms of what we got right and mistakes we made, many of the different goals we had in mind for our network ended up coming to fruition. This starts with the larger goal we were pursuing, which was if we were able to create a network that was able to assign different music tracks to genres within a human level accuracy, which statistically translates to around 70% accuracy in a network. The result we eventually got with the majority of genre accuracies was roughly 90 percent, a benchmark that far exceeded our goals when it came to human comprehension of music genres.

Although some wouldn’t consider this a success, our group along with all the other members of our class were tasked to use Intel DevCloud in order to expedite our network learning process, as our limited computer power was an issue when it comes to the long stretches of time we had to allot to training the network. With DevCloud we were able to train our network, save its weights, and use the weights on our dataset to have the network classify the data as best it can. I would consider this to be a success only due to the fact that certain aspects of DevCloud were notoriously difficult to use in terms of an intuitive process, at least coming from our level of knowledge concerning advanced web services and their UI and authorization conditions.

Some issues we ran into were instances where one genre was overwhelmingly identified as another, such as every other rock song being incorrectly chosen as a pop song. This interaction also could be seen in the Experimental music genre, as it is often used as an umbrella term for more ‘unconventional’ patterns that can be seen by the network in its mel-spectrogram. Although it was not our first choice, we ended up taking out these genres entirely to improve accuracy. A lot of these issues originated with the need to use unlicensed data - we were unable to pick songs that we thought fit in a genre, and instead had to rely entirely on a dataset that we were fairly unfamiliar with. Although this was not that big of a problem, this created issues when what we thought was pop music was nothing like the pop music in the dataset. With a longer time or more resources to work on this project, we would be able to use multiple different sources to create a dataset that was more specified to our goals, but the constraints of this project did not allow for that.

Another problem that turned out to be unique to our project was one of licensing and music rights related to the songs we were using as our dataset, as their intellectual properties are protected. Rather than cite each and every song we were supposed to use in this case, we were able to find a dataset called the Free Music Archive in order to use a large volume of unlicensed music that was thankfully open to the public for neural networks such as these. Only now do we understand that certain fairly renowned data sets have been available just for this purpose, including the Ballroom Music Data Set, which certainly meets the criteria for an open dataset full of public domain songs. With these resources, we were able to utilize a large dataset in order to better train our network.

Going forward, if we were to recreate this objective with only the idea of improving our network in mind, the choices that we would make would be to increase the amount of genres that this network was able to classify. Although it is possible that having more and more different genres may cause the same issue that we experienced with the Pop and Rock genres interacting badly with one another, it is also entirely possible that adding in more data would strengthen the differences that the neural network can find between these genres with overlaps.

In terms of the different types of networks one can use for this task, it quickly becomes clear that a convolutional network is the most popular in terms of the plain fact that it makes sense, and is inherently easy to understand; take a track, convert it to an image, and run it through the network to train it. However, as is seen above, many different scientists have experimented with different types of networks and have achieved great success in music classification, often exceeding the threshold of human recognition. While we were very comfortable with our final results and accuracies, an important point in understanding and creating networks is that as long as you have the time to support your endeavours, it’s a good thing to experiment. Be it convolutional, conditional, recurrent, a dual network with both recurrent and convolutional, or even a convolutional recurrent network that “take[s] advantage of convolutional neural networks (CNNs) for local feature extraction and recurrent neural networks for temporal summarisation of the extracted features”[4](Choi, pg. 1), each network has its own set of strengths and weaknesses, and it’s anyone’s game to see which one ends up classifying music in the end.

All in all, this project was an overwhelming success compared to our initial expectations. We reached an overall accuracy of 85% throughout five separate genres and had accuracies of up to 97% in specific subgenres. We were able to take an unsorted dataset and use limited resources to sort it entirely into the correct folders, and then create a neural network that can classify both new and old music correctly a vast majority of the time. Although other projects and studies were more successful in the end, these studies had the advantage of much more time, people, experience and resources. While this is something that we still want to keep working on and improving, the current state of our classifier leaves little to be desired.

Bibiliography

1.

Dong, Mingwen. *Convolutional Neural Network Achieves Human-Level Accuracy in Music Genre Classification*. 2018. *EBSCOhost*, [proxy-sm.researchport.umd.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edsarx&AN=edsarx.1802.09697&site=eds-live](http://proxy-sm.researchport.umd.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edsarx&AN=edsarx.1802.09697&site=eds-live).

2.

Medhat, Fady, et al. Automatic Classification of Music Genre Using Masked Conditional Neural Networks. 2018. EBSCOhost, doi:10.1109/ICDM.2017.125.

3.

Feng, Lin, et al. *Music Genre Classification with Paralleling Recurrent Convolutional Neural Network*. 2017. *EBSCOhost*, [proxy-sm.researchport.umd.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edsarx&AN=edsarx.1712.08370&site=eds-live](http://proxy-sm.researchport.umd.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edsarx&AN=edsarx.1712.08370&site=eds-live).

4.

Choi, Keunwoo, et al. *Convolutional Recurrent Neural Networks for Music Classification*. 2016. *EBSCOhost*, [proxy-sm.researchport.umd.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edsarx&AN=edsarx.1609.04243&site=eds-live](http://proxy-sm.researchport.umd.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edsarx&AN=edsarx.1609.04243&site=eds-live).

5.

Liu, I.Ting, and Bhiksha Ramakrishnan. *Bach in 2014: Music Composition with Recurrent Neural Network*. 2014. *EBSCOhost*, proxy-sm.researchport.umd.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edsarx&AN=edsarx.1412.3191&site=eds-live.