Logan Blakely, Leila Hawana, Erik Schmauss, Eric Tsai, and Li-Yun Wang

CS546: Advanced Topics in Machine Learning

Final Project Report

12 June 2018

Genre-Bending GANs

Blurring the Boundaries of our Favorite Musical Genres

## I. Introduction

Music is an interesting experience because it is so vastly different for everyone. Performers can communicate emotions, consumers can enjoy it on a variety of different levels, and each piece conveys a story or a message in its own way. Within music, there are genres that help connect the performers and the listeners by similar tastes in style, composition, or overall art. As time goes on, these genres develop characteristics that their followers appreciate. These characteristics could be level of complexity, harmonic structures, lyric content, or even simply the overall experience from listening to a particular genre of music. This is part of what makes the art so interesting.

We explore the characteristics of these genres from a machine learning perspective. In our work, we use a generative adversarial network (GAN) written by Olof Mogren [1][3] to generate music based off of the features of the genre that the network learns. With this generated music, we attempt to take our experimentation further by combining features to create music that sounds like the combination of the genres. We explore a variety of ensembling techniques in the following sections and report on the results below.

## II. Background Information

In this section, we will discuss some of the pertinent background information required for understanding the fundamentals of this project. We will begin by talking about some basic music theory fundamentals and then transition into pertinent machine learning techniques.

### Music Theory

When it comes to characterizing a particular style or genre of music, there are a number of musical features that are assessed. Some of the major features include harmony, key, and rhythm. We will discuss these three features in brief.

#### Harmony

Harmony is the combination of a particular set of simultaneously sounded notes that produce chords and chord progressions that have a desired effect on the audience. Sometimes these progressions are familiar and comforting while other times they are unsettling and lead the audience on an emotional roller coaster. Genres utilize harmony as a signature feature of the music portraying its style. For example, pop songs utilize the “four chord progression” which can be heard in songs like *Hey, Soul Sister* by Train, *Let it Be* by the Beatles, and perhaps the best song ever written, *Don’t Stop Believing* by Journey, to name a few. To emphasize this, Axis of Awesome wrote a piece combining famous pop songs via the four chord progression and their respective lyrics. This video can be found in [6].

#### Key

Coupled with harmonic progressions, certain keys become associated with musical genres. A key is a collection of pitches that can be used. Each pitch serves a different purpose and has corresponding chords with different emotional effects. One example of this is the tonic vs dominant pitches within a key. The dominant note acts as a pitch causing tension while the tonic note releases the tension caused by the dominant. If a dominant note is played without the following tonic, the audience may be left unsettled, depending on the context.

One common example of characteristic key in a genre is found in jazz; instead of using key signatures typical of pop, classical, electronica, or other common genres, jazz utilizes the blues scale which modifies the typical key signatures of regular scales by lowering some of the pitches in the key. This gives it the characteristic laid back and funky feel (along with its rhythmic swing) that people associate with the genre.

#### Rhythm (and Tempo)

The third feature, rhythm (and tempo), also have the ability to characterize a genre. By definition, rhythm refers to the systematic arrangement of musical sounds, principally according to duration and periodic stress.[7] Tempo refers to the steady pulse in which rhythm lies. Some genres like to experiment with rhythm while others tend to keep it consistent and predictable. For example, in the romantic era, musicians bent time to romanticize the rhythm. This became a trademark of music of the romantic era. Another example is jazz music. In jazz, musicians like to “play it straight” or “swing it”. These are two variations on rhythm that are distinctive jazz. A third example is the rhythmic motif associated with dance music (which can be audibly visualized by repeating “boots and cats” over and over).

There are genres that use rapid rhythms and tempos while there are others that remain slow. One example that is famous for the 120 tempo marking are marches. This occurred out of necessity rather than stylistic choice. Since marches are traditionally performed in parades or marches to war, the tempo of the march needed to be one that was comfortable for soldiers and musicians to march to. Thus, marches are associated with the steady 120 tempo.

### Generative Adversarial Networks (GANs)

For the heart of the experimentation, our group decided to use a generative adversarial network (GAN) to generate music. A GAN is an unsupervised learning technique implemented by a system of two [neural networks](https://en.wikipedia.org/wiki/Neural_network) competing against each other in a [zero-sum game](https://en.wikipedia.org/wiki/Zero-sum_game) framework.[8] The combination of a generative neural network and a discriminative neural network working together create a system that will generate new data samples when fully trained.

The generative neural network is a network that learns how to generate the data. It trains on the input samples to the network and learns to detect significant features in the samples. The goal of the generative network is to create a new, generated sample that is indistinguishable from the training samples. During the training phase the network is fed original samples (classical music clips for example), trains using a variation of gradient descent for some number of epochs, and then a series of new samples are generated.

The discriminative neural network trains on the same original data as well as the newly generated data and tries to identify whether or not the music it is seeing is generated or not. The discriminator also trains for several epochs, receiving both original and generated data as input.

Training of the GAN as a whole continues in this way, alternating between training the generator and the discriminator. The generative model is trying to “deceive” the discriminative model while the discriminative model is trying to identify the generated samples. Ideally, the two networks remain in equilibrium as they train, with neither model gaining the upper hand over the other. The GAN is well trained when the generative model is able to adequately “fool” the discriminator and/or when the output being produced is acceptable.

### Ensembling

In order to generate music that combined multiple genres, the experimentation took the approach of ensembling (amongst other techniques). Ensembling is the process of taking multiple models output and combining them to produce a single output. With regards to classification or regression problems, this can be in the form of voting, random selection, averaging, or any other type of system to determine the classification. When it comes to generation, this becomes a bit more complex.

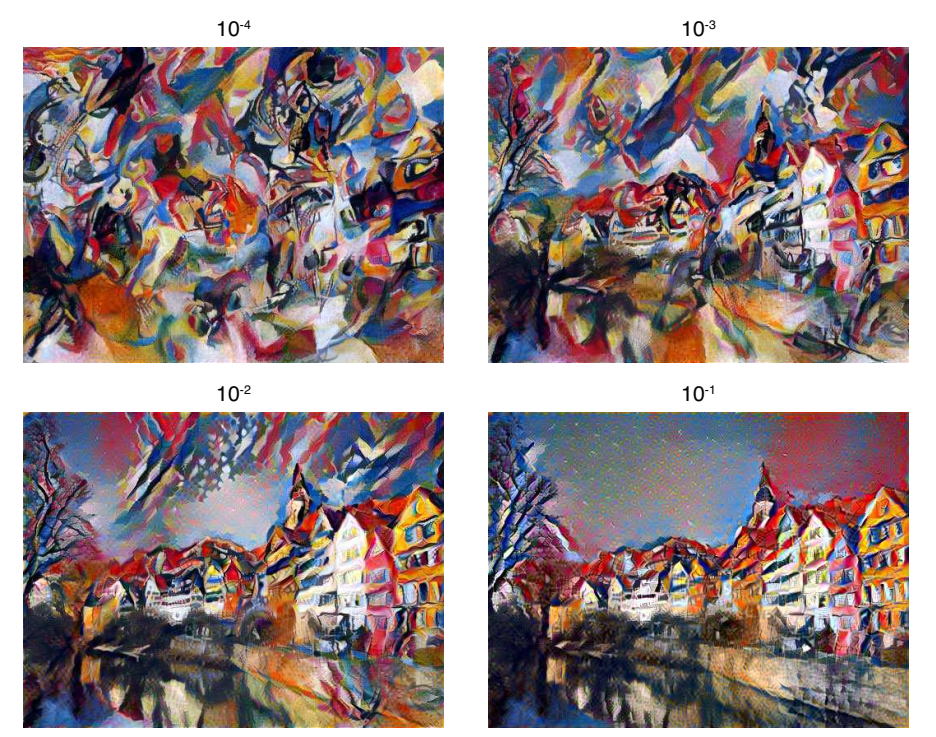
With regards to ensembling GANs, it appears that there is no standard way to do this. This is because the GANs themselves are trained to discriminate and generate their particular data. Thus, combining feature maps, training data, or any other input that could be used for ensembling is not a trivial task. We will explore some experiments for ensembling GANs in later sections.

## III. Related Work

There has been a lot of work done in this field of music generation and style transfer. We will briefly discuss each of these works and their impacts in the sections below.

### Style Transfer

Gatys et al wrote the paper *Image Style Transfer Using Convolutional Neural Networks*[5] to demonstrate how they could detect the style from one image and the content from another and combine them to generate a new image that possessed the style from the first image and the content from the second. In their work, they were able to utilize gradient descent and experimental tuning to manipulate how much style and/or content was represented in the image. An example of this is shown below:



In this image, it is evident that the image with the most stylistic influence is in the upper left corner and the image with the most influence of content is in the lower right corner. Everything else in between is some sort of blend tending toward one of those two extremes. This is an example of how they are manipulating their generated artwork.

In one of the experiments, we attempt to take a similar approach to Gatys et al and manipulate the features to get different blends. We will elaborate more on this in the sections to follow.

### Music Generation

Many musical generation techniques today have been attempted. Each one has used a variety of techniques, from neural networks to autoencoders, grammars, and ensembles that form complex structures. Jean-Pierre Briot, Gaetan Hadjeres, and Francois Pachet examined most of the architectures researched and wrote a survey on the most promising techniques currently in the field. *Deep Learning Techniques for Music Generation - A Survey*[4] is a journey through what has been done in the field of music generation, from data analysis to musical analysis and algorithmic techniques.

In this work, Briot et al explore the different approaches to music generation. On a high level, they note research in the following areas:

1. Multilayer Neural Networks
2. Recurrent Neural Networks
3. Long Short Term Memory Networks
4. Autoencoders
5. Stacked Autoencoders
6. Restricted Boltzmann Machines
7. Variational Autoencoders
8. Convolutional Arch Patterns
9. Conditioning Arch Patterns
10. Generative Adversarial Networks
11. Reinforcement Learning
12. Various Compound Architectures (such as DeepBach, Recurrent GANs, and Hexahedria, to name a few)

With each of these architectures, they proceed to discuss the pros and cons of each and discuss strategies for improvement in the field as well as improvement with the work that has already been done.

This work provides significant insight as to the various directions this project could have gone. For our purposes, we decided to pursue the route of (10) primarily due to our groups’ interest in GANs and desire for experience to work with them. We will discuss in future sections how these are used and their respective pros and cons.

## IV. Architecture

The global architecture for this project is fortunately very simple. On an abstract scale, the architecture consists of two GANs constructed the same way but trained on different data. The output from the GANs then is fed into some sort of ensembling method that will generate music based on features from both GANs. During experimentation, our group tried various ensembling techniques requiring various types of input. The two main forms of input used in this project were the generated midis from each GAN, the input midis for each GAN, and feature maps created from each network. Below is an architectural diagram using data from each GAN to ensemble the networks and generate music.



In the linear regression blending experiment below, the network tunes four distinct features for each genre. These features are:

1. Time since last tone: This is essentially the rhythmic rest between pitches.
2. Tone length: This is essentially the rhythmic value of the pitch.
3. Frequency: This tells the network the pitches that are used.
4. Intensity: This is the dynamic range/accented patterns that emerge in the music.

Interestingly, the network believes each of these features to be pertinent to identifying a genre and generating similar music. As discussed previously, features 1-3 are more directly related to genre classification. However, intensity was not discussed. Dynamic range and intensity by itself is not enough to classify a particular genre. On the other hand, when coupled with other features, it can add some insight to either secure a prediction or cause uncertainty. The reason it is not on its own enough is because there are no distinctive traits the dynamic range has to particular genres. For example, a piece of ska music could be played extremely soft and still be recognized as ska music. Further, a lullaby could be played extremely loud and intense and yet it would still be classified as a lullaby because of the other musical and harmonic features. Thus, dynamic range and intensity is not enough on its own to classify a particular genre in a musical analysis setting.

It is important to note that although a musician can do an analysis with one particular feature to determine genre, a network cannot. This is because a musician can spot the one or two key characteristics associated with a genre based off of background knowledge and their ability to verify suspicions via proof rather than look through an entire search space for the answer. A network on the other hand can only evaluate features it is exposed to. Thus, if a network were to take any one of these features and try to classify all genres based off of the single feature, the network would fail. However, by having an arsenal of features at its disposal, it can identify genres based off of the value it gives for each feature.

## V. Experiments and Results

In these experiments, we worked with a variety of ensembling techniques and data augmentation techniques in order to try to improve the output of the network. We did this in order to investigate a wide range of techniques to ensemble the GANs and generate appropriate output. In the process, some important features became eminent (as discussed above) and were used as guidance in the ensembling process.

Each experiment will be explained below along with its corresponding results.

### Data Augmentation Experiments

#### Normalizing the Input

In this experiment, we ask the question about what happens to the quality of the output when it is normalized? This can be done in a variety of ways: normalized pitch, tempo, or even dynamic range. For this experiment, we took the original training data and normalized the pitch so that every piece was in the same key and either in that key or its parallel minor (the minor scale starting on the same pitch as the major scale).

Unfortunately, this experiment did not meet much success. The samples, although closer in pitch as expected, did not generate very coherent music. Further, after training for approximately 240 epochs, the music did not improve and began generating blank music files. Thus, the network stopped learning after approximately 240 of the 1000 epochs trained. This could have happened for a variety of reasons. However, the most probable reason was lack of data. Not only was this music transposed (and therefore lacked variety in pitch), it all came from a single composer (Bach) and ended up being less files than our normally trained GANs. Thus, with more samples and greater variety, this may have promising results with some finer grained tuning and diversity in data.

### Genre-Bending Experiments

#### Ensembling Data with GANs

In this experiment, after training each GAN individually on a genre, music of both genres was input into a third GAN to be trained and generate music that combined features of both genres. In our first experiment, we attempted mixing input data from the original GANs to train the new GAN. The diagram for this can be seen below where Original Classical (OC) and Original Rock (OR) were the inputs and the output was Generated combination of Classical and Rock (GCR):



Although this experiment did not use any data generated from the original two GANs, we attempted this as a baseline for the next experiment which used the generated music from each GAN as input to the new GAN. This diagram can be seen below where Generated Classical (GC) and Generated Rock (GR) were the inputs and we hoped the output would be Generated combination of Classical and Rock.



In the first experiment, the GAN did not end up producing music that was recognizably a mix of the two genres. Instead, the model ended up producing music that was similar but not quite the quality of the GAN that produced music of a single genre. We hypothesize that this may be partially due to the large disparity between individual the samples. There are no ‘combined’ original samples for the discriminator and so it is always comparing the generated samples to an original classical or an original rock sample and the possible variety is quite large between, for example, a Metallica song and a Bach song. This makes the job of the generator much more difficult, causing confusion and, thus, lower quality output. So, what is the equilibrium in this situation? From listening to the samples, the output from this experiment was significantly more muddled and frenetic compared to training each genre individually.

To further support this theory, the second experiment failed as well, producing only blank or corrupted files. In addition, we hypothesized that this was due to sample length. Since the original input samples were vastly longer than the generated samples, we believe that the program parsed white space and attempted to use it as input data rather than taking the generated samples as is and parsing them. The samples generated from a fully trained network were ~15 seconds each. We believe this is what led the second experiment to generate blank or corrupted files. It may have been possible to troubleshoot this issue and alter the source code to accommodate shorter input samples.

Overall, this experiment exposed some of the inner workings of the GANs as well as what happens when data is combined in the network. This experiment was not a complete success on the music generation aspect but did lead us to working with our final experiment, linear regression blending.

#### Linear Regression Blending

In this experiment, the generated classical music was combined with output generated from a GAN trained on 209 songs from the band The Beatles. The Classical music GAN was trained for 291 epochs while Beatles music GAN was trained for 244.

The classical music output has wide variation across the four music features examined. The music generated by classical training has a wider range of dynamics, punctuated by loud, drawn-out notes and chords consisting of a wide pitch range. The length of notes vary, all creating an output clip with a relatively high degree of texture. In contrast, the music generated by Beatles training is far more rhythmic and more uniform in intensity. The melody moves around a few pitches spaced relatively close together. This brings to mind a bass line, or a simple melody walking around the space of a few notes.

The generated features consist of an 20x4xP tensor, where each of the four columns represent a different feature of the music and where the dimension P increases with the length of training and represents the number of musical events in the generated sample. The first row of each 20x4 matrix within the tensor is chosen to create a new 2-dimensional matrix of size Px4, such that each matrix within the tensor represents a different musical event (such as a tone change). Rather than choosing simply the first row of each 20x4 matrix, we tried different selection techniques (choosing the ith row or choosing the i mod P row), but this did not drastically alter the output.

As such, we now had five features: the original four columns, plus the index P of the matrix which is the number of the musical event in the output. We hypothesized that each musical event is a combination of the 5 features which gives each style its unique sound. Thus, we can use the P index plus 3 of the features to try to predict the fourth within the classical genre and then separately within Beatles music. We applied linear regression, resulting in the R2 scores:

|  |  |  |
| --- | --- | --- |
| **Predicted Feature Column** | **Beatles** | **Classical** |
| 0 - time since last tone | 0.401 | 0.350 |
| 1 - tone length | 0.391 | 0.528 |
| 2 - frequency | 0.170 | 0.450 |
| 3 - intensity | 0.221 | 0.221 |

Then we used the model trained on classical music to predict a single feature of Beatles music by applying the classical model to the Beatles music as input. For example, we trained the regression model to predict the frequency feature (column 2) based on the other three column features plus the P-index. Then we fed in the Beatles data, minus the frequency feature, into the classical music linear regression model, to create a new predicted frequency feature. We hypothesized that this would give us output with the rhythmic feature of the original Beatles clip, mixed with a wider range of frequencies like we see in the classical music output.

Indeed, our output of blended music is as we predicted. Notably, we hear chords with low bass notes in the mixed music that was missing in the original Beatles clip. However, the same on-beat approach and more uniform dynamics of the Beatles music persist in the blended output. The blending works as expected in the opposite direction as well; classical music with predicted Beatles pitch frequencies keeps the wider range of tone length and tempo, and pairs it with a much simpler melody and a smaller range in pitch frequency.

We expected that features with high R2 scores would produce more obvious blended outputs; however, this is not necessarily the case. Note that in general all R2 values are relatively low. Additionally, the column with the lowest R2, feature 2 of Beatles music, produced some of the best output when blended. While other feature columns were used to produce blended output, the most noticeable changes occurred in the feature 2 prediction. We hypothesize that this is primarily because even a small change in tone is easy to discern, while small changes in the length of a note or the intensity of a note are more challenging to hear.

## VI. Future Work

An alternative method using gradient descent was proposed to minimize a loss function based on the original source music feature and the predicted target feature. The Image Style Transfer paper by Gatys et al., used a linear combination of the squared loss between the feature representations of the two outputs. This was attempted but ultimately unsuccessful. Future work may include combining multiple changed features or weighting the features using the linear regression approach and may include additional exploration into other optimization techniques.

In addition, exploring other kinds of generative models would be beneficial for future work on this project. Through experimentation, it was found that the GAN (although sufficient) was not beneficial for this particular task. Modifications in the GAN or using other architectures would be interesting to compare and to determine what works best for the intricacies of musical analysis.

## VII. Conclusions

For our experiments, we used a generative adversarial network written by Olof Mogren in his paper *C-RNN-GAN: Continuous recurrent neural networks with adversarial training*[1][3] to generate music of different genres. Our primary goal was to ensemble the genres so that we could produce music that had the essence of both styles. Throughout our experimentation, we found that most methods were either failing or too complex with the GAN to use in a limited time frame. In retrospect, there are a number of architectures that have already been shown to produce musical output that is recognizable as belonging to a specific genre. Those may have been a better choice as the primary music generation framework than using a GAN. Additionally, we note that the most success in our experimentation came from the workhorse of machine learning: linear regression. While initially overlooked in favor of more advanced techniques, overall, we learned that sometimes the simplest approaches are the most fruitful.

## 

## Sources

1. C-RNN-GAN: Continuous recurrent neural networks with adversarial training by Olof Mogren (<https://arxiv.org/abs/1611.09904>)
2. Midi Samples ([www.midiworld.com](http://www.midiworld.com))
3. Source Code by Olof Mogren (<https://github.com/olofmogren/c-rnn-gan>)
4. Music Generation Architectures (<https://arxiv.org/pdf/1709.01620.pdf>)
5. Style Transfer Paper (<https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Gatys_Image_Style_Transfer_CVPR_2016_paper.pdf>)
6. Axis of Awesome (<https://youtu.be/oOlDewpCfZQ>)
7. Google Dictionary
8. Information on GANs (<https://en.wikipedia.org/wiki/Generative_adversarial_network>)