COMPARING RNN PERFORMANCE ACROSS MUSICAL GENRES AND INSTRUMENT CLUSTERS

Agustín Krebs
Pontificia Universidad Católica de Chile
akrebs2@uc.cl

Alexander Rusnak
Ecole Polytechnique Federale de Lausanne
alexander.rusnak@epfl.ch

Axel Sjöberg
Lund University
ax3817sj-s@student.lu.se

ABSTRACT

The field of music generation is interesting for its interesting artistic contributions to musical canon, but can also be used to explore the features of musical data in novel and exciting ways. In this work we analyze similarities and differences across genres of music and instruments using neural networks. We use a recurrent neural network trained on modified MIDI file data to generate four separate instrument tracks (piano, guitar, drums, and bass) for three genres: pop-rock, ballad and house music. We want to use the differing loss values and distributions of these models to discover new or further explore existing features of musical notation data. Using these models we found that the melodic instruments carried more complexity in the ballad and pop rock genres than the house genre, while the more rhythmic instruments like drums and bass were more difficult to learn for the house genre. We determined that, particularly in the house music genre, velocity accounted for a higher proportion of measured model loss than pitch loss when looking at rhythmic instruments.

1. INTRODUCTION

One of the fundamental cornerstones in categorizing, organizing and describing music is genre. Borrowed from French, where it literally translates to "a kind", musical genres are used to relate a song to a larger group where the members share typical characteristics. The characteristics that form the basis for differentiating and describing genre are often related to the instrumentation, the harmonic content and the rhythmic structure. The task of articulating precisely the difference between genres is a complex one due to the lack of formal definitions and the soft boundaries between neighboring styles. To make things even more complicated, genre is also to a large extent a cultural phenomenon and two genres might be distinguished based on their cultural context rather than their musical context. Two very similar pieces can belong to different genres because of their context and vice versa two very different pieces can belong to the same genre for the same reason.

In this paper the musical context of genre is analyzed to get a reasonable scope of the study.

Historically, genre classification have been made by human experts [5], but with the surge of available songs in digital format, there has been an explosion in the amount of available data needed for computational analysis of genre. Automatic genre classification has become a very popular research domain, and every year music genre competitions are held by the international contest MIREX. In 2014, Sturm [7] showed that many at the time state of the art Music Information Retrieval (MIR) Systems, were often relying on characteristics in the dataset confused with the ground-truth. These systems were by no means bad at solving their tasks, nonetheless, they were often found to solve tasks not by addressing the musical problem. As a result of this, many of the models might not provide results that the researchers who uses them are interested in. In our research we want to use neural networks, which lack an inbuilt knowledge of musicology, in order to explore what genre distinctions can be observed stemming from their differential ability to learn artifacts of the musical composition.

In our research we use a recurrent neural network architecture with LSTM cells to train 16 models on modified MIDI file data, each to generate a specific instrument (out of piano, guitar, drums, and bass) track for three genres of music: ballads, pop rock, and house music. With this method, our ambition is to minimize our reliance on characteristics confounded by predefined conceptions of ground truth from musicology and to instead use the relatively unbiased training and testing of the models to elucidate the characteristic features and key differences of genre and instrument notation.

2. RELATED WORK

One of the earlier works on the subject was done in 2002 by Eck and Schmidhuber where they demonstrated that a RNN can capture not only the local structure of a melody but also the long-term structure of a musical style. [2]. Since then, using neural networks to both generate new music and retrieve information from music have been a very popular research topic.

In 2010, Li, Chan, and Chun used a convolution neural network (CNN) for automatic musical pattern feature extraction [4]. With their novel approach, the authors showed that the extracted pattern features are informative for genre classification tasks, and today, the state of the art genre classification systems still use CNN.

In 2019, Zhou et al. [8] presented Bandnet; a Beatles-style composition machine based on an RNN that had been trained using MIDI-files. This implementation was similar to ours in the sense that it combined multiple instrument tracks composing concurrently.

3. DATA AND PREPROCESSING

3.1 Datasets

For this work, we used the data from the project Million Song Dataset [1] and the Lakh MIDI Dataset v0.1 [6] . In particular, we used the subset called “lmd aligned” from Lakh MIDI Dataset, which consisted of 45,129. The reason for choosing this subset is that we have matched meta-data for all the songs in this subset. As you can see in Figure 1, several MIDI representations and the artist were available for each song. For all the artists we have genre tags that have been collected using automatic annotation based on web-scraping. An important remark here is that the genre tags are associated with the artist and are contained in a non-empty set which is unique for every artist. The fact that the genre tags of every songs are inferred via the artist, means that all the songs by one artist have the same set of genre tags.

For all the genres in an artist genre-set, there is a normalized frequency score. For every genre tag an artist have, the normalized frequency score of a genre tag describes the the number of times the web-scraping tool has identified a connection between the genre and the artist (frequency), divided by a normalizing factor, which is different for every artist. The normalizing factor of an artist is the maximal frequency in their genre-set. This means that for all the artist, every genre tag in their genre-set have a corresponding frequency score in the range (0,1].

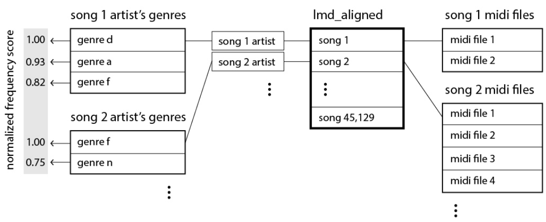

Figure 1. A schema of the information contained in the dataset

3.2 Genre Selection

The number of genre tags in the artists’ genre-sets is on average 11.59. We reduced the number of genres by, for every artist, only retaining those genre-tags in the artist’s genre-set, which had a normalized frequency score higher than 0.9. This was done in order to reduce the irrelevant genres which the artist by accident might have been tagged with and to only get the most relevant tags. After this reduction, each artist had on average 3.5 genre-tags in their genre-set.

Each of the songs in our dataset was paired with the filtered genre tags of its artist. Moreover we created global genre-sets P_{genre}, where the songs were put in one global genre set, if they had that same genre tag in their own local genre-set. Figure 2 shows the total number of pieces in the global genre-sets for some of the most popular genres.

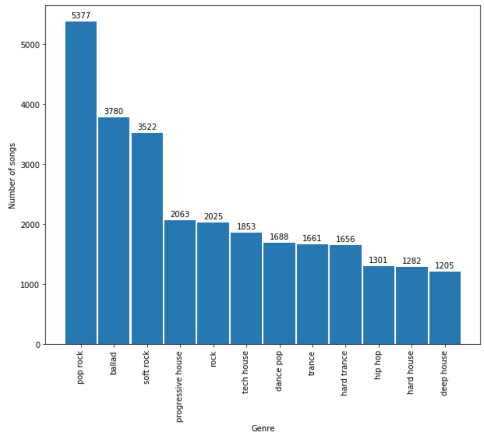

Figure 2. Number of songs for some of the most frequent genres

In order to find which global genre-sets have plenty of overlap, we used the Jaccard similarity, defined as:

$$J(A,B) = \frac{|P_A \cap P_B|}{|P_A \cup P_B|} \quad (1)$$

where A and B are genres and P_A and P_B denote the respective global genre-sets.

We did this because we thought that training models on different datasets with very little overlap would generate larger differences between the models. Our purpose of this work was to find differences between features across genres and with more different models for each genre, the differences in features would hopefully become more pronounced. In Figure 3 the Jaccard similarity for some of the most popular genres is presented. As can be seen in the figure, there are many songs that are tagged with the same set of genres. Some genres for which this phenomena is very pronounced are progressive house, hard house and trance.

We selected pop rock and ballads to train the models as they are the two global genre-sets with most songs in them. In addition to this we merged the global genre-sets: Progressive house, hard house, deep house, trance and hard trance to a "house" genre in order to obtain a set of pieces that is similar in size to that of the pop rock and ballads genre-sets. We find this merging of the house genres reasonable as they are according to us very similar, which the large overlap between the genres also suggests. The reason for not choosing the soft-rock genre was that the merged house genre was both larger and less similar to the pop rock and ballads sets than the soft-rock set was. As stated above, we wanted to find differences between features across genres and by having less similar genres, these differences were thought to be more pronounced.

Table 1, shows the Jaccard similarity between the selected genres. The Jaccard similarity between ballad and pop rock is rather high. This in itself is not something that

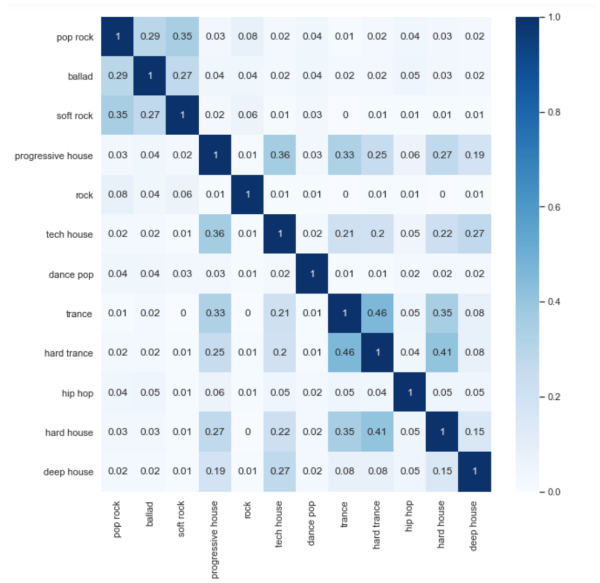

Figure 3. Jaccard similarity for the most popular genres

is wrong, e.g plenty of the songs by bands like Journey and Foreigner could be classified as being both ballads and pop rock songs. The problem arises when there is a song by an artists in the dataset that is not in their predominant genre style (e.g. if one artist that mainly make ballads and then experiment and create a hip/hop or metal song, this song will be classified as a ballad). Obscurities like these makes the task harder. However, this also means that if we are we still able to observe differences between genres, we will have an even stronger result.

	pop rock	ballad	house
pop rock	1.00	0.29	0.05
ballad	0.29	1.00	0.06
house	0.05	0.06	1.00

Table 1. Jaccard Index for the genres used to train the models

3.3 Instrument Merging

Another important aspect of the preprocessing was the instrument part, which focused on reducing (and therefore, transforming) the instrument space. To understand why this was necessary, it is important to acknowledge the fact that a MIDI file is able to encode separately 128 pitch-instruments (also encoded as "non-drum instruments"). Moreover, the distribution of the instruments across the dataset was undoubtedly unequal (usually referred as a long-tail distribution). This shows that most pieces on the dataset only use very few instruments, consequently most of the "other instruments" occur very rarely. Even though this might be an aspect of music itself (some instrument being way more protagonist than other), the volume of data available and our computer capacity were not going to be enough for the model to generalize for each of the 128 pitch-instruments.

Because of the previous scenario exposed, we decided to make an abstraction step to transform the "instrument space" into a smaller one. Specifically, we mapped the original instruments into four families: drums, pianos, basses, and guitars. On this premise, each instrument of a song was mapped to either one of those four families, or ignored.

We chose these specific four families of instruments because of their broadly well known usage. The core instrumentation in rock and pop consists of bass, percussion (drums), guitars and keyboard instruments (it usually also includes the voice lead, but we decided to ignore it since it is not an "instrument" per se).

The grouping described above is to some extent supported by looking at the usage of instruments in the dataset, as it is shown in the instrument clustering performed. There we used pitch distributions to measure similarities between all the 128 pitch-instruments. We took each of the 128 possible pitches as a different feature or independent variable, and measured how present (weighted by note duration) each of them was across all the MIDI files in the dataset, for each pitch-instrument. With this, we performed a hierarchical clustering using euclidean distance, obtaining the cluster assignment shown in Figure 4. It is important to remark that we could not make the same analysis for drum instruments, since their "pitch" value doesn't correspond to the actual pitch, but instead to a different percussion (sub-)instrument. For simplicity, we decided to consolidate this group right away, since they already come separated in a MIDI file as a "drum track".

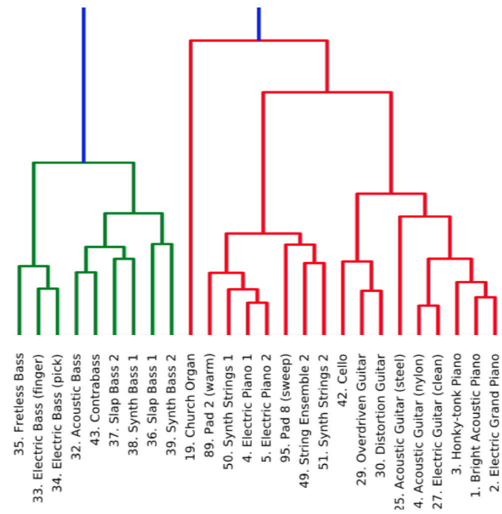

Figure 4. Snapshot of Instruments Clustering's Dendrogram

As can be seen in Figure 4 the bass family was really a defined group (they did not merge with other cluster until the final iterations of the algorithm). This was an interesting finding, since it matched perfectly with our prior intuitions or knowledge. This was on the other hand not the case for the guitar and piano families. As is noticeable in Figure 4, guitars and pianos have mainly the same pitch distribution, since the algorithm merges them in the same cluster in an early stage. This can explained by the fact that both families of instrument use the same pitch range (as opposed to the bass family, with a more narrow range). Acknowledging the finding of this messy cluster, we still decided to keep both instrument families separated, since note distribution is not the only criteria to "catalogue" an instrument. Guitars and keyboards differ widely in their respective "user interfaces", and therefore, so does their playing techniques (aspect not measured in this clustering). With all the previous said, we merged our prior knowledge with this clustering to arrive to the final instrument mapping shown in Figure 5.

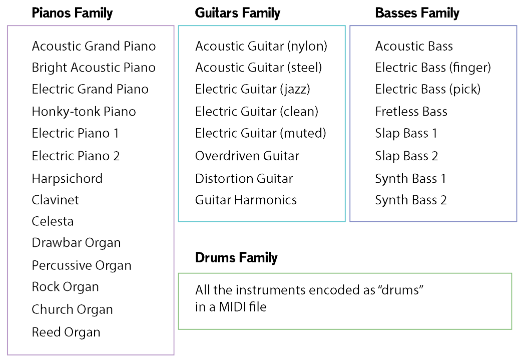

Figure 5. Assignment of instruments to families

For pieces with several instruments of the same family, the notes of related instruments were merged into a single track using the previous mapping. Even though a different technique of merging could have been applied (e.g. taking a random instrument per family instead of merging them all), we decided to use this one to be able to represent the instruments families "as a whole", instead of random subsamples of it. It is important to remark that all MIDIs that did not have at least one instrument per family were removed from the dataset. This was done to avoid feeding the model with empty instrument family tracks.

3.4 Encoding

In order to obtain a representation of the songs that could be used as an input to the neural network, we chose to quantize our input into a discrete time representation, following the existing literature on the field.

Figure 6 shows a visual illustration of the encoding process. On the left of that Figure, there is representation of a single track encoded in MIDI file (in our case, the track of one instrument family). The track consists of an arrangement of notes, sorted by their onset time. Each note has four basic features: pitch number (from 1 to 128), duration and onset instant of time (in seconds), velocity (from 1 to 128). In this visualization, notes are shown in a "piano roll" representation with the temporal component axis.

We used time steps of 0.25 seconds, which best preserved the input data without becoming too sparse. Given this, the encoding places each note in a time step if the onset of the note occurs in that interval. For every note which occurs in a given time step the pitch, duration, and velocity are encoded. Therefore, each row in this encoded song

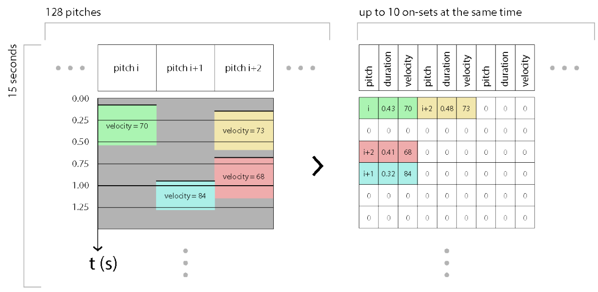

Figure 6. Transformation from MIDI to RNN input representation

corresponds to a certain time step, where the columns represent the three features of each note at that time step. We limited the total potential notes to ten per time step, in order to once again limit the sparsity of the data. This was more than enough for most songs, but since our objective was to use these models to explore the data rather than create the most interesting compositions, we tried to preserve as much as the original data as possible. The columns are filled from left to right, excluding all notes that exceeded the aforementioned limit. This encoding is capable of preserve intact three of the four features of a MIDI note (pitch, duration and velocity), and keeping an approximate version of the fourth one (onset time). Therefore, it was precise enough for the aims of this project.

3.5 Model Architecture

Artificial neural networks rely on a series of 'neurons' or units, each with an accompanying weight and bias, to transform input data into an accurate prediction of a piece of target data. The loss between the output of the neural network model and the target data is then calculated, and the weights and biases are adjusted to make the model more accurate in its predictions using a process called backpropagation. In this case we used our neural networks to map the input data of all four instrument tracks of a given genre to label data of the correct note prediction for particular instruments.

We specifically chose a recurrent neural network because of its unique nature designed to focus on sequential data. Unlike a traditional network, the recurrent units of this type each have an output at each time step and a separate interior state prediction that is passed between the cells of each time step. We picked the very popular long-short term memory cell over several other options because of its suitability to handling larger and more sparse input data. We also strongly considered the popular gated recurrent unit because they are faster to train, however, since GRU cells lack an output gate, they are less able to accurately differentiate relevant data in sparse datasets. Sequential models are intuitively the correct architecture for time sequence data like musical composition, and the literature surrounding the task of music generation strongly confirms this intuition.

Our model architecture features an embedding layer with a shape of 200x512x60, 2048 recurrent units using an LSTM cell, and a dense output layer of 200x1. One batch of input (60 rows of the prior explained encoding method) is the equivalent of 15 seconds of music, and at each step the model predicts its expectation of the next 15 second snippet of the song. We utilize a logits based output with 200 potential values, which is why we use an extra dimension of 200 in both the input and output layers. At each step the model receives the concatenated encoding of all four instruments from a given song in a given genre, but the loss is only determined in relation to one instrument during training. In this way, each model learns to optimize to output only one instrument. We used a learning rate of 0.005 with an ADAM optimizer [3] and sparse categorical cross entropy as our loss function.

With this model architecture defined, we trained 16 models (1 for each instrument for each genre) across the training set for four epochs each. The models were able to achieve relatively low loss values across all genres, partially due to how sparse the training and test data are. By the model initially learning just that most values were zero, the loss value drops dramatically almost immediately in training. However, it continued to drop as the model learned more complex features of the data, which we will elucidate further.

4. RESULTS

We were able to observe some broad distinctions in loss on the test set between various instruments, genres, and features. All future references to loss values refer specifically to loss relative to the entire test dataset, except in the case of more granular analysis relative to velocity and pitch, in which case loss is relative to a random subset of the test data. We found that the models for the pop rock genre were the most effective at generating convincing predictions. These models achieved a mean loss of 0.71, 0.67 for piano, 1.22 for guitar, 0.65 for drums, and 0.29 for bass. As you can see from Figures 9 and 10, ballad genre loss was also low across the board though slightly higher at a global mean of 0.76. These two genres were very similar, with most loss occurring in the pitch features for the melodic instruments (Fig. 11).

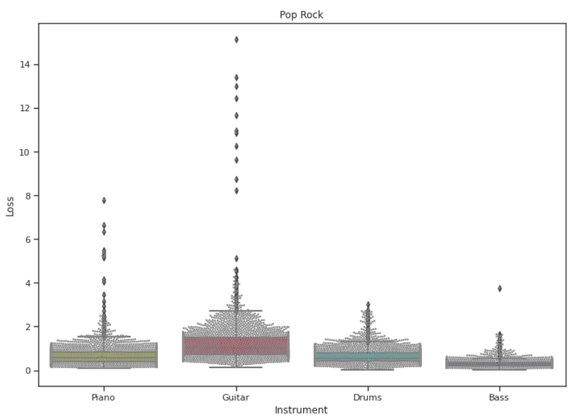

Figure 7. Pop Rock Genre Loss by Instrument

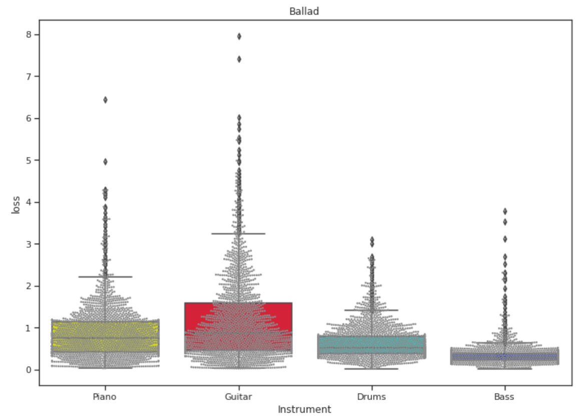

Figure 8. Ballad Genre Loss by Instrument

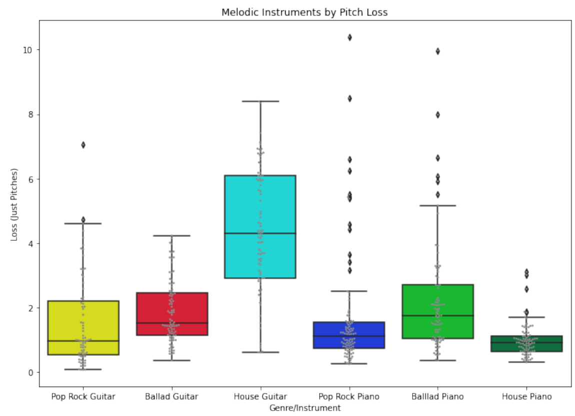

Figure 9. Pitch loss of Melodic Instruments (Piano, Guitar) by Genre

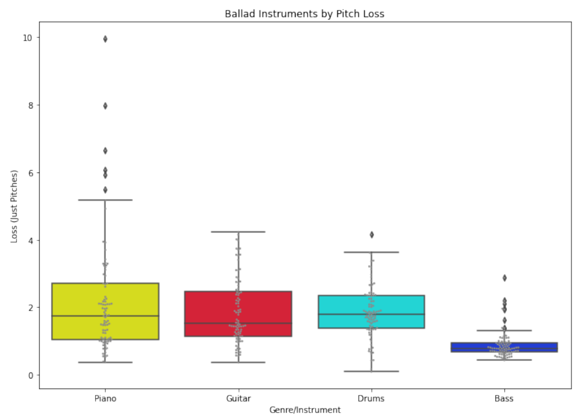

Figure 10. Ballad Genre Pitch Only Loss by Instrument

House had substantially worse loss in an inverse distribution across instruments compared to pop rock and ballad. In other words, while the loss was higher with the piano and guitar for pop rock and ballad, for house the loss was concentrated in the bass and drum instruments. Since the drums had the highest note incidence rate with an average of 369.21 (in comparison to 251.61 for guitar, 160.16 for piano, and 97.8 for bass) we theorized that this contributed to the higher loss. Since the loss for bass was higher than either guitar or piano with fewer notes, we can see that note incidence is likely not the main cause for differences in loss mean. In addition, in Figure 11 you can see the shape of the loss distribution is quite different for each other with drums being much more tightly clustered. You can also see two main clusters in the bass distribution, as far as these loss functions relate to incidence rate these may represent the 'grooves' in relation to the constant BPM of the drums. However, there was also a substantial difference in loss for the velocity feature for house music in the rhythmic instruments: bass and drums (Fig. 14). There was a larger variance in velocity for the house drums than any other genre or instrument. Velocity loss was proportionally higher than pitch and duration loss for house drums, showing that this increased loss rate was not a result of only higher note incidence rate but also the diverse composition across velocity. This is logical when the distinctive sounds of the genre are considered because it often features layered percussion elements at different velocities used to create more complex sounds for the rhythm.

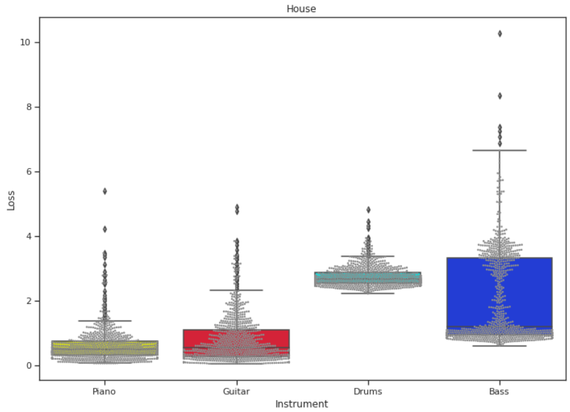

Figure 11. House Genre Loss by Instrument

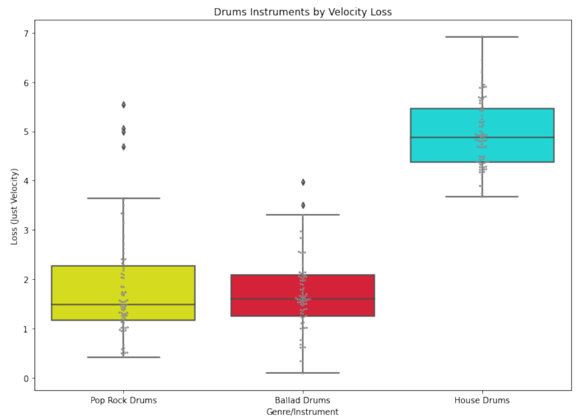

Figure 12. Drums Velocity Only Loss by Genre

5. DISCUSSION AND FUTURE WORK

We noticed several confounding factors in our work that made it difficult to achieve our goal. First was the generally sparse nature of the data, there were many time steps without notes and many time steps with only one or two notes out of ten possible notes. We considered condensing the input to encode fewer possible notes at each time step, and fewer time steps in general, to make it more dense, but decided against this because our aim was to use the models to elucidate the musical data, rather than produce the most realistic possible output. In order to learn this sort of sparse data more effectively, ideally it would require a deeper model with more embedding layers and substantially more units in each layer, larger datasets, and more training iterations to further fit the data. It would also be possible to increase accuracy without losing input information by measuring loss only in rows and columns that have a note onset in order to reduce the importance of empty cells. However, since silence itself can be considered to be a feature of music, this approach is limited. Despite our constrictions on the computation resources used for this project, we were still able to return low loss with demonstrated feature learning as the loss broadly decreased throughout iterations and epochs.

Another source of error was differing sizes of datasets by genre, where pop rock and ballad had 9914 and 7014 songs respectively, while house had only 3217. Though the length of songs is not constant, because many of the MIDI files were only parts of songs the average length was fairly consistent across genre. Though this is not ideal as it relates to the potential set of songs being less likely to describe the total variance of composition expressed in the house genre, we found that this size of data was still mostly sufficient for our smaller scale models. This was demonstrated by the largest decrease in loss occurring in the first epoch for each dataset regardless of size, so it was clear that despite the lower number of samples that the house models still were able to learn some of the features of the house music genre.

There are many possible possible paths forward to both improve this system's ability to generate coherent samples and explore the data more completely. One interesting area of inquiry might be to train models on all 3 genres at the same time to create a better baseline for the instruments and the effect of larger or smaller amounts of data. We might also look into analyzing more specific subsets of the data, such as the cluster of middling loss samples in the bass instrument for house music, to determine what structures are causing that change in loss value. We also would like to expand to model more genres in order to observe more links between broader style definitions.

6. CONCLUSION

In summation, we were able to successful train RNNs for each instrument of multiple genres of music using MIDI files passed through a data pipeline. We found that the melodic instruments (guitar and piano) carried more complexity in the ballad and pop rock genres, while the more rhythmic instruments like drums and bass were more difficult to learn for the house genre. Inside of those subcategories, the melodic instruments had higher loss in relation to pitch features, while the rhythmic instruments varied more strongly based on velocity. There is ample potential to expand this avenue of musicology research in the future.

7. REFERENCES

[1] T. Bertin-Mahieux et al. "Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)". In: 2011.

[2] D. Eck and J. Schmidhuber. "Finding temporal structure in music: blues improvisation with LSTM recurrent networks". In: *Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing*. 2002, pp. 747–756.

[3] D. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: *CoRR* abs/1412.6980 (2015).

[4] Tom Li, Antoni Chan, and Andy Chun. "Automatic Musical Pattern Feature Extraction Using Convolutional Neural Network". In: *Lecture Notes in Engineering and Computer Science* 2180 (Mar. 2010).

[5] Francois Pachet and Daniel Cazaly. "A taxonomy of musical genres." In: Jan. 2000, pp. 1238–1245.

[6] C. Raffel. "Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching". PhD thesis. COLUMBIA UNIVERSITY, 2016.

[7] B. Sturm. "A Simple Method to Determine if a Music Information Retrieval System is a "Horse"". In: *IEEE TRANSACTIONS ON MULTIMEDIA* 16.6 (2014).

[8] Y. Zhou et al. "BandNet: A Neural Network-based, Multi-Instrument Beatles-Style MIDI Music Composition Machine". In: *ISMIR*. 2019.