Jack Stanley

WRI 175

3 December 2019

Word Count: 3985

Cracking the Encode: The consequences of normalizing training data in AI music composition

Abstract

Machine learning algorithms are being applied to a diverse range of applications, maybe one of the most surprising being musical composition. This paper investigates how the encoding of training data affects stylistic components of music, specifically tempo and dynamics. The object of analysis is MuseNet, a recurrent LSTM model capable of mimicking music in a range of genres and composers. Focusing on compositions based on romantic composer Frédéric Chopin, analysis of MuseNet's compositions were compared to human performances of his works. This investigation finds that the AI-written music lacks diversity in both tempo and dynamic range, ultimately limiting the potential for stylistic and emotional aspects of music.

Introduction

Since the dawn of the Computer Age, developers have explored the idea of algorithms that can mimic human thought and creation. Human creativity and emotion are often thought of as traits that computers will never be able to exhibit, yet since the early 1960s, researchers have been attempting to prove this wrong by designing algorithms capable of yielding "creative" and human-like outputs. While their success has been arguable, decades of research has gone into these algorithms.

Music compositions have been around almost as long as civilization itself, making it an extremely developed and abstract form of art. Historical composers have become household names due to their ingenious creations. In addition to following complicated musical theory, music from the classical and romantic periods incorporates stylistic elements that invoke emotions in its listeners. Emotion can be portrayed through music in many ways, including the notes themselves, speeding up and slowing down (tempo changes), and changing the volume at which the music is played (dynamics). While all of these can be written through markings in sheet music, much of the expressiveness in tempo and dynamics is decided by the performer of the music. Composers sometimes leave little to no instruction on how to perform the more subtle interpretation of their music, a practice especially common in classical and romantic music. When performers incorporate these techniques into their playing, music begins to sound much more meaningful and emotional. Music's unique ability to invoke feelings in people makes it a valuable asset in many applications. For example, movies, video games, and even commercials spend large sums of money to have quality music composed to make their product favorable. Automating the process of musical composing could have an enormous impact on numerous industries, making it a decades-long studied topic.

The importance of emotion in music makes it quite difficult for computer algorithms to emulate the sound of a musical performance, a challenge that has led to hundreds of unique programs. Each of these algorithms is varied in what data is inputted, how the data is framed for the algorithm, and how the core training and creation is managed. Many studies have explored to what extent various algorithms are successful in making diverse and accurate chord progressions and pitches. Nadeem et al. discuss the effectiveness of different types of algorithms in producing unique music and the causes of multiple shortcomings in different algorithms, giving insight into the most modern and accurate types of algorithms (2019). Choi focuses more specifically on machine learning algorithms and how the data inputted in training sets affects the produced notes, introducing the important topic of encoding data. While these articles provide insight as to how different methods lead to accurate and varied melodies, what remains to be studied are the effects that encoding can have on more stylistic aspects of classical music. As machine learning continues to develop algorithms that are increasingly capable of mimicking human output, it is important to analyze what basic components of encoding lead to more accurate results. This investigation focuses on MuseNet, a transform neural network capable of producing very convincing music in a variety of genres, and how the encoding of MuseNet's training data affects the stylistic components of its compositions. By observing several of MuseNet's classical compositions and comparing them to real works of the same genre, it is evident that while the algorithm effectively incorporates some tempo changes into its compositions, they contain very narrow ranges of tempos and volume, leading to a limited communication of style and emotion.

Literature Review

Due to this investigation's focus on musical performance techniques appearing in AI composed music, it was necessary to refer to sources regarding both music and machine learning. Some of the sources are focused predominantly on the expressiveness of classical music performance while others mainly pertain to the encoding and outcomes of AI composers. By analyzing sources in these two separate categories, it was possible to understand what to look for from a musical theory standpoint in terms of style and expressiveness as well as what factors of encoding in the MuseNet algorithm may have led to these factors.

Deciding which elements of style and expressiveness are contained in MuseNet's compositions required first a strong understanding of what techniques professionals use to convey emotion when performing classical music. De Poli discusses the often-subtle deviations from the sheet music that performers make which make playing sound more human and emotional (2010). The article models these changes from a technical standpoint, attempting to describe logically why performers make these expressive interpretations. Understanding why human performers make stylistic choices provides a useful starting point when analyzing why an AI would (or wouldn't) make these same choices.

Other studies conducted in recent years have analyzed the use of numerous AI techniques to compose music. Nadeem et al. discuss how different types of algorithms, mainly Symbolic AI and machine learning, result in different levels of diversity and style due to their architecture (2019). They conclude that machine learning and LSTM networks like the one used in MuseNet ultimately tend to produce more diverse-sounding outputs and incorporate a wider variety of stylistic components due to their basis on already produced music rather than theoretical rules. LSTM networks are further

discussed in the context of AI composition by Choi et al. (2016). Although this article discusses drum tracks rather than piano music, it discusses many of the same issues related to the encoding of training data and how it effects long term structure and accuracy. Foster builds on these ideas in the context of MuseNet discussing how the use of an LSTM network effects the outcome of the MuseNet algorithm, describing some of its successes and shortcomings (2019).

Background

Many earlier music-writing algorithms rely on symbolic methods, which use manually inputted music theory rules and many layers of abstraction to build up a piece based on fundamentals. According to Nadeem et al., symbolic processes often "result in highly predictable music with a limited variety of styles in them," meaning the products lack diversity in terms of the actual melodies they produce. On the other hand, the article explains that machine learning algorithms can create patterns of notes that are more unique and diverse. What this investigation aims to discover is whether AI can produce varied structures in stylistic elements of music rather than simply their melodies, and therefore it follows that the object of analysis should be an AI based on machine learning neural networks. More specifically, LSTM algorithms (Long Short-Term Memory) are being readily adopted in the fields of natural language processing and other similar applications. LSTM networks are a form of transform networks in which during training, the algorithm studies patterns within a large number of preceding tokens before predicting the next. The MuseNet algorithm was chosen as it makes use of an LSTM neural network and represents some of the most recent methods of machine learning.

MuseNet was created by OpenAI, a San Francisco-based company aimed to produce AI algorithms that benefit society (OpenAI). OpenAI recently became a hot topic in the news due to their release of the GPT-2 algorithm, a recurrent neural network originally purposed to write realistic news articles about any given topic. While GPT-2 became famous for its natural language processing, the algorithms versatility allows it to be applied in many other fields, including music. MuseNet uses the same GPT-2 algorithm, except instead of using words as tokens in its training and generation, it is fed musical notes as tokens. In order to train the algorithm, first thousands of songs were collected and categorized by artist and genre. Then, the algorithm went through each song note-by-note, analyzing patterns in the data. In the case of MuseNet, the LSTM network examines patterns in the last 4,096 tokens, allowing it to exhibit "long-term structure and melodic structure across a range of styles" (Foster 2019). Once training is completed, MuseNet can generate music utilizing two main starting points: either it is given a short prompt of a few notes, or it begins from scratch. When the algorithm receives a prompt, it considers these beginning notes and then predicts what a logical next note would be. From there, it continues to predict the next note based on its own previous choices, accumulating more and more to base its next note on. If there is no prompt, the algorithm begins by choosing a random note and continuing from there. The process of generating notes can go on forever as the output is only based on what it wrote previously.

Before MuseNet can train, the data must be inputted in a way that makes patterns visible to the algorithm. MuseNet and most other Machine Learning music composers train from midi files, a format that saves pitch, volume, and time information rather than raw sound data. Using actual sound data would make training much more difficult as it could be hard for algorithms to pinpoint exactly which

notes are being hit at which time and how slight deviations in recorded pitch are connected. Midi files consist of a list of notes each with a start and end time, as well as a pitch and key velocity each represented by a number from 0 to 127. Figure 1 shows a visualization of a midi file. Blue lines further to the right represent notes that get played later in time, while their length represents how long the note lasts. The notes closer to the top represent higher pitches than those near the bottom. Finally, the color of the lines corresponds to the volume at which the note was played. Darker blues represent quiet notes while light blue represents louder notes.

Figure 1: A visual representation of midi data. The blue lines represent notes, the x-axis representing time and the y axis representing pitch. The darker colored lines represent quieter notes, while the brighter liens represent louder notes.

As illustrated by the image above, the data in a midi file is definite and concise, unlike traditional audio files. This makes midi files a great way to train algorithms because every note is easily represented by a set of numbers in a relatively small range, making patterns more apparent. However, inputting raw midi data into a training algorithm still isn't enough. In order to expedite the training process and influence it to find the important patterns, it is first necessary to encode the data in a helpful way. By manipulating the data to make some aspects more similar, the algorithm can more easily compare separate pieces of music and make connections between them. On the other hand, normalizing the data in this way can also lead to a lack of diversity in the final products because the training only contains music in one key. Thus, encoding data is extremely important to the success of a machine learning algorithm because it has the potential to completely change the way that data is processed and replicated. When studying the presence of stylistic choices in AI composed music, it is important to focus on encoding because the decisions to normalize data related to tempo or dynamics can have noticeable effects on the use of style. Some algorithms normalize tempo so that all songs are on the same time scale, while others use absolute time, potentially sacrificing the success of training to some extent.

Before the effects of encoding on musical style can be discussed, it is first crucial to understand what aspects of classical music performance are considered stylistic and contribute to the emotion in a piece. According to De Poli, expressiveness is a concept that includes "the means used by the performer to convey the composer's message and to his/her own contribution to enrich the musical message," suggesting that expressiveness is a product of both the composer's original message and the performer's interpretation. Composers don't typically mark every single tempo change or dynamic accent that they intend, instead leaving these stylistic choices up to the player. De Poli cites the main measurable deviations from sheet music as tempo, dynamics and articulation. These three aspects are the technical foundation of musical style, making them perfect points of interest for this investigation.

Methodology

The stylistic choices that this investigation analyzes are tempo and dynamics. These were chosen because of the three foundational stylistic components listed by De Poli, these two were the most

clearly defined in the midi files. Articulation was not included because it relies on many factors that cannot be quantized due to their reliance on context. Since MuseNet can compose music in a variety of genres and composers, it was necessary to narrow the focus of the investigation to a single category in order to make accurate comparisons. Romantic composer Frédéric Chopin was chosen because multiple MuseNet samples are written in his style and new music in his style can be generated from the MuseNet website. Chopin is also from the Romantic period, a subset of the Classical music period, meaning that his compositions adhere to much of the theory surrounding classical music.

In order to understand Chopin's style, Higgins's article "Tempo and Character in Chopin" was utilized to understand the typical tempo and stylistic structures used in Chopin's compositions (1973). By observing what stylistic markings Chopin includes in his compositions, we can conclude what types of stylistic choices must have been a consequence of the performer. According to Higgins, Chopin only sometimes included tempo markings in his music, often instead just using words to describe the tempo at different points. This means that in many cases, the performance tempo and deviations in it are controlled primarily by the performer. The same is true in dynamics, as Higgins mentions multiple pieces written by Chopin that contain no dynamic markings. Therefore, the main method by which music was analyzed was by looking for variations in tempo and dynamics that indicate stylistic choices in both performances of Chopin's music and MuseNet's compositions.

The next step in analyzing the music was to find a source where midi files could be collected for both composers. MuseNet allows visitors to their website to generate music in the style of Chopin and download the newly created song as a midi file. Each song was generated from scratch, meaning that it was not based on the beginning of another song. The Chopin pieces were obtained from *Classical Archives*, one of the main sources that MuseNet gathered its training data from.

Finally, to extract the useful data from the collected midi files, two simple python scripts were utilized—one for tempo and one for dynamics. The first script recorded the elapsed time between key presses while we manually tapped to the beat of the song. This not only showed the tempo over general periods of the music, but also showed moments at which the tempo suddenly sped up or slowed down or if there was a short pause. To measure the dynamics of a song, a second script looked through each note in all the songs that were collected, recording the key velocity (a component of every note in a midi file). Key velocities range from 0 to 127, with higher numbers representing louder notes. The values were gathered from 128 Chopin performances and 50 MuseNet compositions.

Findings

Though analysis of the tempo data collected, this investigation found that despite the adjusted time scale used in the encoding of the data, the music produced contained many of the important stylistic aspects related to tempo. MuseNet's website explains that the before the data is used for training, the tempo is first augmented, "effectively slightly slowing or speeding up the pieces" (Payne 2019). This means that in order to make pieces with different tempos more comparable, the tempo is normalized towards an average tempo. Although this makes patterns more easily recognized by the algorithm, it is also essentially destroying information about what tempos the music is written at. Since many of Chopin's compositions don't include tempo markings, choosing what tempo to play at becomes an important stylistic approach, meaning normalizing the tempo during the encoding process could potentially flatten the stylistic choices mimicked by the algorithm. The effects of this tempo change are

depicted in figure 2. Notice that in each example, the pitches of the notes are not changed. Instead, the length and spacing between the notes is scaled to either speed up or slow down the music, normalizing it toward a central tempo. This means that some of the information related to tempo is completely lost during the encoding process.

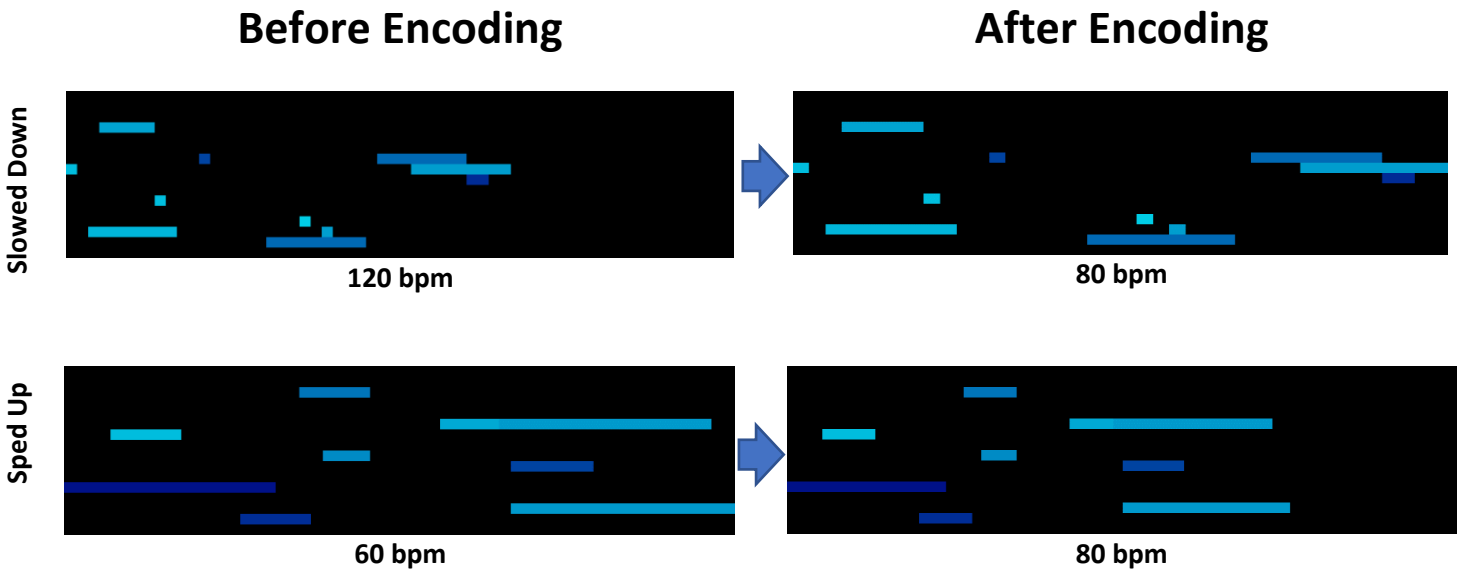

Figure 2: The diagrams above show the data in midi files before and after they are encoded by an algorithm similar to that used by MuseNet. The data in the diagrams has been simplified to make it easier to read and the changes emphasized to make them more visible. This diagram also only represents the changes based on the tempo of the music, not the dynamics. For the sake of this visualization, theoretical tempos were chosen, the "central" tempo being 80 beats per minute (bpm).

Since the tempos of all the inputted songs end up being similar after being encoded, it follows that MuseNet's compositions might have a less varied range of tempos as well. However, it can also be expected that MuseNet will still pick up on smaller tempo changes throughout the song, such as ritardando and accelerando, where performers slightly slow or speed up the tempo at a point to create an emphasis. In order to test whether MuseNet successfully includes these stylistic strategies in its music, the tempo throughout Chopin's "Waltz in C# Minor" and one of the MuseNet samples were tracked and recorded using a python script. The script required that the beats be manually tapped by a person as they listened to a song, meaning that not every beat was tapped consistently and there were constant fluctuations in the measured tempo. To make the graphs easier to read, a red line representing the approximate average tempo over that time was added. Gray boxes represent slight pauses in the music added in during performances.

First, the overall structure of each piece was analyzed, looking at major patterns and changes in tempo. In the human performance of Chopin's music, it is apparent that the tempo makes a few sudden changes as well as gradually speeds up or slows down in a few places (Figure 2). These same patterns can be observed in MuseNet's composition, including both sudden and gradual changes as well (Figure 3). It is also important to note that in both compositions begin at one tempo, change drastically in the middle, and then return to the same tempo in the end. This demonstrates MuseNet's analysis and recreation of long-term motifs in music, as suggested by Foster. The grey boxes extending upwards and downwards highlight moments in which the song briefly paused or skipped a beat. Both compositions contain multiple of these stylistic techniques, once again showing that MuseNet does pick up on tempo

deviations despite normalization. Together, these similarities suggest that the normalization of does not affect the tempo changes relative to the original tempo. This is to be expected because even if the tempo of a song is adjusted so that the beginning is near a certain value, the tempo changes will still occur in the same proportion relative to the initial tempo.

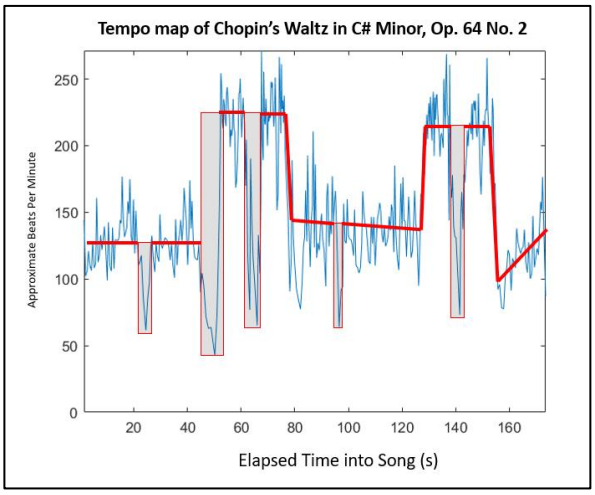

Figure 3: A graph of the tempo in beats per minute over time of Chopin's Waltz in C# Minor. The red line represents the approximate tempo during that time while the gray boxes represent slight pauses during performance.

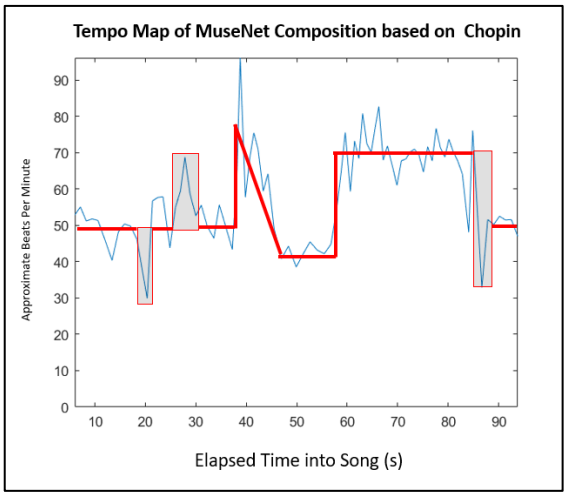

Figure 4: A graph of the tempo in beats per minute over time in one of MuseNet's compositions based on Chopin. The Red line represents the approximate tempo during that time. The gray boxes below the line represent slight pauses while the gray boxes above the line represent times at which the song skipped a beat.

In addition to tempo changes, the overall tempo at which a piece is played can have large effects on expressiveness as well. While the tempo changes MuseNet's compositions seem to be modeled accurately, the range of tempos that the music is written in is quite limited. This makes sense based on the methods of encoding used. All the data being inputted into the training algorithm is at a similar tempo due to the time normalization, meaning that the neural network sees almost no variance in the tempos of different songs. This suggests that the final products will likely be lacking in diversity of tempo. In order to measure and visualize the levels of variance in tempo, the approximate average tempo of each Chopin and MuseNet composition was measured manually and compared in a box and whisker plot (Figure 5). The blue data represents Chopin's music while the orange data represents MuseNet. The MuseNet data clearly covers a much smaller range of tempos, meaning that it is less diverse. This suggests that the normalization of the tempo data when it is encoded leads to a lack of diversity in the tempos of created music, ultimately resulting in a flatter style.

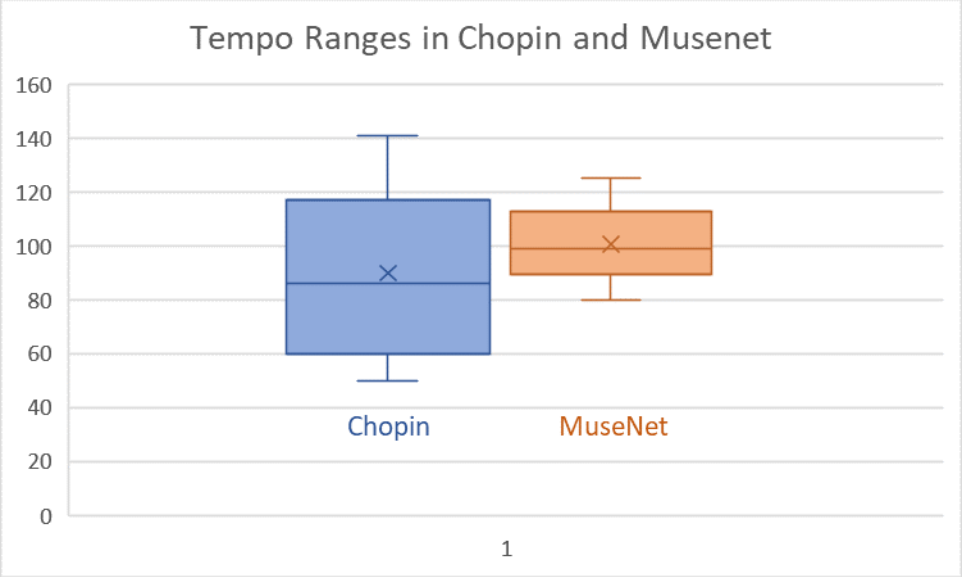

Figure 5: A box and whisker plot made from the approximate average tempos in beats per minute of a range of Chopin and MuseNet compositions. The "x" in each box represents the mean tempo.

The other aspect of style being investigated is dynamics. Like the tempo, the volume of the notes is augmented before training begins, "turning up or turning down the overall volumes of the various samples" until it is closer to a central value (Payne 2019). This augmentation of the data is meant to make patterns more recognizable and prevent extreme fluctuations in volume, but it can eliminate important stylistic choices as a side-effect. In this case, the algorithm is missing information about dynamics, which contributes greatly to the emotion portrayed in a song. One study found that nine-out-of-ten emotional categories were "significantly affected by dynamics," suggesting that it is extremely important to mimic dynamic choices to create a successful algorithm (Chao 2016). The figure below illustrates how the encoding of the notes leads to less diversity in the training set data. Notice that the while the two midi files on the left are quite different in their range of volumes whereas the images on the right side consist of very similar ranges of volume. This demonstrates that when each song is encoded, diversity in dynamic range is virtually erased in the training set.

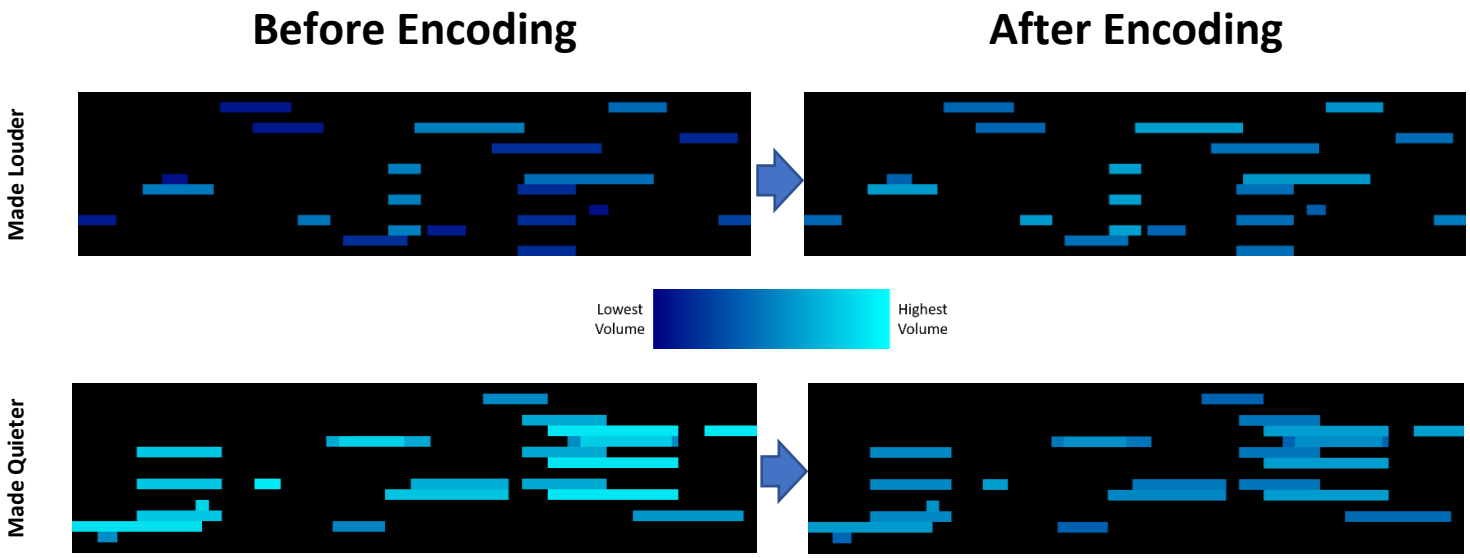

Figure 6: The diagrams above illustrate how the encoding dulls the volume information in the songs when they are encoded. Once again, the modification to the data has been amplified to make it more visible and does not represent exactly how MuseNet alters the dynamic information during its encoding phase.

A comparison of the dynamic ranges in Chopin and MuseNet's works is shown in figure 7, another box and whisker plot. The data included in the chart is the key velocities of all the notes in every Chopin and MuseNet song being studied. As predicted by the analysis of the encoding, the range of volumes used in MuseNet compositions is significantly smaller than the range in Chopin's works. This means that as a result of the normalization of the volume, MuseNet's compositions are lacking in many stylistic techniques relating to dynamics.

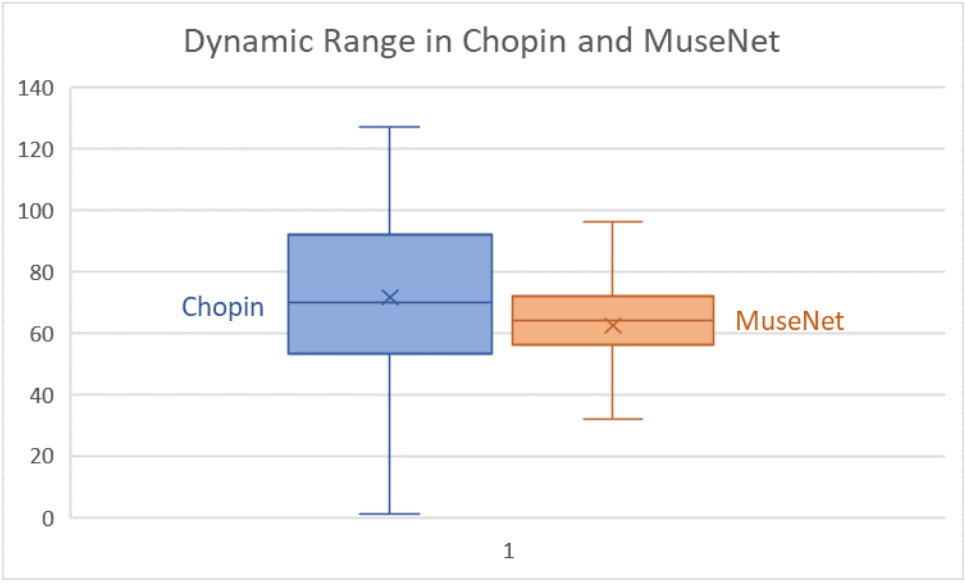

Figure 7: A box and whisker plot made from the velocities of over 250,000 notes in Chopin and MuseNet compositions. The x in each box represents the mean tempo.

Conclusion

Ultimately, the encoding of MuseNet's training data limits the extent of stylistic components relating to both tempo and dynamics. Although MuseNet successfully mimics tempo changes relative to the beginning of the tempo, it lacks diversity in the main tempo that its pieces are written at and produces notes whose volume is within a narrow range. The distributions of the average tempos as well as the volumes of the notes in MuseNet's compositions were significantly less spread-out than those found in performances of Chopin's music. MuseNet's Machine learning approach to music composition surpasses the diversity in notes produced by symbolic methods, yet it lacks the ability to accurately mimic the style and emotion that human players exhibit when they play. As an abstract form of art, much of the meaning and importance of music comes from the emotion portrayed in its performance, and thus MuseNet does not create truly diverse and meaningful music despite its unique melodies.

While this investigation analyzed tempo and dynamics in MuseNet's compositions, further research could investigate how encoding affects articulation, the third and final aspect of musical style suggested by De Poli. Also, more research could be done as to how altering the encoding process changes results with respect to both the general structure of generated music as well as its stylistic aspects. Understanding which elements of music are important to retain in encoding could lead to machine learning algorithms that make very convincing and styled compositions. AI-written music has the potential to impact the arts, businesses, and paves the way for numerous other applications of generative machine learning. By understanding how encoding effects music compositions, it is possible to draw conclusions about encoding that apply to the entire field of machine learning.

Acknowledgements

First, I want to thank Dr. Penman for his support and feedback regarding this essay as well as all his instruction throughout this semester. Next, I want to thank my dad for talking through some of the brainstorming of my essay with me and giving me an idea of the success of some older composing AIs from his college years. Finally, I want to thank Andy and Julio for their help during our D3 conference and the rest of the class for all of the feedback on my writing during this semester.

References

De Poli, G. (2004). Methodologies for Expressiveness Modelling of and for Music Performance. *Journal of New Music Research*, *33*(3), 189–202. doi: 10.1080/0929821042000317796

Chau, C.-J., Mo, R., & Horner, A. (2016). The Emotional Characteristics of Piano Sounds with Different Pitch and Dynamics. *Journal of the Audio Engineering Society*, *64*(11), 918–932. doi: 10.17743/jaes.2016.0049

Choi, K., Fazekas, G., & Sandler, M. (2016). Text-based LSTM networks for Automatic Music Composition. *Conference on Computer Simulation of Musical Creativity*.

Foster, D. (2019). The Future of Generative Modeling. In *Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play* (pp. 274–289). Beijing ; Boston ; Farnham ; Sebastopol ; Tokyo: O'Reilly.

Higgins, T. (1973). Tempo And Character In Chopin. *The Musical Quarterly*, *LIX*(1), 106–120. doi: 10.1093/mq/lix.1.106

Lopez-Rincon, O., Starostenko, O., & Martin, G. A.-S. (2018). Algoritmic music composition based on artificial intelligence: A survey. *2018 International Conference on Electronics, Communications and Computers (CONIELECOMP)*. doi: 10.1109/conielecomp.2018.8327197

Nadeem, M., Tagle, A., & Sitsabesan, S. (2019). Let's Make Some Music. *Internatinal Comference on Electronics, Information, and Communication*.

OpenAI. (n.d.). About OpenAI. Retrieved November 30, 2019, from https://openai.com/about/.

Payne, C. (2019, October 10). MuseNet. Retrieved December 2, 2019, from https://openai.com/blog/musenet/.