Conor Bailey (cpb32)

Using a Deep Neural Network to Produce Novel Music!

Supervisor: Dominique Chu

UNIVERSITY OF KENT

ACCESS TO A MASTER'S DEGREE OR

POSTGRADUATE DIPLOMA DISSERTATION

In accordance with the Regulations, I hereby confirm that I shall permit general access to my dissertation at the discretion of the University Librarian. I agree that copies of my dissertation may be made for Libraries and research workers on the understanding that no publication in any form is made of the contents without my permission.

Notes for Candidates

1 Where the examiners consider the dissertation to be of distinction standard, one copy will be deposited in the University Library.

2 The copy sent to the Library becomes the property of the University Library. the copyright in its contents remains with the candidate. A duplicated sheet is pasted into the front of every thesis or dissertation deposited in the Library. The wording on the sheet is:

> "I undertake not to use any matter contained in this thesis for publication in any form without the prior knowledge of the author."

Every reader of the dissertation must sign and date this sheet.

3 The University has the right to publish the title of the dissertation and the abstract and to authorise others to do so.

.....

SIGNATURE **DATE**

10/09/2022

.....

FULL NAMES

Conor Bailey

CERTIFICATE ON SUBMISSION OF DISSERTATION

I certify that:

1 I have read the University Degree Regulations under which this submission is made;

2 In so far as the dissertation involves any collaborative research, the extent of this collaboration has been clearly indicated; and that any material which has been previously presented and accepted for the award of an academic qualification at this University or elsewhere has also been clearly identified in the dissertation.

.....

SIGNATURE **DATE**

10/09/2022

Table of Contents

Abstract

The analysis of different data collection techniques that can be used to aid the production of novel music when using a Wave Net deep neural network structure. The project looks into what information can be gathered from either MP3, WAV, or MIDI file formats and how a deep neural network can use this data to produce music on its own based on patterns it identifies.

Introduction

The use of different types of Deep Neural Networks has been prevalent within different industries to better understand the data being used to obtain certain results. Within the music industry many attempts have been made to try and figure out what types of data can be gathered in order to produce a completely new song. However, with artificially producing music it has proven difficult for many people, believed to be due to the creative nature of the artform and how a human mind is typically required as a subject to judge and involve themselves within the music being produced.

Within this project the idea is to try and see what different datasets could be used to produce music, both new and already tested. Although it may be difficult to reproduce songs that contain an element of human passion and creativity, it could prove interesting to see what can be learnt from a set of music and how they can be interpreted. There are many different ways of extracting data from songs and these have been used for many different things. Firstly there's WAV files which are the most expansive with more data to be extracted, making it great for editing and mixing music. MP3 files are more compact as they have been compressed so contain less data, however, in computer learning these files have proved useful for music classification. Finally, MIDI files which comprises mostly of instruments and the notes they play like sheet music, which is the most beneficial to machine learning as play patterns are better understood. All can provide their own insights into a song and what data can be retrieved, and using different tools this data can be interpreted for different uses.

Using a dataset comprised of organised music in a particular datatype, the idea is to find patterns using a Deep Neural Network to see what can be produced. Within this project a comparison would be made based on what patterns a network could find based on the datatype, found either through my own discoveries or comparisons found by previous attempts within the field. One of these previous attempts this project is based on is Jukebox, and Neural Network developed by DeepMind to produce MP3 music based on a dataset organised by genre and artist. Another is Wave Net, originally used for speech interpretation and extraction using raw sound files, however, through testing MIDI files also prove effective.

Overall, the aim of this project is to compare the differences between different music files and how they can be interpreted by a Deep Neural Network to produce something new. With inspiration from previous attempts a better understanding should be formed with the goal of finding the best extraction of data when it comes to learning and reproducing music.

Problem Description

So far using Neural Networks to handle musical data has proved useful within the field when it comes to music classification. A prime example of this in use is through Spotify and how they are able to automatically detect the genre and mood of a song based on data previously processed to a very accurate degree[Jones, 2022]. This classification is done through the analysis of frequencies and trends that each genre has, based in meta data such as instruments used and spikes in certain sound data. Music classification is currently the leading use of music data when it comes to Machine Learning, mainly due to its usefulness when it comes to understanding customers and their trends when choosing music.

To produce music using AI is more of an experimental idea to expand on the capabilities of the field and see what can be learnt by computers. As music is more of a human artform it is hard to reproduce this process in a convincing manner, although attempts show how far the field of Machine Learning has come. There has also been a lack of research in the field when compared to similar topics like photo generation, where recently DeepMind has created the WALL-E application that uses Deep Learning to produce original photos based on a description given by the user. With similar breakthroughs in regards to music there could be a few interesting uses for such a product.

Firstly, within the music industry it is common to hear about artists having a creative block, making it hard to produce any new material. With the use of such a product it could be useful for an artist to gain inspiration based on what can be interpreted by their already established style. Although this would mainly be regarded as a fun tool too use, the use of which could still help. Another idea is that for new artists who haven't established a reputation may find such a tool useful for producing artificial beats or tunes to aid their development. However, with these uses its important to note that these are merely to help aid people or for fun, as music is human in nature it shouldn't be thought that artificial music generation would be something to replace artistic talent.

In total, the main interest in producing music through the use of such algorithms is to understand patterns in the data as well as extend the capabilities of the Machine Learning field. With current research being beneficial to the industry in classifying music and understanding trends from music data and the users listening to it, the expansion of research may help provide more beneficial results.

Technology Review

Currently, within the field of Machine Learning, there are two interesting approaches to the computational understanding of sound these being Jukebox made by OpenAI and Wave Net by Deep Mind. These provide different approaches to a deep neural network structure that can interpret raw audio datasets. Wave Net was the first example of such technology being produced, with the network structure being made public by Deep Mind in September 2016. Whereas Jukebox was made as its own adaptation to the same problem with its program being released in April 2020. Jukebox had inspiration from the work done within Wave Net as the problem remained the same, but an expansion was made to the technique. Within this small space of time, the two approaches were able to open a wide door with the idea of AI-produced music technology.

Comparison

Research within both of these projects acknowledged the challenges of using raw audio data, due to the density of the data and the computational power required. Using the Jukebox program the dataset is encoded before going through the model, to compress the file into manageable chunks before decoding them back to a better-structured state. Whereas with the Wave Net method, unless you add a similar system, the model is designed to handle the raw audio itself. This makes sense with smaller sample sizes that Wave Net focused on, as the opposite was the case for Jukebox which was trying to recreate samples larger than 30 seconds. These different approaches make sense with the data that each model was looking to handle, both proving effective in their respective study.

Present in both of the deep neural network models for the approaches was 1-dimensional casual convolution. Following a similar process to that of image processing, this feature makes sense due to the number of timesteps required within raw audio. Jukebox expanded on this idea by providing a VQ-VAE, which compresses the raw audio into lower dimensional space with the input being encoded and mapped to a sequence of tokens using a codebook before decoding the input layer into the model. Both of the approaches use a similar method, however, Jukebox added a VQ-VAE method to better appropriate the model to raw audio with larger amounts of data being compressed before a sequence of inputs are processed by the model's convolutional layers.

As previously mentioned the two approaches were using different datasets for their models to learn from. With Jukebox a larger database of music was used with around 1.2 million songs from various artists, with metadata of the respective artist, genre, and mood. With the Wave Net model the size of the dataset wasn't stated, however, it was consisting of piano rolls from different artists. With the large dataset of Jukebox, the metadata of artist and genre played an important role in the models learning, whereas Wave Net didn't focus on such things. The file types were also different, with Wave Net opting for MIDI files rather than MP3 like Jukebox. Wave Net was designed to handle raw audio like MP3, however, this was mainly reserved to its use for voice recognition and producing. Another note with the datasets, is that Jukebox focused on being able to use samples of 44.1kHz, whereas Wave Net was considerably less with 16kHz. This is an arbitrary difference with no impact on the functionality of each model, however, it does prove a difference in approach and goals.

Analysis

The results produced by both of the approaches were both positive in regards to what they were trying to accomplish. Jukebox has become a program that can produce a range of different musical samples based on any user's suggestion of either genre, artist, or mood. Being one of the largest datasets used for this purpose, it has the ability to reproduce such samples that can range from anywhere from 30 seconds to 4 minutes. On the other hand, Wave Net for the studies done by Deep Mind it was able to produce 30-second samples of MIDI filed piano rolls. For the use of AI-generated music, both of these results from the approaches had a huge impact on the field of study. However, as Wave Net had other focuses in mind, Deep Mind was able to use the model for voice processing and with the learning of text sequences, they were able to generate automated sentences with a voice also created by the system. This was a huge breakthrough for audio generation.

Concluding Remarks

Although both of the models had differences in goals and results, their approaches remained fairly similar and revolutionary in their own rights. Wave Net was the first model to be able to implement a deep neural network approach to understand raw sound, with music production being in the form of MIDI files. Jukebox later extended the approach, incorporating a VQ-VAE method to their approach of the model in an effort to be able to condense larger files before going through the network. This additional input was able to extend what was a 30-second sample size, to what could be up to 4 minutes. Overall, both perform effectively and have been able to show what can be accomplished by a deep neural network when interpreting musical datasets.

Approach

When getting into this project there was a range of approaches that needed to be considered for different aspects of the process. Research played a heavy role, where different techniques and resources needed to be considered to see what would best help towards the goals and aims set. With the different techniques considered for the project, there were different approaches undertaken to best experiment with the data and other resources being tested. To achieve the goals of the project the best approach needed to be adopted in order to ensure a smooth process.

Research

Online there is a range of papers and articles about different techniques used when it comes to the collection and processing of raw audio data. Through a selective research approach, focusing only on articles that used keywords and technologies relevant to the project, a summary of resources was collected. In making sure the data and research were approved, it was important to focus on sites with verification and some that may have open source files as proof of concept.

When starting the project, research was focused on articles that related to products like Jukebox and Wave Net. As these were established products within the industry, they provided a good starting point for research as more knowledge could be formed on technologies and approaches mentioned in their papers. Through this, the research began to yield results in terms of what datasets could be used for such purposes, as well as how to interpret this data. Using an experimental research approach from here when building the product of the project, different research avenues were taken as the data was interpreted in different ways. This meant that when looking at the deep neural network structure the data collected would need to fit certain requirements. When researching different approaches, a testing approach needed to be adopted to ensure the product would work effectively.

Collection of Datasets

Different datasets were gathered throughout the project due to the different purposes they may have within the system. A similar approach was taken with the storage of the datasets within the program files, however, the way they were gathered was different. With music files, it was important to gather the data with caution and legally, as a license may be required depending on the rights of the artists and record companies. Although for personal consumption and use the licensing rights are a bit more lenient than the public display of said music. As the project was for analysis and not public display, the gathering of songs was a little easier but caution was still required.

For the MP3 and WAV music file data collection, to make sure no issues were encountered, the files were purchased beforehand. This meant that the license to the song was already obtained during the purchase of the music, covering the use in which the data was purposed. This meant that the music used was large raw files, with both WAV and MP3 copies of the same songs. Although, towards the end of the project the only files used were WAV due to their usefulness and to meet the size requirements of the final product.

With the MIDI files used within the project, more caution and research were needed, due to these types of files being limited online and different in nature from that of WAV and MP3. As MIDI files can be used in different formats based on the production of the song and the instruments and notes used.

This meant some datasets lacked information relevant to the purposes of the project or were formatted in a manner in which the data retrieval may be difficult. With the dataset used for the project a site that provided license-free MIDI files was used. Different datasets were tested to see what fit the project needs best, the final result being MIDI files that contained one instrument's data for a piece of music. This provided a full dataset of musical data that can be used within the project to test different approaches to data collection and use.

Current Resources

In the field of automated music analysis and deep learning, there is a range of different resources for Python which were researched for this project. Within the articles previously mentioned in the research section of the project, many different techniques and resources were used for a variety of purposes. Using these as a basis for the project, the best approaches were found for the different aspects of the project and what needed to be achieved.

One of the first steps of the project, when the dataset was established, was to find and use a Python library that can interpret raw audio files. With this task there was a range of different resources, most of which would've been able to accomplish the task, however, a few stood out due to mentions in articles read. One of these is Librosa, a Python library used for the deconstruction of raw audio files for musical analysis and editing. This resource had the beneficial feature of being able to use a raw song file structure to generate new files that can be saved in either MP3 or WAV format. Testing was run on this library to see what data can be collected using this approach, finding the ability to generate visual samples of songs for a better understanding of the files at hand.

Another library was also discovered for musical analysis with MIDI files, this was Music21. This library was a bit more basic and less well-known than that of Librosa, however, its purpose was different and still relevant to the project. With Music21 information was able to be extracted from MIDI files, such as what instruments were used and what notes and chords they each played. Not only was the ability to retrieve this data of a MIDI file accessible, but new note and chord data created could be used to produce a new MIDI file that can be saved as its own file.

During initial research a technology found was Wave Net for the deep neural network development, which was the approach used within this project. As this was a pre-established approach to developing a neural network for a similar purpose to the project at hand, research was needed into how it operated with tests being run to change the structure to suit the data being used. The first step in using this network approach is the training in which the samples of raw audio or MIDI information are given to the network based on data gathered previously. The network would then use these as a basis for predicting the next timestep's frequency or note in terms of MIDI files. After the training of the network, it then goes through the inference phase which starts with the processing of the audio data given to it by the training, selecting a random sample array as a starting point. The second step is to predict the output sequence of the sample based on what was produced for the sample by the testing phase. After this, it chooses the most probable note or frequency based on all the samples before creating an array for a sample of its own. The fourth step is the deletion of the first element, with the passing of the appended array made in the last step as the input to then repeat the process from the second step of the inference phase. As the network runs, the finalised product should then be transferred and created into a newly generated sample. This approach was found effective in research so the same approach was used to test the network structure before alterations for the datasets used in this project.

Work Done

There were several different components involved in the making of this project, covering different aspects of the development cycle. Firstly a dataset needed to be created and organised in a manner that makes sense for an easy retrieval of samples. With this dataset the music needed to be interpreted by the system through the retrieval of data, it was also necessary to make sure the system wasn't overloaded with unnecessary information at this stage. Then finally the Deep Neural Network can be established, using the data previously prepared to understand and find patterns before producing its own samples.

Dataset

When creating the dataset for the project one of the main goals was to create a file structure that would make it easy for a user to add their own music for the network to learn from. As such the file structure was set out so that the files were organised by artist and album. The hope for the product was that the network can learn from all songs of an artist and certain albums separately.

As the project progressed different datasets were implemented, due to changes in datatype of the project. The main structure was retained when moving from MP3 to WAV, this is due to similarities in the datatypes and how they're typically structured. These files were whole songs of various artists and albums, which were broken into segments of 30 or 10 seconds using python script. However, when it came to MIDI files the same structure could've been implemented but as it was used alongside WAV files for an experimental cause, a dataset consisting of Beethoven symphonies was labelled by just song name.

Understanding the Data

There are many python libraries to choose from when it comes to playing and interpreting music, the two chosen for this project are Librosa and Music21. Both of these libraries made it possible to deconstruct and produce music in different ways as well as interpret the songs as needed. Librosa worked best on raw music data like WAV and MP3, being able to produce graphs for the file at hand. For MIDI files Music21 worked better as it was able to gather instrumental data within a file, being able to individually gather what notes are played for a specific instrument. More data could also be gathered but during this project these were the main focus.

Using Librosa there are a few different ways you are able to view audio files depending on the information the user is concerned with. With this library you are able to generate different graphs that can display different aspects of a track, with the ability to manipulate them slightly for music mixing if necessary. These graphs typically show frequency or amplitudes in a variety of forms for the duration of the song. Within the python code you can retrieve and edit the sample rate of the song file, as well as retrieve the song data which is stored as a array but can reveal a series of data shown through these graphs.

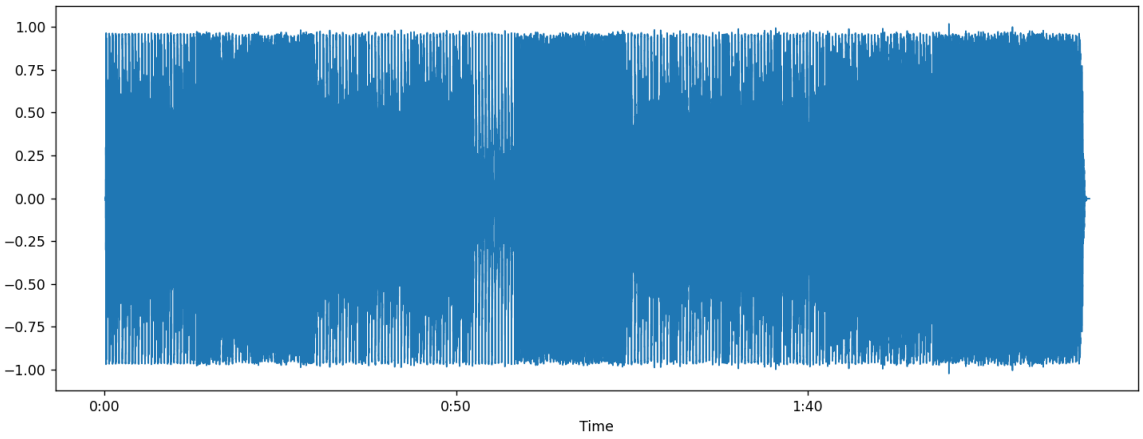

Figure 1 - Standard Spectrogram

The above spectrogram shows the amplitude of the song throughout the song. Each node within this spectrogram is the amplitude at the time stamp dictated by the sample rate, which can be changed if necessary within the code. The sample rate used for these examples is 22kHz for 8-bit sampling, the standard for typical WAV files is 44kHz at 16-bit. Although, as seen above the amplitude ranges from -1 to 1, varying frequently throughout the song. This graph was gathered from both a WAV and an MP3 file, as they were the same file the data obviously didn't change.

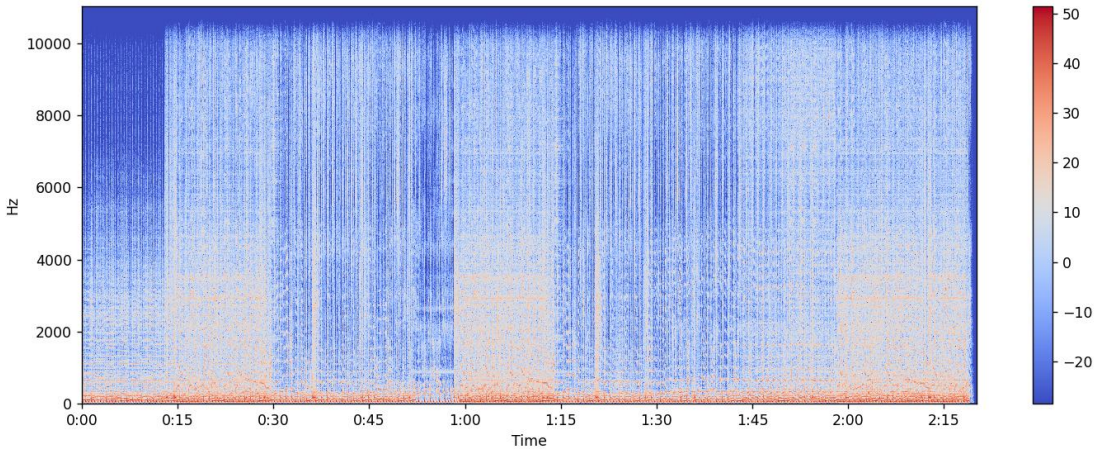

Figure 2 - STFT Spectrogram

With a Short-term Fourier Transform (SFTF) diagram the frequencies within the song are isolated based on strength during a certain point in the song. The strength of the frequencies within this graph dictate the loudness of the certain sounds at the points indicated. When creating this graph using the Librosa library there is already a function to apply a Fourier Transform with the sound data being the only requirement. As this function is usually fairly mathematical and better understood with greater knowledge of sound manipulation, Librosa makes it easier for users to access such information. Unlike the graph shown in *Figure 1* the information gathered for a STFT is not as easily gathered or manipulated due to its complex nature.

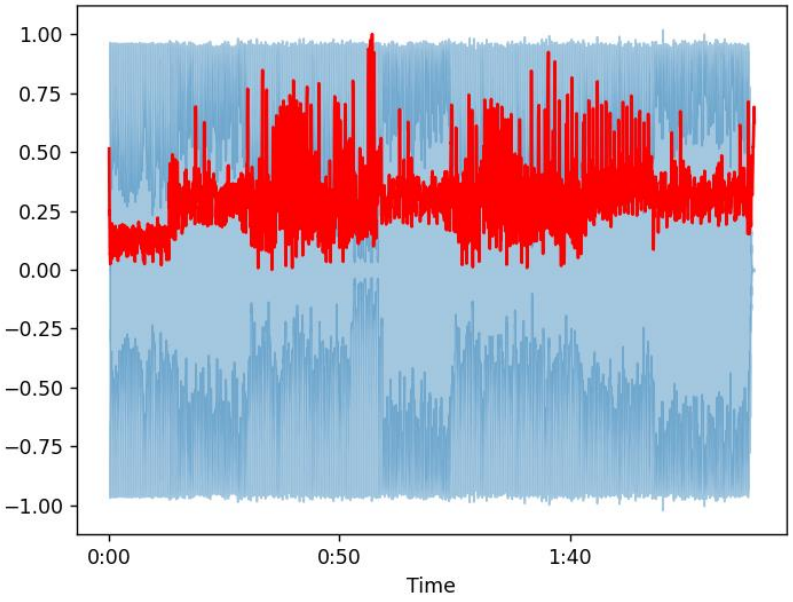

Figure 3 - Spectral Centroid Spectrogram

The Spectral Centroid Spectrogram is similar to the graph shown in *Figure 1*, however, it provides further and more simplified insight into the data. With this graph a calculation on the mean of each frequency at each timestamp is being run, this is shown with the red lines covering the spectrogram. This better simplifies the data as it can provide insight into a more compressed and manageable form of song data. As the data provided by a WAV or MP3 file can be quite large, this simplification of data could be useful when handling and manipulating the data.

Extracting and Manipulating Data

As previously discussed there are different methods by which you could extract and manipulate music data. When understanding how the data can be visualised and what different functions are carried within the song data, showcases how complicated the data can truly be. As it stands within the field of Machine Learning there is no correct way to extract data for an algorithm to learn and produce its own samples. Music is still human in nature and although we can compress this data to understand a song computationally with patterns being present within this, it is still arbitrary and provides very limited insight. Within this project, data was extracted on WAV and MIDI datatypes. The data extracted and used for a WAV file was the song data itself, with complications embedded within this. With MIDI data the data extracted was note data based on a particular instrument, which for this project was the Piano.

When loading a WAV or MP3 file into a python script with the Librosa library, the two pieces of data that are of concern is the song data and the sample rate. The song data, as previously mentioned, is an array that dictates certain aspects of a song at a given time point. With the use of the NumPy library, the array can be better understood and extracted effectively. As shown by the spectrograms in the previous section the song data can be narrowed down to frequencies ranging from -1 to 1, which can be used to produce a sound. Manipulating this within the song data, it is possible to input the new array into a Librosa function that reconstructs the audio into a new WAV or MP3 file.

The issue that comes with extracting and changing the song data of a WAV file is the fact that it is incredibly large, making it hard to process effectively. Although the original 16-bit data was halved for

the purpose of this project, it still proved too large to handle. The other issue was the fact that manipulating this data would still provide no real patterns, grouping was tried, and creating separate arrays depending on the frequencies at hand but this still proved difficult due to the size of a file. Shortened from 30 seconds to 10-second clips this became slightly more manageable but the structure of the data was still too complex with no patterns, making white noise when manipulated in any way. What was discovered was that although the song data is most accessible and fairly easy for a user to read, when manipulating the sheer scale of the song data became too unmanageable to manipulate in any meaningful way.

On the other hand, extracting and manipulating song data using a MIDI file is far easier in comparison. The issue with this file type, however, is the fact that it only really holds sheet music and has no sound to it unless ran in a responding program. When loading a MIDI file using the Music21 library the music is split based on instrument, showing the notes and chords being used. For this project the notes of the Piano were the main focus, extracting the data for this was done by creating an array of the notes along with the pitch. The array containing each note was then numbered, giving each note a number of its own. Then the network is used to find the frequency of notes as well as any patterns that may be contained in the data. This note data can then be changed and converted back into a MIDI file by reversing the numbering system of the notes in the array and putting the notes into the new file. If instrument data was also to be manipulated this process would be fairly straightforward as well, however, for this project that was not necessary for comparison.

Overall, there are differences in what data can be extracted and how they can be manipulated depending on the file type. Raw musical data can be extracted using Librosa, with an array being readily made and available by the library. However, due to the size of a WAV or MP3 file, this information is hard to effectively process and change in any way that can be usable. With MIDI data, on the other hand, the note data is easily obtained for an instrument in the song and an array can be organised effectively and ready for analysis of a network.

Deep Neural Network

When considering a network structure to use for this project, Wave Net was an obvious choice as it is designed for raw audio. However as discussed earlier, using the sound data of a WAV file for music there is a complication with the size of the data and the structure in terms of producing music. Wave Net is a Deep Neural Network structure that is used for voice recognition and resampling created by Meta's Deep Mind. During the process of this project, due to the complex nature of musical WAV files, I decided to use a structure based on Wave Net but for MIDI files.

Using this structure the hope was to move the raw song data from a WAV file to work within this structure, however, more complications arose when it came to organising the frequencies. The organising thought to work was done by moving the relevant numbers within the files array based on high frequency (close to 1), mid-frequency (close to 0), and low frequency (close to -1). The issue with creating these new arrays with the frequencies ranging within their respective group was structure. The complexity of finding where the frequency spikes are required too much computer power for this project and would have still made white noise with a random structure based on an infinite variety of frequencies captured from the different songs in the dataset.

The final network made was only compatible with MIDI files, even though the attempts at structuring the frequencies within the WAV files array remain as a proof of concept. The idea of the final product was to take the notes found within each song in the dataset and find similarities. Given each timestep

of the song, the network figures out the most probable note to go next based on the notes array given for each song in the dataset. As the dataset wasn't raw audio the network matched frequent numbers which were previously given to each of the notes. The initial idea of adding a WAV file's array to the network would've had the most common or the mean common frequencies matched and predicted based on the dataset. Overall, the network predicted the next note of a song based on the dataset given, then created a new array to be used as a new song ready to be converted back to a MIDI file.

The build of the network however was run using TensorFlow and Kera's using 1-dimensional convolution. This means that the output of the predictions made by the network isn't affected by the previous or coming results of the system. The training and evaluation data was split 80 and 20 percent in the respective field. The activation function of these casual 1D convolution layers was Rectifier and SoftMax activations. The input would go through these convolution layers with the output being fed again into two more. Each of the inputs are sequences of the songs generated into an array with numbers representing the notes, these then go through the network within the corresponding time stamp of the song. The best model then outputs the array of a new sequence made based on what was learned, and this array gets deciphered back into the notes they correlate to before a new sample is produced.

Limitations

One of the major limitations of the project was the scope of the model. With this, the original purpose of the project was to create a model that would be able to effectively listen to and learn from a dataset consisting of full songs in MP3 or WAV format. This was due to inspiration from the Jukebox neural network produced by OpenAI. However, as the project went on with a different approach to understanding the sound files and how a Wave Net structure could be incorporated to understand the frequencies within the raw song data, it became more apparent that the file types have varying methods in which the data can be interpreted. With this, experiments were then done to see what datatypes were able to produce using the same deep neural network as that in Wave Net. This change in research helped to provide the results gathered by this project, with the further insight being available.

In regards to the dataset used, it was found that the density of WAV and MP3 files provides complications with retrieving data that has patterns for the model to learn from. As the raw song data was used to get the frequencies of the file at hand, the data was too clouded as it contained a range of different sounds. This provided a large range of varying frequencies amongst timestamps that were placed close together within a short sample. Even with shrinking the sample sizes, the density of data due to this was still an issue. If the dataset was comprised of music with less noise contained within a sample, the issue here may be resolved.

Results and Findings

Within this project, different datatypes for raw music files were explored and analysed in order to find if a Deep Neural Network was able to produce music of its own. Although there were some dead ends in regards to the datatypes such as WAV and MP3 files, the extracting of data was scrutinised to see if potential patterns could be made. As found the MIDI file preformed best to recreate songs, however this wasn't using raw data as WAV and MP3 but sheet music instead.

Initially MP3 files were tested to see what data could be collected for a network to use and learn from. The issue found here was the fact that MP3 was more compressed with data being lost, proving better for music mixing or classification learning tasks. After these findings a move was made towards WAV files, due to them being richer in data that could potentially be of use for such a project. With this similar data as that in MP3 files was able to be produced, with the added access to the music file which was in the form of an array of frequencies. Here it was tested to see what could be done with these frequencies to better understand what patterns could be made and if music could be produced using them. Although the data seemed fairly difficult to find patterns and to do anything meaningful with them, if the array was changed or if a new one was put into a sound file then white noise could be produced. Although this still didn't fit what was needed for the project, the frequencies being able to make noise proved that a potential was there. The next step was to create the Neural Network, where a Wave Net structure was used. With the music files in the dataset making non-sensical arrays, the network wasn't able to process the WAV files effectively. From here the MIDI files were introduced, where the notes were able to be retrieved with patterns being easily found within them.

So although the Wave Net structure was made for raw audio, in this project it only worked effectively on sheet music. This could be down to the idea that with a music dataset consisting of full songs, containing vocals and all backing beats, in WAV format had too much noise at once. If each song was split into respective instruments or beats as well as vocals, maybe a better understanding could be made. The issue with this being the difficulty of segregating these sounds without the correct software to do so and save them in different files. Although this was tried in another project by Keith Bloemer [Bloemer, 2021] with just guitar audio, producing just white noise after going through a Wave Net deep learning structure. Overall, raw music files are too complex unless segregated effectively, compared to sheet music where an algorithm is able to quickly assign patterns due to note orders and frequencies.

Summary

This project tried to tackle the task of producing novel music using raw audio processing within a deep neural network structure, in doing so experimenting with different approaches. Musical data is very complex and can be extracted and understood in a variety of ways, depending on the type of information it can also be manipulated and changed. Within this project, it was found that the data extracted from the raw audio file needed to present certain patterns for the network to interpret and predict sequences that might come next within a song. The complexity of music makes this pattern finding hard when interpreting a whole song, but if broken down it could be made easier.

Research and experiments were done with a few Python libraries within this project for varying tasks from extracting data to creating the neural network. Librosa was used for interpreting WAV and MP3 data at the beginning of the project, being able to extract frequency and amplitude data, and presenting them as understandable graphs with the help of matplotlib. When moving to MIDI data the focus was more on instrumental data and notes played within a sample, this was extracted using Music21. Both libraries provided insight into the music at hand and helped to create arrays of the data that can be interpreted by the network. The deep neural network approach used was based on the Wave Net structure created by Deep Mind, within Python being built using TensorFlow.

The creation of the network and the data being processed provided many challenges to it. Regarding MP3 data it was found that the data collected wouldn't have been enough to create patterns, however, this approach may prove more useful when used for another purpose like genre or mood classification. With WAV data the information collected from raw audio was in the form of a large array containing frequency information based on the sample rate and timestamp of the song. This data, although it could be interpreted and manipulated to create new sounds, proved too dense and complex to create anything of significance using the approach at hand. Although in concept this method could work effectively, within this project using the resources at hand it was more accessible to use the MIDI data previously extracted.

Along the process of developing the project, much research was done into different approaches and technologies that could be used to achieve the goals at hand. The approach taken for the final product was based on this research as well as through on hands testing of the libraries and data used, to fully understand how these components impact and help the process of producing novel music. At the end of the project, it was found that some of the data were unable to fit the requirements for the network, however, MIDI was the file type that worked best. It was found that there are different ways of interpreting this data, through research these gave light to how these collection methods prove useful to other areas of practice within the industry.

Further Work

When working on the raw audio using both WAV and MP3 files within this project, the dataset may have been overambitious. The dataset comprised of full songs that would be broken down into smaller samples using python script. The issue with this is that the data is very dense and the songs comprise of a variety of different sounds from an assortment of instruments and vocals. In the future if a dataset was made where each song was broken down into their respective instruments or sound labels, then it might be possible to create a network that can understand these patterns separately before mixing them together to make new samples. With this approach it might be that the WAV file's frequency array might show more obvious patterns for certain instruments, where the Wave Net structure could make better predictions like it does with voice data currently.

Another approach is using the spectrograms to gain a perspective of patterns that could help recreate frequency patterns for a new song sample. Currently this approach is used for classification problems with music genres or artist identification. However, using the Librosa library a feature that could use a generated spectrogram and convert that back into raw audio is unavailable as yet due to the complexity of such a task and the simplicity of the data presented in a spectrogram. In theory this approach could potentially work, it would be better suited for beats or instrumental datasets rather than vocal data due to the nature of understanding gathered by such data.

Conclusion

The project started off with a focus purely on using raw audio files like that of WAV and MP3 to feed into a neural network to produce new song samples. As more research was conducted and the files within the dataset were scrutinised further, it became interesting to find which filetype would be better for the task of recreating music and why. Although WAV files were found to be best for learning and gathering information on raw musical files, it proved too dense with data that was hard to learn from and manipulate in any meaningful way. With MP3 data it was found to be a useful and condensed dataset that can be used for classification problems within the music industry, however this wasn't the scope of the project. The final music dataset found consisted of MIDI files, which held instrumental data with notes and chords, much like sheet music. Using a Wave Net network structure this data was able to be used to find patterns to predict notes that should be played in sequence. Although Wave Net is meant to be a structure for raw audio, it is hypothesised that the WAV file dataset was contained too much musical information to make anything relevant to the scope of the project.

Overall, after this project the idea of being able to recreate music still sounds far off due to its complex nature. Although, some movements can be made to understanding certain aspects of raw music data in isolated cases. The more musical data needed to process, the more processing power required and the bigger the dataset needs to be. Although samples that resemble musical pieces can be made, shown through applications like Jukebox, this project has seen how complex music is if learning from raw audio. Its interesting to see how the different datatypes can provide varying avenues for understanding data within musical files. Using the Wave Net structure also provides and interesting look into how raw audio can be used, and with the right set of data maybe it could prove useful to recreate the musical process.

Bibliography

Bloemer, K., 2021. *Neural Networks for Real-Time Audio: WaveNet*. [online] Medium. Available at: <https://towardsdatascience.com/neural-networks-for-real-time-audio-wavenet-2b5cdf791c4f>

Dhariwal, P., Jun, H., Payne, C., Kim, J., Radford, A. and Sutskever, I., 2020. *Jukebox: A Generative Model for Music*. [online] arXiv.org. Available at: <https://arxiv.org/abs/2005.00341>

Oord, A. and Dieleman, S., 2016. *WaveNet: A generative model for raw audio*. [online] Deepmind.com. Available at: <https://www.deepmind.com/blog/wavenet-a-generative-model-for-raw-audio>

Jones, M., 2022. *Spotify's Algorithm Explained*. [online] Bippermedia. Available at: <https://bippermedia.com/spotifys-algorithm-explained/#:~:text=The%2030%20second%20rule%20explains,descriptive%20words%20about%20those%20songs.>

Korstanje, J., 2021. *Machine Learning on Sound and Audio data*. [online] Medium. Available at: <https://towardsdatascience.com/machine-learning-on-sound-and-audio-data-3ae03bcf5095>

Mandapaka, K., 2021. *Handling audio data for machine learning*. [online] Medium. Available at: <https://medium.com/mlearning-ai/handling-audio-data-for-machine-learning-7ba225f183cb>

Pai, A., 2020. *Automatic Music Generation | Music Generation Deep Learning*. [online] Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2020/01/how-to-perform-automatic-music-generation/>

Korstanje, J., 2021. *What is sound?*. [online] Medium. Available at: <https://towardsdatascience.com/what-is-sound-691988d780bb>

Briot, J. and Pachet, F., 2018. *Deep learning for music generation: challenges and directions*.

Ayushi Rawat, Aayush Roy, K. Shambavi, Music Generation using Wavenet Architecture, International Journal of Electrical Engineering and Technology (IJEET), 12(6), 2021, pp. 12-18.

Jean-Pierre Briot, Gaetan Hadjeres and Francois-David Pachet, Deep Learning Techniques for Music Generation, Computational Synthesis and Creative Systems, Springer, 2019.

Snell, C., 2021. *Understanding VQ-VAE (DALL-E Explained Pt. 1)*. [online] Ml.berkeley.edu. Available at: <https://ml.berkeley.edu/blog/posts/vq-vae/>