Investigating the Interpretability of Recurrent Neural Networks for Music Genre Classification

Leander Stephen

Wheeler High School

Advanced Scientific Research

Dr. Cody

May 5, 2023

Abstract

Artificial intelligence, particularly neural networks, has been increasingly prevalent in our daily lives during the past few decades. Music recognition is a frequent use of neural networks. The accuracy of these neural networks, particularly recurrent neural networks (RNNs) and long short-term memory networks, is becoming increasingly crucial as people incorporate more technology into music recognition (Wang & Sohail, 2022). RNN models have been developed to conduct music genre classification, but there has been very little investigation into how these neural networks work in the present literature. This study sought to contribute to the present literature's limited research on the explainability of RNN models for music genre classification. A deep learning RNN model with a final accuracy of 94.17% was built to conduct classification on a dataset of electronic and jazz MIDI recordings. Following that, the model was given a sample jazz MIDI file and accurately identified it. The RNN's prediction was then interpreted using saliency mapping and input optimization; however, the maps that were developed revealed few patterns. Overall, the results demonstrated that music genre classification is too complex for an RNN to explain. As a model's complexity increases, its explainability decreases (Bhardwaj et al., 2018). To attain high accuracy, the RNN required a high level of complexity, making the saliency map and input optimization note maps difficult to interpret. These findings suggest that future research could address the limitations of this study and further investigate the explainability of RNNs for music genre classification.

Key Words: Recurrent neural network (RNN), Music genre classification, Deep learning

Table of Contents

Investigating the Interpretability of Recurrent Neural Networks for Music Genre Classification

In the modern era, digital music is gradually replacing physical music. Because everyone has different musical tastes, music genre classification technology has the potential to improve all types of music software (Gao, 2022). Given the volume of music available in the current online library, a program capable of accurately categorizing genres is required (Yu-Huei Cheng et al., 2021). Accurate music genre recognition is the key to audience management, collection, and recommendation systems (He, 2022). Deep learning system development has recently accelerated, hastening the process of voice recognition research (Cui & Wang, 2022). The accuracy of these neural networks is becoming increasingly important as people incorporate more technology into music recognition, particularly recurrent neural networks and long short-term memory networks (Wang & Sohail, 2022). Because neural networks are difficult to interpret, the ability to explain the process from input to output is critical. The purpose of this research is to add to the limited knowledge on the interpretability of recurrent neural networks for music genre classification.

Artificial Intelligence

Over the last few decades, humans have made incremental advancements in creating intelligent machines that can carry out human-like jobs. Artificial intelligence is the discipline in question. An area of artificial intelligence called "deep learning" was developed to simulate the activity of neurons in the neocortex, which accounts for 80% of human thought (Bhardwaj et al., 2018). Machine learning is carried out via neural networks, which are deep learning computing systems made up of artificial neurons, or nodes. A neural network's fundamental component is a computer learning a task by examining training datasets. These neural networks have been highly successful in various applications, such as image classification and natural language processing.

However, the downside of neural networks is their interpretability. On the one hand, neural networks can achieve high accuracy in complex tasks, such as image classification or language translation, which can be difficult to achieve with other methods. On the other hand, the inner workings of neural networks can be difficult to interpret, as they often involve many layers of nonlinear processing that can make it challenging to understand the relationships between inputs and outputs.

Recurrent Neural Networks

A recurrent neural network (RNN) is a neural network that is efficient for large datasets due to its ability to repeat itself over different subsets of data (Ghotra & Dua, 2017). A large dataset results in a more accurate representation of a music genre, allowing the neural network to specialize in detecting specific properties of music. Due to this, RNNs are often used in music recognition and synthesis systems (Wang et al., 2018). However, a limitation of RNNs is the need for them to perform backpropagation (a method to reduce error) through time (Bhardwaj et al., 2018). Although it is possible to perform backpropagation through time, it often produces poor results due to the vanishing gradient problem in RNNs. This issue can be solved by implementing a neural network with a minimal dataset without compromising the number of files needed to achieve high accuracy.

Saliency Maps

Saliency maps are a type of visualization tool used in machine learning and deep learning to identify the most important features or elements in an input that contributed to the model's output or decision. In essence, saliency maps aid in highlighting the aspects of an input that were most influential in determining the output, allowing users to better understand how the model made its decision. This is especially useful when the decision-making process of a model is

opaque or difficult to interpret. There are several methods for creating saliency maps, but one common method is to compute the gradients of the input with respect to the output. This enables the user to determine which features or elements of the input had the most influence on the final decision. Saliency maps have a wide range of applications, including image recognition, natural language processing, and audio analysis. Saliency maps can be used in audio analysis to determine which segments or elements of an audio signal are most important in determining a particular classification or output.

Input Optimization

Input optimization is a machine learning and deep learning strategy that modifies input data to maximize or reduce a given output or decision made by the model. The purpose of input optimization is to find the lowest change in input that results in a significant change in output. In the case of binary classification, this means finding the least change in input that results in a flip in classification. Input optimization can be used to improve a model's resilience by finding places where it is particularly sensitive to adversarial attacks, or to detect shortcomings in a model's decision-making process. In some circumstances, input optimization can even be used to generate new inputs that meet a predefined set of requirements, such as synthesizing realistic images or audio samples. Gradient descent is a typical technique used in input optimization, in which the model's gradient is used to iteratively adjust the input in a way that maximizes or minimizes the output. This is especially useful when the model is differentiable, which means that its output can be stated as a function of its input.

Rationale

As the complexity of artificial intelligence (AI) applications in the music industry grows, the interpretability of these AIs deteriorates. As a result, the efficiency and comprehension of neural networks are critical to the capability of numerous AI applications. This study's approach to solving this problem is to implement an RNN for music genre classification and use saliency mapping and input optimization in order to investigate the model's explainability. A better understanding of the RNN model created will lead to a better understanding of AI creations in general, as well as future research on the development of reliable music genre classification applications. Because there are few distinct differences between musical styles, this study examined how well a neural network can categorize music (Fan, 2022). When an AI replaces humans in this task, the accuracy of music recognition improves significantly. The research in this study was conducted under Professor Arthur Choi, a computer science professor who specializes in neural networks and AI. This research broadened both my knowledge and that of my professor. Understanding this topic propels future advancements in more sophisticated AI and improves our general understanding of how neural networks work.

I have used a variety of artificial intelligence in music applications, including music recommendation systems and synthetic music synthesizers, as a passionate music listener and creator. Verifying the reliability of these numerous deep learning applications in the music industry was one of the goals of this study. Additionally, music business trends can be easily found and predicted using automated music genre classification (Chowdhry, 2021). Additionally, this research added to the corpus of knowledge on neural networks' algorithmic structure.

Statement of Focus and Subproblems

The interpretability of AI models is a subject of growing concern as AI applications become more widespread. In the music industry, interpretable AI is becoming more and more crucial because it enables users to comprehend how the model makes predictions and to believe the recommendations made (Wang & Sohail, 2022). For AI models to be transparent and accountable, it is essential to uncover how a neural network functions. Additionally, it aids in the discovery of potential biases and enhances the general precision and dependability of these models. In addition to enhancing the precision and effectiveness of music applications, research on the implementation of neural networks for music genre classification is crucial for ensuring the interpretability and transparency of AI models. Building trust and ensuring the accountability of AI models requires interpretable AI. For a neural network to perform at its best and to increase its accuracy and dependability, it is essential to understand how it functions. For this reason, the goal of this study is to successfully implement an RNN for music genre classification and use saliency mapping and input optimization in order to investigate its predictions. In order to reach this objective, the following question was synthesized:

- Can a recurrent neural network that performs music genre classification be implemented and explained?

These subproblems were also explored in this study:

- Basic Subproblem: What properties of music can an RNN detect in a MIDI file to classify music?
- Basic Subproblem: What is a viable way to measure a neural network's effectiveness?
- Design: Create and implement a recurrent neural network to perform music genre classification on a dataset of Musical Instrument Digital Interface (MIDI) files. The

measure of success for this subproblem is the accuracy of the neural network's ability to classify, which will be used to evaluate each iteration of the design. A successful design will have an accuracy of at least 90%.

- Design: Perform input optimization on the loss function to flip the classification of a sample MIDI file and create a note map of the altered MIDI file. Print a saliency map based on the gradient of the sample MIDI file. The measures of success include:
    - Creation of an input optimization note map
    - Creation of a saliency map

Literature Review

Introduction

The goal of this study was to develop an RNN capable of performing precise music genre classification and to use saliency mapping and input optimization in order to investigate its interpretability. The significance of artificial intelligence's reliability is steadily growing as it is more and more integrated into human daily life, particularly RNNs and LSTMs (Wang & Sohail, 2022). In this study, an RNN that can categorize music into genres was built and designed so that it can be interpreted and analyzed.

Significant information from this literature review was used in the design process of the RNN. The development, testing, and analysis of the neural network were aided by current literature, which also supplied essential background knowledge for the analysis of the retrieved results. The design of this study depended on understanding the characteristics of music that a neural network can recognize and how to retrieve the RNN's accuracy.

Properties of Music That Can Be Detected

Understanding how a neural network moves from input to output was a key component of this study. Due to the difficulty of selecting and extracting appropriate audio features, music genre classification is considered a challenging task (Yu-Huei Cheng et al., 2021). The key challenge of determining which aspects of music are most crucial for achieving this goal is addressed in this question. The interpretability of the developed neural network will grow if it is able to identify the characteristics of music that the neural network is detecting in order to categorize music as a particular genre. Researchers may create better models, increase the overall accuracy and explainability of music genre categorization, and more accurately categorize music by knowing the characteristics of music that an RNN can detect.

Restraint of MIDI Files

Most researchers employ Waveform Audio File Format, MPEG Audio Layer-3, and other file formats that store waveforms (He, 2022). Musical Instrument Digital Interface (MIDI) files, which solely contain events rather than waveforms, which are visual representations of sound as time on the x and y axis (Chowdhry, 2021), were used in this investigation. These files only preserve the pitch, step, and duration of notes, as opposed to waveform files, which can also preserve rhythm, strength, and other qualities that can only be examined in waveforms. As a result, MIDI files limit the ability of a neural network to categorize audio genres. The potential properties that the neural networks can detect are reduced when they are limited to MIDI files with fewer fine details. Less detailed data, however, gives the neural network less to work with and is likely to result in less accurate classification. However, if the effectiveness can be justified, the lesser accuracy is a reasonable trade-off. MIDI data is crucial to this study as it consists of simpler data that will allow for increased interpretability.

Complexity of Dataset

Deep learning neural networks require a large amount of training data, and the quality of this data has a large impact on the accuracy of the model (Cui & Wang, 2022). The behavior of a neural network can frequently be better understood if it is trained on a less complicated dataset that does not compromise quality. The larger and more complex the dataset, the harder it is to train the network effectively (Ghotra & Dua, 2017). This is due to the fact that complicated datasets may include a number of confusing variables, making it challenging to pinpoint the precise features that are responsible for the model's predictions. However, it becomes simpler to comprehend the relationship between the input features and the model's output when the dataset is less complicated. Reducing the number and complexity of the MIDI files included in the

model is a way to increase explainability. It can be challenging to determine which input features are most crucial for the performance of the model when there are numerous of them and they each include many pattern variations. It is simpler to determine which characteristics are responsible for the model's predictions and to comprehend how these features relate to one another when there are fewer input features.

Specific Musical Features

MIDI files offer a digital representation of the music together with information about the instruments, tempo, and note order. These traits can be taught to an RNN, which can then use them to categorize music. The note sequence of a MIDI file, for example, can be used to identify recurring melodic motifs, rhythmic patterns, and other elements of a specific musical genre. The tempo of the music can be evaluated by the RNN to determine how quickly the notes are played and to discriminate between music from different eras or genres (Wang & Sohail, 2022). The neural network is also able to determine how the music is instrumented. The RNN may also examine chord progressions to find the harmonic patterns incorporated into the music. Also, it may assess the music's structure to classify it according to how it is organized, such as verse-chorus-bridge in pop music or the several movements in a classical composition. An RNN can determine the distinctive qualities of various genres of music and categorize them in accordance with these diverse MIDI file features.

Measuring Neural Network Effectiveness

It is difficult to tell if a neural network is succeeding in its objectives and whether it is a practical method for classifying music genres without a reliable and accurate means to gauge the performance of a neural network, which is why this study's focus issue required such a method. Measuring efficacy becomes much more important in the case of an explainable neural network

because the network's success depends heavily on its capacity to produce outcomes that can be understood (Cui & Wang, 2022). This study pinpointed areas for improvement and worked toward the network's performance optimization by analyzing the neural network's efficiency. Additionally, the research located architectural flaws in the neural network, which will guide future research on explainable neural networks.

Accuracy

There are several metrics that can be used to assess the effectiveness of a neural network, depending on the specific task and application (Bhardwaj et al., 2018). One of the best measures for measuring a neural network's effectiveness in classification tasks is accuracy. The percentage of accurate predictions made by the network is what is known as accuracy (Bhardwaj et al., 2018). The accuracy in this study refers to how many MIDI files were correctly categorized into their genre.

Cheng, Chang, Nguyen, and Kuo (2021) used accuracy as one of their evaluation metrics in their study on automatic music genre classification based on CRNN. They reported that their CRNN model achieved an accuracy of 71.8% on the dataset they used, which they deemed to be a satisfactory performance.

It is crucial to remember that accuracy might not always give a full picture of a neural network's performance. For instance, a network might occasionally attain high accuracy while still experiencing problems with false positives or false negatives. When considering the performance of the network in terms of true positives, false positives, true negatives, and false negatives, other measures like precision, recall, and F1 score can be helpful.

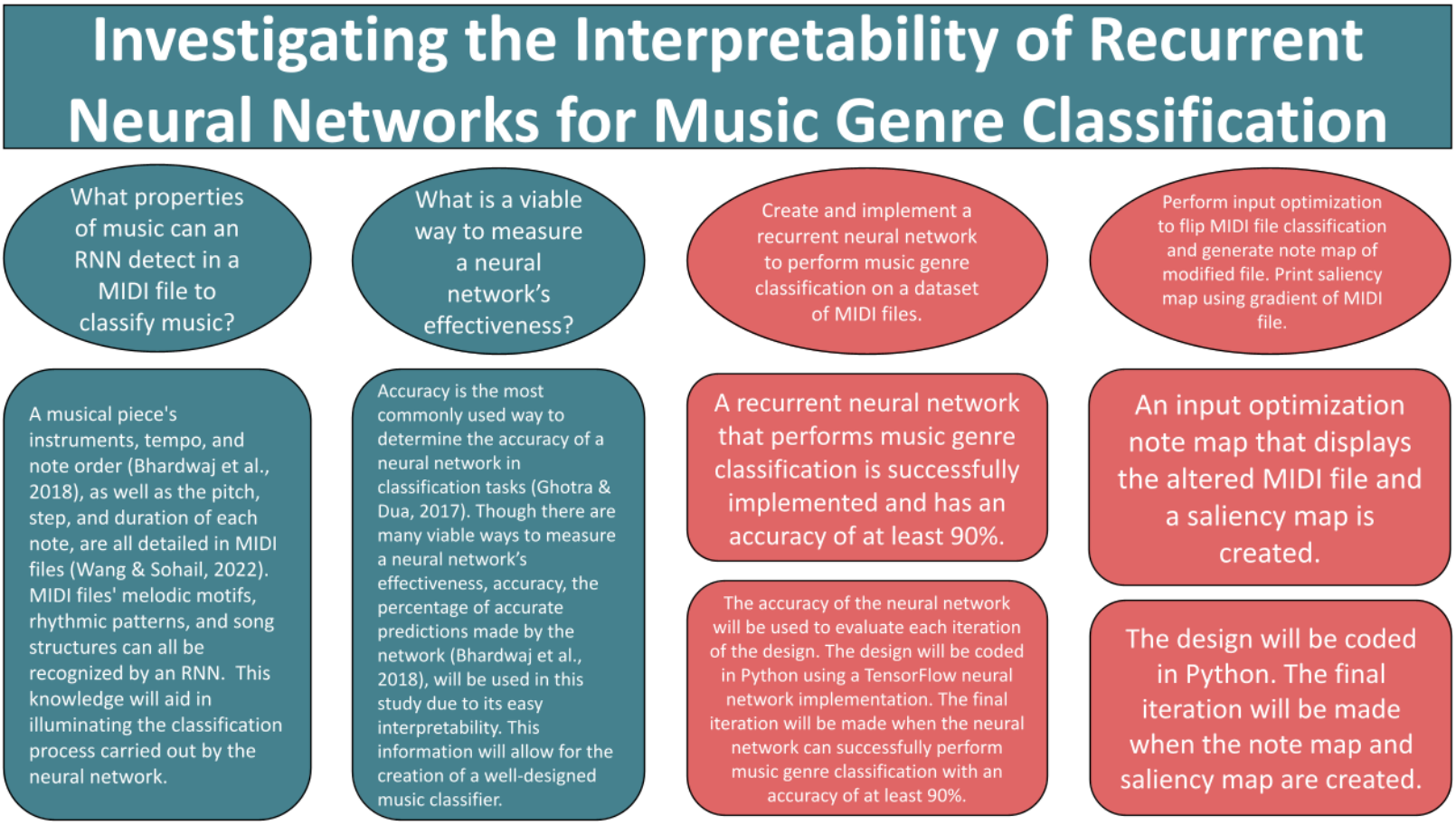
Loss

The loss, or the discrepancy between the output of the network and the anticipated output, is a gauge of how effectively the neural network is carrying out its function (Ghotra & Dua, 2017). The main goal of training a neural network is loss minimization because doing so increases the predictive power of the network. The error should go down as the network continues to train, showing that it is becoming more adept at its duty. As a result, the loss can be used as a metric to assess a neural network's effectiveness. The accuracy of the network's predictions and its effectiveness are both correlated with its loss level. The TensorFlow RNN model used in this study automatically retrieves the loss for each epoch in the training process, making it a reliable gauge of effectiveness.

Summary and Implications

The applications of music genre classification in current literature allow for this study to contribute to literature because there is limited information on the explainability of recurrent neural networks for music genre classification. Current research on music genre classification shows that properties such as tempo, song structure, and melodic motifs are able to be classified by RNNs (Wang & Sohail, 2022). This information will accelerate the process of explaining how the RNN model functions. The current studies and research on music genre classification propose valid methods to measure the effectiveness of RNNs using metrics such as accuracy and error. These methods propelled the study forward by ensuring that the neural network’s effectiveness was accurately represented. The literature indicates that a dataset of simple MIDI files and a RNN will result in data that can be analyzed and examined to reach the goal of this study.

Research Map



Scope of Study

Delimitations and Limitations

- The results of this study only apply to recurrent neural networks and not other neural network architectures.
- This study only focuses on the ability of an RNN to classify electronic and jazz music, which limits the generalizability of the model to music genre classification in general.
- This study is limited by each MIDI file having a maximum of 100 notes in order to ensure reasonable training time.
- The study is limited to a dataset of 642 MIDI files to ensure reasonable training time.
- This study is limited by the computing power of the Intel i7 CPU core of an Acer Predator Helios 300 computer.

Assumptions

- The MIDI files used for training the RNN accurately represent the characteristics of their genre.
- The accuracy metric provided by TensorFlow correctly identifies the percentage of correct classifications.
- The RNN will train and compile in a reasonable amount of time.
- The size of the provided dataset is enough for the RNN to detect sequential patterns in the data.
- Saliency mapping and input optimization are enough to explain the RNN's predictions.

Key Terms

Key Term	Definition
Neural Network	A neural network is a computational system composed of interconnected processing elements, or neurons, that work in parallel to solve a specific problem (Bhardwaj et al., 2018).
Recurrent Neural Network (RNN)	An RNN is a type of neural network that can take sequences of data as input, and uses the previous outputs as feedback in the next step of computation (Ghotra & Dua, 2017).
Music Genre	A music genre is a category and a set of conventions that certain music identifies with. Music genre classification is a way to classify music into different styles based on their features (Chowdhry, 2021).
Musical Instrument Digital Interface (MIDI) File	MIDI files offer a digital representation of music with information about the instruments, tempo, and note order (Chowdhry, 2021).
Node	The fundamental units of a neural network are nodes. They take in information, process it, and then generate outputs that other network nodes can use as inputs (Ghotra & Dua, 2017).
Layer	A layer is a collection of processing nodes or neurons that carry out a single task, such as data input, processing, or output. Typically, a neural network has multiple layers that are connected to one another in a particular way (Bhardwaj et al., 2018).
Activation Function	The output of a neural network node can be made non-linear by using mathematical functions called activation functions. Before it is sent to the network's next layer, a node's output must pass through an activation function. Based on the input it receives, activation functions are used to decide whether a neuron should be activated or not (Bhardwaj et al., 2018).

Saliency Map	A saliency map is a representation of weights that highlights the most important aspects of the input data that affect the data's output (Yu-Huei Cheng et al., 2021).
Loss Function	A loss function is a measurement of the deviation between predicted and actual output values (Bhardwaj, Di, & Wei, 2018).
Optimizer	An optimizer is used to minimize the loss function by updating the weights and biases of the neural network (Bhardwaj, Di, & Wei, 2018).
Adam Optimizer	The Adam optimizer is an extension to stochastic gradient descent that incorporates moving averages of the parameters and their gradients to scale the learning rate dynamically (Bhardwaj et al., 2018).
Epoch	A complete iteration through a training dataset during the training of a neural network (Yu-Huei Cheng et al., 2021).

Methodology

The purpose of this study is to build an RNN that can perform music genre classification and then use saliency mapping and input optimization in order to explore the RNN's explainability. The implementation of the RNN is the subject of the first design subproblem. The creation of the saliency map and input optimization note map is the subject of the second design subproblem. The methods and data collection techniques used to investigate this subproblem will be covered in this section. Understanding the characteristics of music that an RNN can recognize to distinguish between genres and finding a practical way to evaluate the network's performance are necessary when designing an RNN to classify music genres in MIDI files. This study used MIDI files, which only contain events, as opposed to waveforms (visual representations of sound as time on the x and y axis). These files only keep the pitch, step, and duration of notes, as opposed to waveform files, which can also keep rhythm, strength, and other qualities that can only be examined in waveforms. Given that it reveals the capabilities of the RNN, this information was crucial for figuring out how interpretable it is. A viable method of measuring the effectiveness of the neural network was also critical to the success of its implementation. A method of evaluation was needed in order to determine whether the RNN is successfully performing music genre classification before attempting to determine its explainability.

Implementation of an RNN

TensorFlow was used as the implementation platform for the RNN, and the RNN was programmed in Python 3.9 in Visual Studio Code. The first step in the design of the RNN was preprocessing and implementing the MIDI files from the dataset into the neural network framework. Preprocessing refers to making sure that the MIDI files are ready to be used by the neural network. This includes truncation, or shortening, the MIDI data (Bhardwaj et al., 2018),

and encoding the files with labels containing their genre. The ADL Piano MIDI dataset was downloaded onto the computer used for the neural network testing and condensed down to just the genres of electronic and jazz. The MIDI files were accessed through the file path in the computer's directory and entered into the framework of the code. The MIDI files were converted into arrays, which is a format that an RNN can take in as a dataset, using the NumPy library in Python. These arrays consisted of numbers representing the pitches of each note in each MIDI file. These arrays were stripped down to a maximum of 100 notes each to ensure reasonable training time. Each file had a corresponding label in an alternate array to represent the genre of the MIDI file. Next, the neural network architecture was created, which included writing code that implements TensorFlow and optimizing the number of nodes, layers, and the activation function to find a balance between accuracy and simplicity. After that, the model was compiled, which included specifying the loss function, optimizer, and metric (accuracy). Next, the testing hyperparameters of epochs and batch size were written. The architecture, compilation, and testing hyperparameters were tested and adjusted until the neural network achieved at least 90% accuracy after the code had been debugged and functioned properly. The accuracy was automatically provided after the completion of each training round. Iterations of the neural network occurred until this accuracy was attained. Each iteration was documented via a plot of accuracy and loss over each epoch of the training round and looked over by my mentor to make sure that the code was functioning.

Input Optimization and Saliency Mapping

Saliency Mapping

A MIDI file of "Two of a Kind," a jazz track by Bobby Darin and Johnny Mercer, was given to the RNN, and a genre prediction was made after the RNN's code had been written,

tested, and an accuracy of at least 90% was achieved. The prediction made by the RNN was investigated using a saliency map. A saliency map highlights the most important aspects of the input data that affect the data's output (Yu-Huei Cheng et al., 2021). The saliency map in this instance took the form of a weighted array. Each of the 100 notes in the test MIDI file was represented by a weight, and the higher the weight's value, the more that note contributed to the predicted genre. This array gave insight into which notes contributed most to the RNN's prediction process.

Input Optimization

The classification that the RNN made was investigated through the optimization of the jazz MIDI file. A delta variable and a loss function, which is defined as the square difference between the predicted label and the opposite label plus a regularization term that encourages minor perturbations, were defined in order to execute the optimization. It then employed the Adam optimizer for a few iterations to minimize the loss with respect to the delta variable. The resulting delta variable was then appended to the original jazz MIDI file sequence to generate a new sequence, which was then fed into the model to generate a new prediction. The prediction before and after optimization, as well as the corresponding label, was documented. A note mapping event for the MIDI file was created. The resulting note map demonstrated the changes that the RNN made in order to flip the classification.

Validity, Reliability, and Trustworthiness

By using a sizable dataset of MIDI files from various genres and making sure the dataset is properly balanced, the study's reliability and trustworthiness is guaranteed. Wang and Sohail (2022) classified musical genres using a dataset of 10,000 MIDI files from different genres. Through the inclusion of an equal number of MIDI files from each genre, they made sure the

dataset was balanced. The ADL Piano MIDI dataset, which includes 11,086 piano pieces from ten different genres, was used in a manner similar to that. The interpretability of the RNN was assessed, as well as the consistency and dependability of the results, using a validation MIDI file. The MIDI file was used for saliency mapping and input optimization in order to demonstrate the explainability of the RNN. Saliency mapping and input optimization are state-of-the-art methods of exploring interpretability. The methodology was clearly documented and transparent to allow for reproducibility and peer review.

Results

The purpose of this section is to explain the process of developing an explainable RNN for music genre classification as well as input optimization and saliency mapping. According to the literature, deep learning models' nonlinear processing makes it difficult to understand how they arrived at their input (Bhardwaj et al., 2018). This study fills the void of knowledge on the interpretability of music genre classification models by implementing an RNN for music genre classification and performing input optimization and saliency mapping in order to investigate its predictions.

Development of the Model

A model development and documentation process was implemented to achieve the goal of creating an RNN that can perform music genre classification on MIDI files. The objective of this subproblem was to achieve an accuracy of at least 90% on the trained RNN model.

Preprocessing

Before the model was developed, the dataset had to be preprocessed in order to ensure successful training. The ADL Piano MIDI dataset was shortened in order to ensure reasonable training time. The original 11,086 MIDI files were cut down to 642 files, which included just the electronic and jazz folders. The files that were located in the electronic folder in the dataset were labeled "0," and the jazz files were labeled "1". These values were used to create an array titled "labels" for supervised learning using. The MIDI files were then converted into arrays representing note pitches using the NumPy library in Python, which is usable data for an RNN. A variable titled "dataset" represents the note pitches. Each MIDI file was then truncated to a maximum of 100 notes to ensure a reasonable training duration (Figure 1).

Figure 1

Shortening of Arrays

```
29  new_dataset = [song[:100] for song in dataset if len(song) > 100]
30  new_labels = [label for label, song in zip(labels, dataset) if len(song) > 100]
31  new_dataset = np.array(new_dataset)
32  new_labels = np.array(new_labels)
```

Architecture Creation

After the data was preprocessed and implemented, the RNN architecture was created and compiled (Figure 2). TensorFlow's Keras API was used to build the neural network architecture. Iterations were made until the RNN reached an accuracy of at least 90%. The model was compiled with the binary cross-entropy loss function, the Adam optimizer, and accuracy as the metric for evaluating training performance. The compilation parameters stayed constant throughout each iteration. The parameters that were changed throughout each iteration included the activation function of each layer, the number of dense layers, the number of nodes in each layer, and the number of epochs used in training.

Figure 2

Compilation

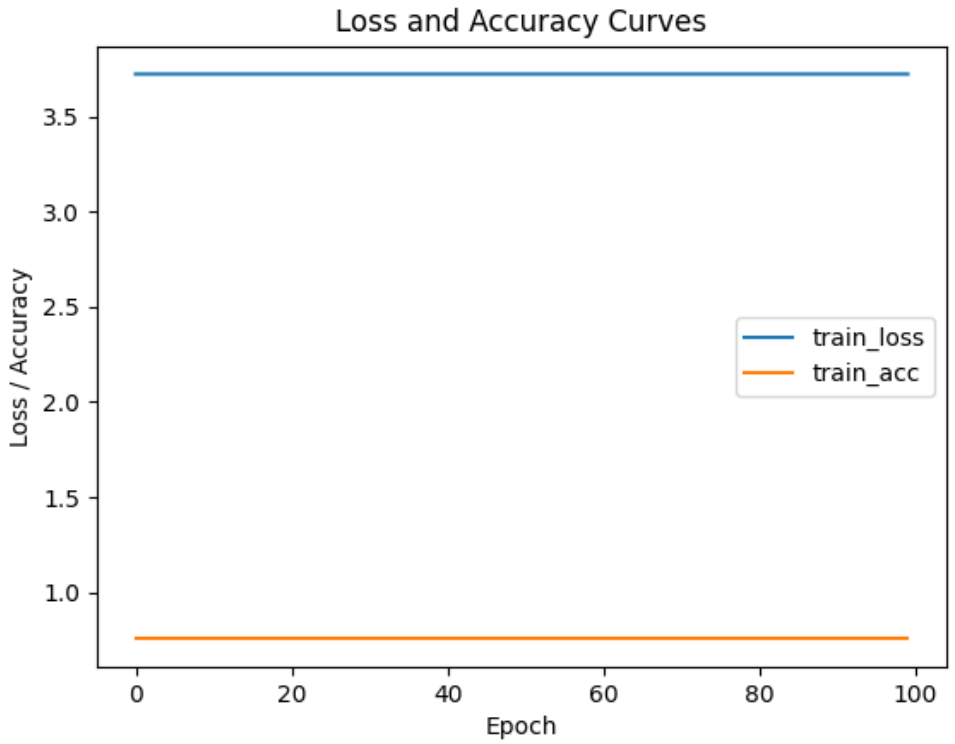
```
50  model.compile(loss='binary_crossentropy',
51                optimizer='adam',
52                metrics=['accuracy'])
```

Iteration 1.

A SimpleRNN layer with 50 nodes and a ReLU activation function was implemented as well as a Dense layer with one node and a ReLu activation function. The model was trained with 100 epochs. As a result, the accuracy remained constant at 0.7599 and the loss remained constant at 3.7223 for each epoch.

Figure 3

Iteration 1 Accuracy and Loss Curve



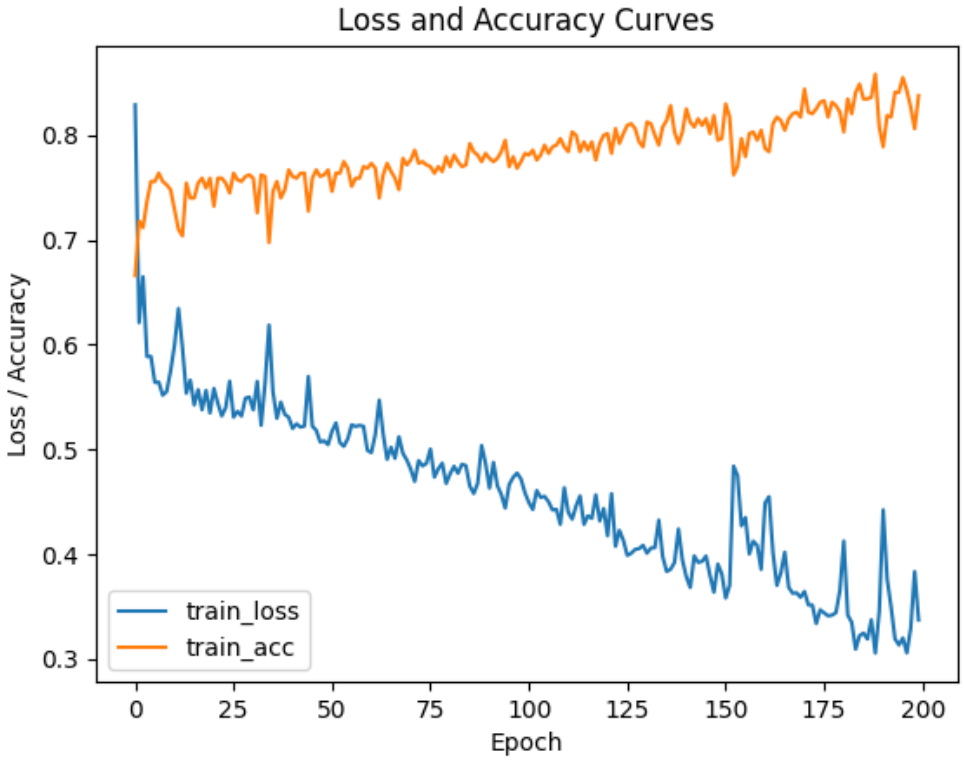
Iteration 2.

In the second iteration, a Simple RNN layer with 40 nodes and a ReLU activation function was implemented, as was a Dense layer with 16 nodes and a ReLU activation function, and a Dense output layer with one node and a sigmoid activation function. The model was trained over a period of 200 epochs. The accuracy and loss curves increased and decreased every few epochs at random amounts. The accuracy curve generally increased steadily with its lowest point of 0.6661 at the first epoch and its highest point of 0.8551 at epoch 196. The loss curve showed a huge spike downwards from 0.8290 to 0.6211 at the first epoch and began to decrease steadily until the spike up from 0.3706 to 0.4843 at epoch 153. The loss curve had its lowest point of 0.3060 at epoch 197 and its highest point of 0.8290 at the first epoch. The accuracy

curve ended at 0.8349, and the loss curve ended at 0.3456. The two curves showed an inverse relationship throughout the training process.

Figure 4

Iteration 2 Accuracy and Loss Curves



Iteration 3.

A SimpleRNN layer with 50 nodes and a ReLU activation function, as well as a Dense layer with 128 nodes and a ReLU activation function, and a Dense output layer with one node and a sigmoid activation function, were implemented in the third iteration. The model was trained over a 200-epoch period. The accuracy and loss curves increased and decreased every few epochs at random amounts. The accuracy curve generally increased steadily, with its lowest point of 0.6268 at the first epoch and its highest point of 0.9433 at epoch 183. The last accuracy value was 0.9417. The loss curve showed a huge spike downward from 1.2918 to 0.6207 and

then began to decrease steadily. The loss curve had its lowest point of 0.1453 at epoch 200 and its highest point of 1.2918 at the first epoch. The two curves showed an inverse relationship throughout the training process. The final model architecture (Figure 6) and tables of the adjustments made for each iteration (Table 1) and the accuracy and loss for each iteration (Table 2) are shown.

Figure 5

Iteration 3 Accuracy and Loss Curves

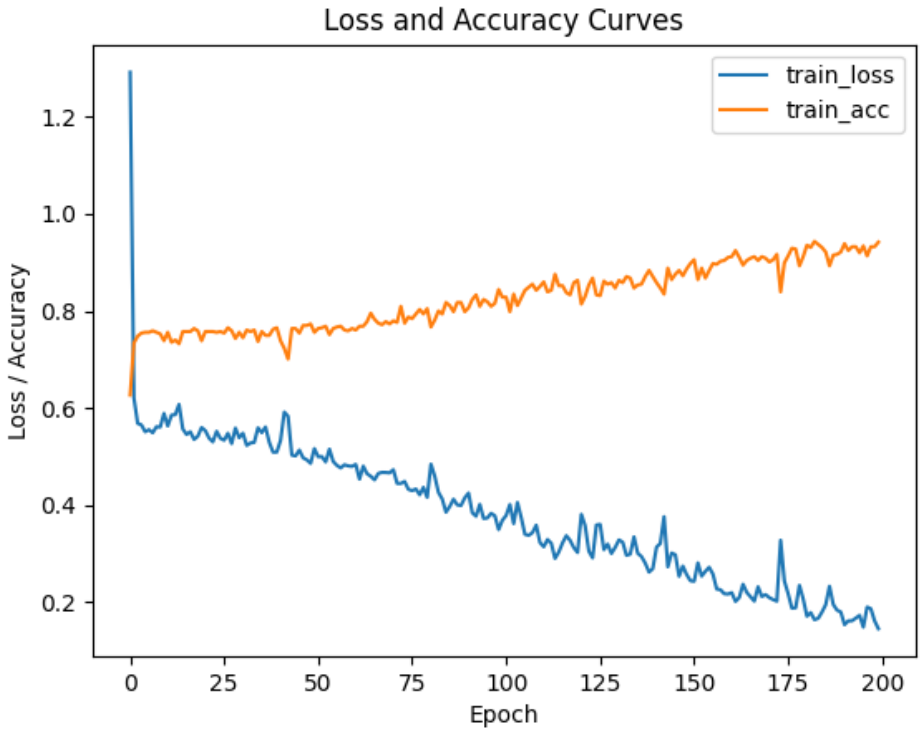


Figure 6

Final Model Architecture

```
43  model = tf.keras.Sequential([
44      tf.keras.layers.SimpleRNN(50,activation="relu"),
45      tf.keras.layers.Dense(128, activation="relu"),
46      tf.keras.layers.Dense(1, activation="sigmoid")
47  ])
```

Table 1

Adjustments for Each Iteration

Iteration #	Dense Layers #	Epochs	SimpleRNN Layer		First Dense Layer		Second Dense Layer	
			Activation Function	Nodes #	Activation Function	Nodes #	Activation Function	Nodes #
1	1	100	ReLU	50	ReLU	1	-	-
2	2	200	ReLU	40	ReLU	16	Sigmoid	1
3	2	200	ReLU	50	ReLU	128	Sigmoid	1

Table 2

Accuracy and Loss for Each Iteration

Iteration #	Accuracy %	Loss
1	75.99	3.7223
2	83.49	0.3456
3	94.17	0.1453

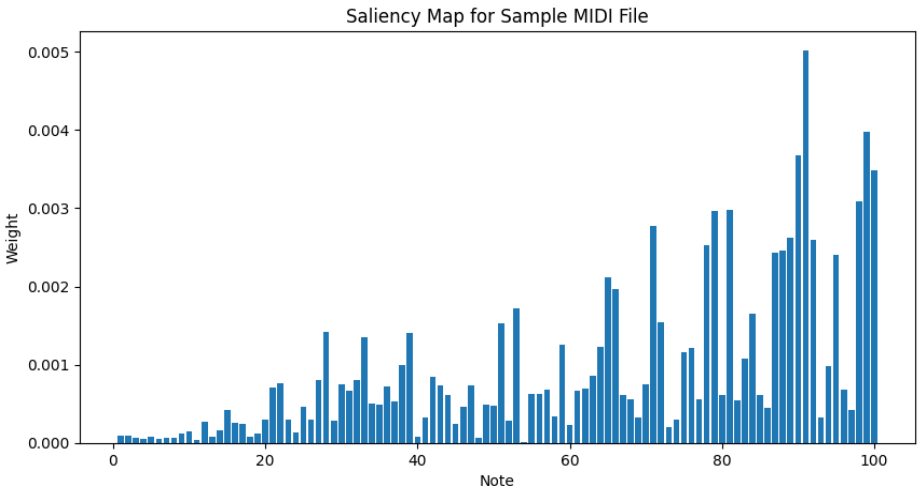
Creation of Saliency Map and Input Optimization Note Map

Saliency Map

A sample jazz MIDI file was accessed by the code through the directory of the computer that this study was performed on. The RNN from the first design problem correctly predicted the file to be jazz. To compute the saliency map, TensorFlow's automatic differentiation functionality was used, which enables us to compute the gradient of the model's output with respect to the input sequence. The gradients were then converted to a numpy array, and the absolute values were taken to get the magnitudes of the gradients. Finally, the saliency map was obtained by selecting the gradient magnitudes of the MIDI file input. The saliency map is displayed in the form of a 100 value array, a weight for each note in the input MIDI sequence (Figure 7). Although the weights increase and decrease randomly, they have a slight positive relationship with the note number. The lowest weight is 0.000007 at note 54, and the highest weight is 0.005 at note 91. The data shows a left skew, with most of the larger weights accumulating at the larger note values and the smaller weights accumulating at the smaller note values.

Figure 7

Graph Displaying Weight of Each Note in Saliency Map



Input Optimization Note Map

To minimize the difference between the predicted output label (jazz) and the target label (electronic), the code used an Adam optimizer and a loss function. The optimization was accomplished by adjusting a delta variable added to the input sequence. With a learning rate of 0.1, the code was run for 11 iterations. Following the completion of the optimization, the delta variable was converted to a numpy array. The resulting delta variable was appended to the original jazz MIDI file sequence to generate a new sequence, which was fed into the model to generate a new prediction. The predicted output label for both the original input sequence and the input sequence with the optimized delta was then obtained. Before optimization, the RNN classified the sample MIDI file as 1.00 (jazz). After optimization, the file was classified as 0.45 (electronic). “Input sequence + delta” represents the altered MIDI file after optimization, and “Delta” represents the difference in note pitch between the original MIDI file and the altered file (Figure 9). The delta data randomly changes value by large amounts and has a generally positive association with note number. The delta values gradually get farther from zero at the end of the note range.

Figure 8

Note Map of the Original MIDI File

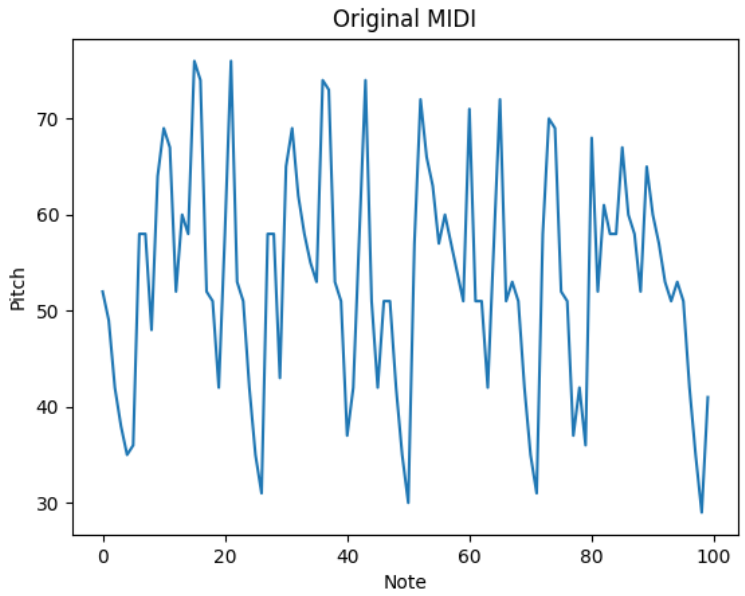
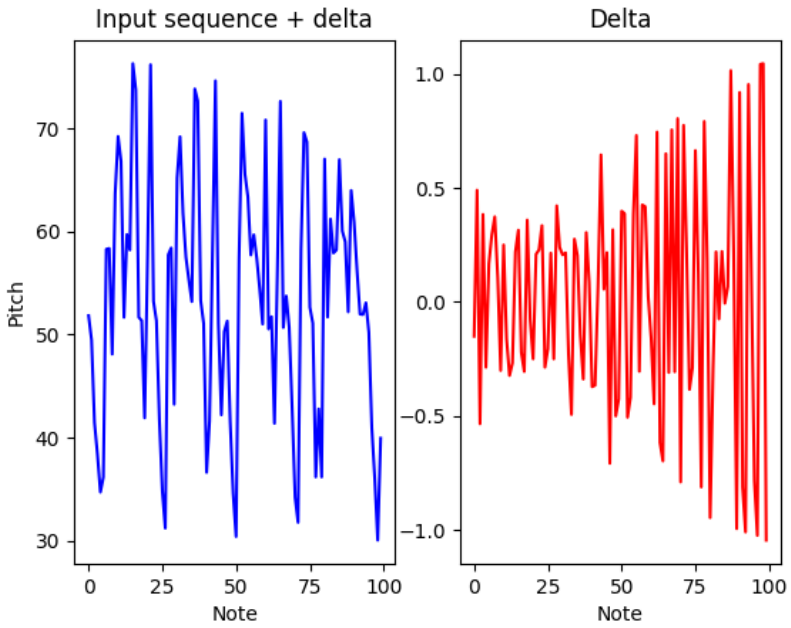


Figure 9

The Note Map of the Altered MIDI File and Delta Values



Conclusions

This study aimed to address the basic subproblems of MIDI file attributes that an RNN could detect as well as a realistic method of measuring a neural network's accuracy. RNNs have been shown in the literature to detect recurring melodic motifs, rhythmic patterns, and other aspects of a given musical genre. The RNN can analyze the pace of the music to determine how quickly the notes are performed and to distinguish between music from different eras or genres (Wang & Sohail, 2022). According to the literature, accuracy is one of the best measures for determining a neural network's efficacy in classification tasks (Bhardwaj et al., 2018). Two design subproblems were also devised. The first design subproblem centered on the development of an RNN capable of performing music genre categorization with an accuracy of at least 90%. The second design subproblem sought to perform input optimization and saliency mapping on a sample MIDI file identified using the first design subproblem's model. Literature indicates that saliency mapping and input optimization are valid ways to investigate a neural network’s interpretability (Chowdhry, 2021).

The RNN classification accuracy was 94.17% after executing the first design subproblem. The final design required three iterations, with each iteration improving accuracy by adjusting the parameters of dense layers, activation functions, nodes, and epochs. The results met the subproblem's requirement of achieving at least 90% accuracy in music genre classification. The second design subproblem was likewise successfully completed, including the creation of the input optimization note map and saliency map. These results met the requirement of creating an input optimization note map and saliency map.

Limitations

The RNN model and training method had numerous limitations. This study solely looked at an RNN's capacity to distinguish between electronic and jazz music, disregarding other genres. This limits the model's generalizability to music genre classification in general. Another limitation was the dataset, which contained a total of 642 MIDI files, each of which was shortened to 100 notes in order to ensure reasonable training time. A larger dataset with longer tracks may yield better results.

Implications

The purpose of this study was to add to the current literature and my mentor’s research on the explainability of RNNs for music genre classification. Although an RNN was successfully developed, as well as a saliency map and an input optimization note map, the extent to which these results can be interpreted is limited.. The saliency map revealed no obvious patterns that could have been predicted, and the input optimization note map revealed that the altered MIDI file changed note values at random, showing that the RNN did not completely learn the difference between the electronic and jazz genres. This was most likely due to the complexity of the model required to attain accuracy greater than 90%. The extent to which a model can be explained reduces as its complexity increases (Bhardwaj et al., 2018). This study demonstrated an RNN with extraordinarily high accuracy that could not be perceived as a disadvantage. This study investigated how the complexity of the RNN required for successful genre classification impacts its explainability. This lays the groundwork for the implication that, for the time being, music genre classification is too complex to be explained by an RNN.

Future Studies

The success of this RNN model's accuracy provides the framework for many future investigations. For example, rather than only electronic and jazz, future research may look into RNNs' capacity to classify diverse genres. This could add to the current body of knowledge about the explainability of RNNs for music genre classification. Another feasible study might compare different RNN models' capacity to conduct music genre classification. This study only looked at one model; utilizing additional models could provide more information about the explainability of these RNNs. More work could be done with exceptionally huge datasets to improve the chances of high classification accuracy. This study only used saliency mapping and input optimization as interpretability approaches, although other methods, such as spectrography, are available (Chowdhry, 2021). To further investigate the explainability of RNNs, audio formats such as MP3 files could be used instead of MIDI data, and spectroscopy could be used to analyze the results. Using these methods could result in data that can be interpreted.

References

Bhardwaj, A., Di, W., & Wei, J. (2018). *Deep learning essentials : your hands-on guide to the fundamentals of deep learning and neural network modeling*. Packt.

Chowdhry, A. (2021, May 7). *Music genre classification using CNN*. Clairvoyant. https://www.clairvoyant.ai/blog/music-genre-classification-using-cnn

Cui, Y., & Wang, F. (2022). Research on audio recognition based on the deep neural network in music teaching. *Computational Intelligence & Neuroscience*, *2022*, 1-8. https://doi.org/10.1155/2022/7055624

Fan, M. (2022). Application of music industry based on the deep neural network. *Scientific Programming*, 1-6. https://doi.org/10.1155/2022/4068207

Gao, H. (2022). Automatic recommendation of online music tracks based on deep learning. *Mathematical Problems in Engineering*, *2022*.

Ghotra, M. S., & Dua, R. (2017). *Neural network programming with tensorflow: Neural networks and their implementation decoded with tensorflow.* Packt Publishing.

He, Q. (2022). A music genre classification method based on deep learning. *Mathematical Problems in Engineering*, *2022*,.

Wang, W., & Sohail, M. (2022). Research on music style classification based on deep learning. *Computational & Mathematical Methods in Medicine*, *2022*, 1-9. https://doi.org/10.1155/2022/3699885

Wang, X., Jin, C., & Zhao, W. (2018). Beijing opera synthesis based on straight algorithm and deep learning. *Advances in Multimedia*, *2018*.

Yu-Huei Cheng, Pang-Ching Chang, Duc-Man Nguyen, & Che-Nan Kuo. (2021). Automatic music genre classification based on CRNN. *Engineering Letters*, *29*(1), 1-5.