RESEARCH OF APPLYING RNN(RNN-LSTM), CNN(RESNET-18), AND MNN ON MUSIC GENRE CLASSIFICATION PROBLEM

Yujin Qin
UMass Amherst
yujinqin@umass.edu

Tengzhi Zhuo
UMass Amherst
tzhuo@umass.edu

Haochen Ren
UMass Amherst
haochenren@umass.edu

Abstract

This research project compares multiple approaches to music genre classification problem by integrating Recurrent Neural Networks (RNN-LSTM), Convolutional Neural Networks (CNN-ResNet18), as well as the Multi-layer Neural Networks (MNN) algorithms. Considering that CNN and MNN may neglect the complex relationships between sequences and hierarchical patterns in music, we want to compare the performances between RNN, CNN and MNN to figure out whether it is necessary to take the time sequence into consideration while classifying music genre.

We employ a diverse GTZAN Dataset, which contains multiple music genres as well as segmented audio files with the same time lengths to evaluate the effectiveness of our approach. The training process consists of optimizing the loss function to guide the model in learning a robust representation of musical genres. We hope that by exploring which model can provide more accurate classification, so that It can be used not only as a powerful classifier, but also as a tool to explore and understand the intrinsic structure of different music genres.

The results of our study indicate that both Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) achieved better performance than Multi-layer Neural Networks (MNNs) in terms of classification accuracy . After further investigating the comparative analysis of RNNs and CNNs, we found that CNNs would achieve higher average accuracy than RNNs.

In conclusion, our research presents a comprehensive and effective approach to music genre classification, as well as the demonstration of the adaptability of integrated RNNs algorithms for timestamped datasets. We hope that this work will help the continued development of the music information retrieval field and further explore the importance of deep learning architectures in audio analysis field in the future.

1. Introduction

In these modern times, with an overwhelming volume of music available online, classifying and organizing this vast musical content is crucial for both music enthusiasts and content providers. Especially when we talk about this kind of art that relies more on the senses and feels, because everyone feels differently about music. Music is more difficult to categorize through technology than other issues. For example, a song selection may contain a blend of several different musical styles, such as pop music, country music, or jazz, making it difficult to easily categorize the selection.

As music lovers, we are passionate about improving the technical issues related to music and providing a purer, cleaner and more comfortable music environment for other music lovers.

The main focus of this research is to compare three different neural network algorithms on classification of music genre. One important aspect of our investigation involves a comparative analysis of the adaptability between Recurrent Neural Networks (RNN-LSTM), RESNET-18 of Convolutional Neural Networks (CNN-ResNet19), and Multi-layer Neural Networks (MNN). Our primary goal is to find out which model exhibits superior performance in classifying the music genre based on music clips, thus providing valuable insights into their respective strengths and applicability to the unique challenges posed by classifying music .

In order to gain a deeper understanding of the research problem and develop a clearer solution strategy, we conducted an extensive review of academic papers in the field of machine learning. A comprehensive exploration of the existing literature in the field played an important role

in forming the theoretical foundation of our research. This section is explained in detail in the literature review below.

Through our ongoing research, we aim to provide music lovers with instant genre identification of their favorite music and enable them to reap a better musical experience.

2. Literature Review

While exploring the suitable algorithms for music classification, we examine two papers that mainly combining CNNs and RNNs for effective model training and evaluation in music classification scenarios. One of the papers emphasized that CRNNs (Convolutional Recurrent Neural Networks) show a strong performance with respect to the number of parameter and training time, indicating the effectiveness of its hybrid structure in music feature extraction and feature summary [2]. In Lijiang's paper, she also mentioned the combination of RNNs and CNNs.

Given the existing literature's general emphasis on Convolutional Recurrent Neural Networks (CRNNs), we aim to observe deeper into CNNs, RNNs, and MNNs separately. Therefore, we conducted our research questions to investigate and compare the contributions and accuracies of CNNs, RNNs, and MNNs in music classification field.

3. Methodology

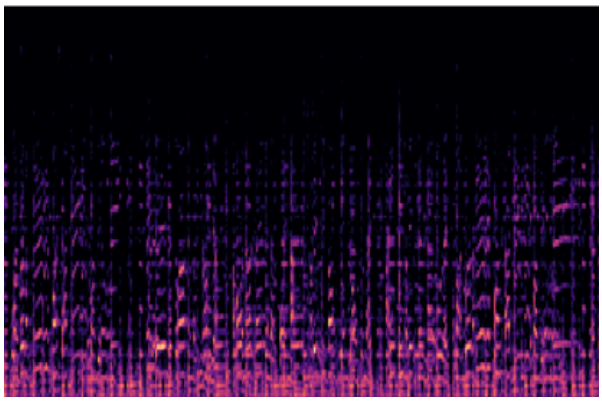
3.1. Data Collection and Description

The dataset utilized in this study was obtained from an online source, which is listed in the Dataset Resource section below. This dataset was chosen because of the need for a clearly timestamped and comprehensive dataset that met the research objectives.

The original collection of this dataset has 3 distinct datasets for different model using. The image dataset contains 100 audio files (images) of each of the 10 genres: blues, classical, country, disco, hip hop, jazz, metal, pop, reggae, and rock. One examples of blue genre music is listed below.

In addition to the original collection, this dataset also includes two CSV files containing features from the audio files. The $features\_30\_sec.csv$ file contains the mean and variance of each 30-second song, which is calculated from multiple features extracted from the audio files, such as the chroma feature (describes the angle of pitch rotation as it traverses the helix [1]) and the spectral feature (an acoustic musical practice where compositional decisions are often informed by sonographic representations and mathematical analysis of sound spectra [7]). An example of unprocessed

Figure 1. Visual Representation of a Blue Genre Music



data in $features\_30\_sec.csv$ file is provided below for a intuitive understanding.

	Chroma_stft Mean	Chroma_stft Var
1	0.3354	0.0910
2	0.3431	0.0861
3	0.3468	0.0922
4	0.3636	0.0869
5	0.3356	0.0881

Table 1. Unprocessed Data in $features\_30\_sec.csv$

The other $features\_3\_sec.csv$ file has the same structure, also contains the mean and variance, but the songs have been previously segmented into 3-second audio files to obtain more extensive data to improve the model performance.

3.2. Dataset Source

The dataset was sourced from KAGGLE, which is a up-to-date online platform providing massive and high quality dataset.

3.3. Data Preprocessing

Due to the utilization of three distinct algorithms and specific data requirements inherent to each raw dataset, different approaches were applied on each of the algorithms. In the pivotal step of splitting the dataset into training and testing sets, a standardized ratio of 8:2 was adopted. 800 instances were used for model training and 200 instances were used for model testing. For a fair and comprehensive analysis, a equal number of songs were selected for each category to promote a balanced representation of all genres.

3.3.1 CNNs: RESNET-18

CNNs requires the data to primarily focus on the dimensions of images. The raw image data has an original shape of $(999, 288, 432, 4)$, where it initially had 4 channels. However, the 4^{th} channel was intentionally removed as a result of its null standard deviation. After this removal, the dataset had a shape of $(999, 288, 432, 3)$, and the standard normalization method was subsequently applied to each channel. For each channel, the Z-score normalization function, often referred as the standard normalization, used is given by:

$$\frac{\text{value} - \text{mean\_value}}{\text{std\_value}}$$

The output data was initialized as a numpy array with all zeros and a shape of $(999, 10)$, corresponding to the 10 categories of songs. Each music label was assigned a 1 for its correct category and 0 for others.

3.3.2 MNNs

In the data preprocessing phase of the MNNs, we mainly focused on sound clips rather than image data, which is consistent with the properties of MNNs. The original dataset had a shape of $(1000, 57)$, with each song containing 57 features. The output data was initialized as a numpy array with all zeros and a shape of $(999, 10)$, corresponding to the 10 categories of songs. Each music label was assigned a 1 for its correct category and 0 for others.

To ensure uniformity and enhance model convergence, the mean value and standard deviation for each of the 57 features were calculated and the Z-score normalization technique was applied to the entire dataset for each feature. This approach not only harmonizes the proportion of each feature, but also guarantees the robustness and unbiased characteristic of the training model.

3.3.3 RNNs: LSTM

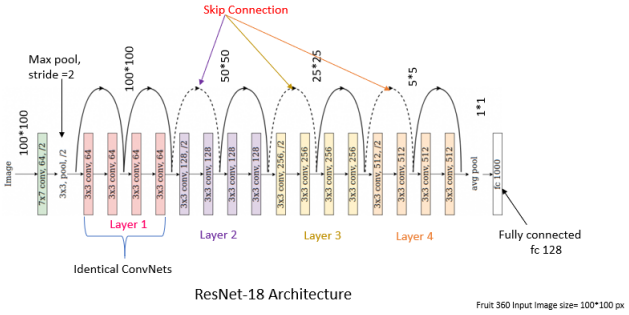
In the context of Recurrent Neural Networks (RNNs), we remained focusing on the sound clips rather than image data, maintaining continuity with the methodology suitable for RNNs. However, in contrast to MNNs, each sound segment was subdivided into 9 smaller and equal segments.

The original dataset had a shape of $(1000, 9, 57)$, with each song containing 9 segments and each segment contains 57 features. The normalization steps applied to the feature set remained consistent with those employed in the preprocessing of MNNs. Mean values and standard deviations for each of the features across the sound segments were computed, and the Z-score normalization technique was applied.

3.4. Model Building

3.4.1 CNNs: RESNET-18

Figure 2. Structure of RESNET-18 Algorithm



Residual Neural Network, often referred as RESNET, is a deep learning model in which the weight layers learn residual functions with reference to the layer inputs [6]. In this project, we chose to use RESNET-18, which has 18 deep layers and provides more accurate model. In order to make the model consistent with the task of classifying music genres in 10 different categories, the last layer was modified to produce a 10x1 predicted output. Moreover, a Softmax nonlinear function was added to the end of the network because the Softmax functions are particularly suitable for classification tasks, especially in scenarios involving multi-class classification problems. As a result, this addition not only helps to interpret the output as a probability score, but also ensures that the range of values is [0, 1]. The introduction of the Softmax function enhances the interpretability and usefulness of the model predictions, making it well suited for classification tasks such as music genre identification.

The architecture of the RESNET-18 was designed with the following layers:

models.resnet18() Pytorch Resnet18, weights = ResNet18_weights.IMAGENET1K_V1

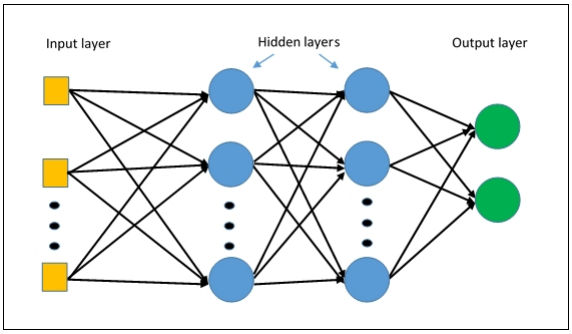
models.resnet18.fc() Modified the last layer. Last layer of our model contains one linear function, and one softmax function.

In the context of hyperparameter tuning, 4 distinct learning rates, $6e-6, 8e-6, 1e-5, 3e-5$, were tuned in order to find the best accuracy. For the final result, we used the hyperparameters listed below to achieve an optimal accuracy around 0.698.

Learning Rate 1e-05

3.4.2 MNNs

Figure 3. Structure of MNNs Algorithm



The Multi-layer Neural Networks (MNNs) contain two and more layers of artificial neurons or nodes. In this project, particularly, we selected the Multi-layer Perceptron algorithm, which consists of fully connected neurons with a nonlinear kind of activation function, organized in at least three layers [5]. The architecture of the MLP was designed with the following layers using PyTorch:

nn.Linear(57, 64) Input layer, also linear layer, with 57 input features and 64 output neurons

nn.LeakyReLU() Activation function, preventing dead neurons

nn.Linear(64, 128) Linear layer, with 64 input features and 128 output neurons

nn.LeakyReLU() Activation function, preventing dead neurons

nn.Linear(128, 10) Output layer, also linear layer, with 128 input features and 10 output neurons

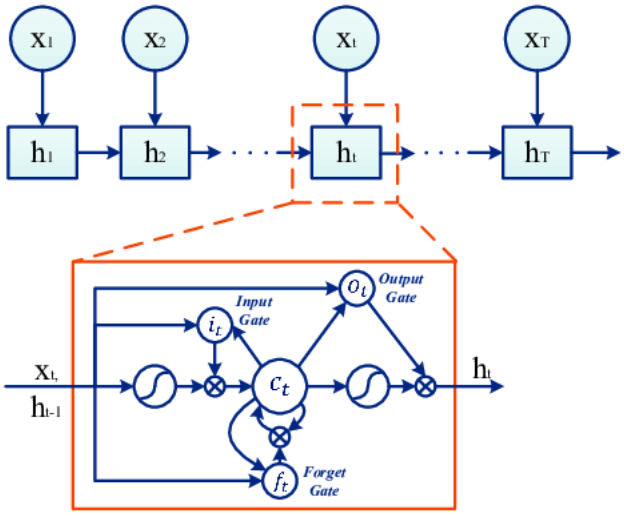
nn.LogSigmoid() Activation function, ensuring output to be within [0, 1]

In the context of hyperparameter tuning, 5 distinct learning rates, $1e-5, 3e-5, 5e-5, 8e-5, 1e-4$ were optimized in order to find the best accuracy. For the final result, we used the hyperparameters listed below to achieve an optimal accuracy around 0.565.

Learning Rate 8e-05

3.4.3 RNNs: LSTM

Figure 4. Structure of RNNs: LSTM Algorithm



Recurrent Neural Networks (RNNs) are a class of artificial neural networks specialized for processing sequential or timestamped data. Particularly, the Long Short-Term Memory (LSTM) is a specific Recurrent Neural Networks (RNNs) architecture that was designed to model temporal sequences and their long-range dependencies more accurately than conventional RNNs [4]. The LSTM architecture is well-suited to our timestamped dataset and was chosen primarily due to its proficiency in mitigating the vanishing gradient or exploding gradient issues and its superior capability in preserving past temporal information. The architecture of the RNNs: LSTM was designed with the following layers using PyTorch:

nn.LSTM(57, 128, 3) Input layer, also LSTM layer, with 57 input features and 128 hidden layer neurons, number of layers equal to 3

nn.Linear(128, 10, bias=True) Output layer, with 128 input neurons and 10 output neurons

nn.LogSigmoid() Activation function, ensuring output to be within [0, 1]

In the context of hyperparameter tuning, learning rate, the number of nodes in each hidden layer, and the number of hidden layers were optimized in order to find the best accuracy. For the final result, we used the hyperparameters listed below to achieve an optimal accuracy around 0.6.

Learning Rate 8e-05

Number of nodes in each hidden layer	64
Number of hidden layers	3

4. Result

The results of this project are mainly evaluated by the analysis of model accuracy. After model training, we can see that both CNN and RNN achieved optimal accuracy over 0.6, but MNN only gained optimal accuracy of 0.565. This proves that MNN is not efficient enough for music classification problem. In addition, CNN optimal accuracy was 0.698, approaching to 0.7, but RNN optimal accuracy was 0.6. In this case, CNN would be suggested due to its higher accuracy.

The correctness and rationality are mainly evaluated by the decreasing cost history graphs. The cost history of 3 distinct models are attached below. One thing need to be mentioned is that the cost history plot of CNN and RNN have zigzag trends. This may be caused by the imperfections of our original models and noise of data.

This result is in line with what we've come to expect from the results and it's not surprised. Because although the music dataset is suitable for the RNN algorithms, the complexity of the underlying patterns of dataset could still influence the model performance.

Figure 5. Cost History of CNNs: RESNET-18

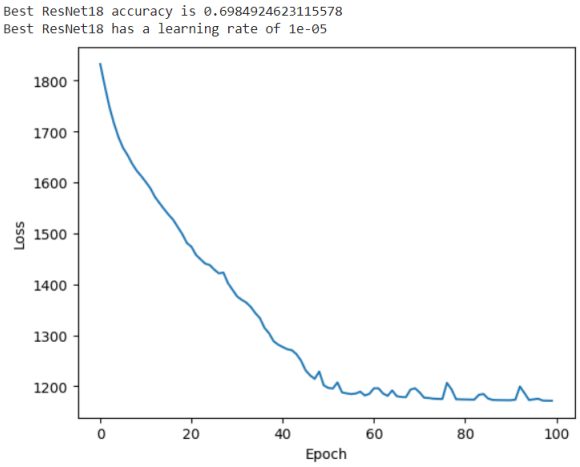


Figure 6. Cost History of MNNs

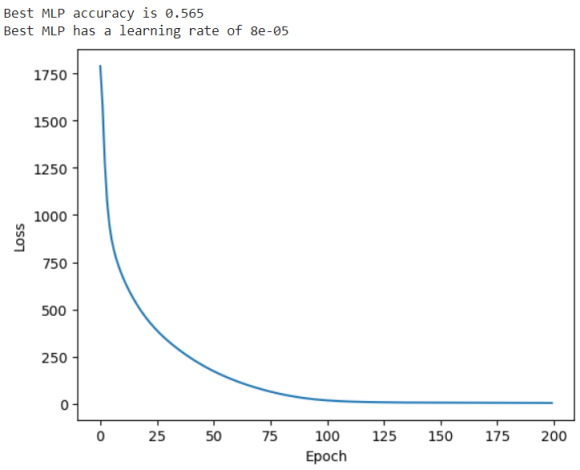
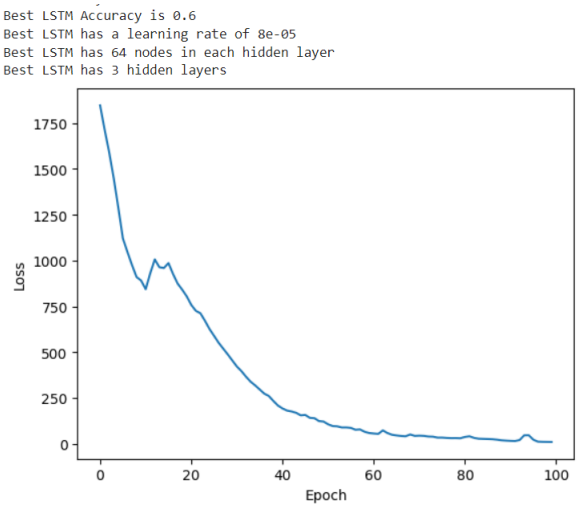


Figure 7. Cost History of RNNs: LSTM



5. Conclusions and Recommendations

In summary, the RNN and CNN gained better performance than MNN. Within the choice of RNN and CNN, we would more suggest CNN due to better performance and higher accuracy.

After the research, our findings demonstrated that both RNN and CNN have their own strength. As the music classification problem needs to focus both timestamped data and sound image data, combining them will lead to more accurate result. Just like the conclusion in previous research, particularly Lijiang Feng's paper, her findings emphasized that in contrast to utilizing CNN alone, all of the CNN with paralleling RNN can improve the performance of music genre classification [3].

The potential applications of this classification model could be online music applications such as Spotify and Apple Music, where the music recommendation functionality can significantly increase the user experience. By exploring and observing the complex patterns of user listening behaviors, the model can learn what kind of genre the use like, thus provide more targeted and specific music recommendations based on music genres. This personalized approach has the potential to not only engage users, but also create their connection to the platform, ultimately the user experience.

Going forward, the further extensions of this project could involve more normalization approaches rather than simply employing the standard normalization for every model. Considering alternative approaches such as PCA-Sphereing or no normalization could provide a more comprehensive understanding of the model's behavior under different preprocessing conditions. However, as we considered that these methods were more resource intensive, they were not employed in this project so far. Also, we can try to modify the structures in our three models to try to get a higher accuracy. For example, we can try to apply multiple LSTM layers in the RNN model to compare the results. Other than that, we can try to combine CNN and RNN to train a more powerful model. Hopefully, the potential insights gained from the implementations can optimize and further improve the accuracy and robustness of the model. This extension of research can provide valuable considerations for fine-tuning classification models in the real world.

References

[1] Juan P. Bello. Six tonalities in pop/rock music, 2018. Accessed: December 6, 2023.

[2] Keunwoo Choi, Gyorgy Fazekas, Mark Sandler, and Kyunghyun Cho. Convolutional recurrent neural networks for music classification. *Queen Mary University of London, Centre for Digital Music*, 2017.

[3] Lijiang Feng, Shuo Liu, and Jianwei Yao. Music genre classification with paralleling recurrent convolutional neural network. 2017.

[4] Alex Graves, Abdel rahman Mohamed, and Geoffrey E. Hinton. Speech recognition with deep recurrent neural networks. *CoRR*, abs/1303.5778, 2013.

[5] Wikipedia contributors. Mlp definition, Year of the last edit. Accessed: December 7, 2023.

[6] Wikipedia contributors. Resnet definition, Year of the last edit. Accessed: December 7, 2023.

[7] Wikipedia contributors. Spectral music, Year of the last edit. Accessed: December 6, 2023.