Music Genres Classification with Convolutional Neural Networks

Xincheng Lv
School of Computer Science
Nanjing University of Posts and Telecommunications
Nanjing, China
1023040921@njupt.edu.cn

***Abstract*—Music genre classification is a challenging task in the field of audio signal processing. This paper presents a novel approach to music genre classification using Convolutional Neural Networks (CNNs) and extracting audio features with the Librosa library. The dataset which we used consists of various music genres, and the proposed model is designed to effectively classify these genres based on learned features. The comprehensive assessment includes training and validation accuracy, along with test accuracy, providing insights into the model's capabilities. Visualizations of loss curves and accuracy trends over epochs offer a deeper understanding of the learning dynamics of the model. Furthermore, a normalized confusion matrix is presented, depicting the model's proficiency in predicting music genres within the dataset. This matrix is visually represented using a heatmap, offering an intuitive portrayal of the classification results. In summary, the proposed method effectively demonstrates the effectiveness of CNN in music genre classification, and the results highlight the model's accurate classification of various music genres.**

***Index Terms*—music genre classification, convolutional neural networks , features, accuracy**

I. Introduction

The field of music classification has witnessed notable developments over the years. While the concept of categorizing music based on its inherent characteristics has been explored for decades [1], significant strides have been made more recently with the advent of advanced machine learning techniques.

In particular, Convolutional Neural Networks (CNNs) have emerged as a powerful tool for music classification tasks. Although CNNs have roots dating back to the 1980s, their widespread adoption for object classification tasks, including music genre classification, gained momentum in the early 21st century. The pivotal work by Krizhevsky et al. in 2012 marked a turning point, demonstrating the effectiveness of CNNs in large-scale visual recognition [2].

Since then, the landscape of music classification has undergone a transformation, with Convolutional Neural Networks (CNNs) playing a pivotal role in propelling the field forward. These networks have been instrumental in achieving significant milestones, accurately categorizing music genres, and discerning intricate musical characteristics [3], [4]. The transition from manual feature engineering to automated feature learning, facilitated by the prowess of CNNs, has markedly improved the precision and efficiency of music classification systems. This paradigm shift mirrors a broader trend in pattern recognition tasks, emphasizing the versatility and effectiveness of deep learning techniques in pushing the boundaries of state-of-the-art music classification methodologies.

Drawing inspiration from these advancements, our paper proposes a novel approach to music classification. By harnessing the capabilities of Convolutional Neural Networks (CNNs) and leveraging automated feature learning, our methodology aims to further enhance the accuracy and effectiveness of music genre classification. In contrast to conventional methods, our strategy embraces the potential of deep learning, elevating the state-of-the-art in the intricate task of music analysis.

II. Methods

A. Mel-frequency cepstral coefficients

Mel-frequency cepstral coefficients (MFCCs) are a prevalent feature representation in audio signal processing, especially in speech and music analysis [5], [6]. They efficiently capture the spectral characteristics of an audio signal by representing the short-term power spectrum in a manner aligned with human auditory perception. The extraction process involves applying the Mel filterbank to the power spectrum, logarithmically compressing filterbank energies, and applying the discrete cosine transform (DCT) to obtain the cepstral coefficients.

In this study, we harness the effectiveness of Mel-frequency cepstral coefficients for audio feature extraction. The extraction process is facilitated through the use of the Librosa library, a powerful Python package for music and audio analysis. Librosa simplifies the computation of MFCCs through functions like librosa.feature.mfcc(), enabling us to leverage the discriminative power of MFCCs in the task of music genre classification as part of our proposed methodology.

B. Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) represent a class of deep neural networks specifically designed for tasks involving grid-like data, such as images and, more broadly, multidimensional arrays. CNNs have become a cornerstone in the field of computer vision and pattern recognition due to their ability to automatically learn hierarchical representations of data [7].

Key components of CNN architecture include convolutional layers, pooling layers, and fully connected layers [8]. Convolutional layers employ filters to convolve across input

data, capturing local patterns and spatial hierarchies. Pooling layers down-sample the spatial dimensions, reducing computational complexity while preserving essential features. Fully connected layers, often present in the network's final stages, learn global patterns and contribute to the network's decision-making process.

CNNs excel in feature extraction from complex data, making them particularly effective in image recognition, object detection, and, increasingly, various domains beyond visual data. Their adaptability and success in capturing hierarchical features make CNNs a powerful tool in tasks where understanding hierarchical structures is crucial, providing state-of-the-art performance in a variety of applications [9].

The following diagram illustrates a typical CNN network structure. The specific architecture employed in this study will be detailed in the subsequent section.

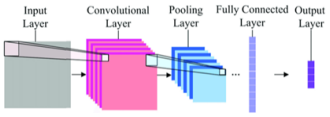

Fig. 1. Convolutional Neural Networks

C. Dataset

Our study employs the widely recognized GTZAN dataset for music genre classification. The GTZAN dataset comprises a diverse collection of audio tracks, spanning multiple genres such as rock, jazz, pop, and more. Each genre category is represented by a substantial number of audio samples, providing a robust and varied dataset for training and evaluation [10], [11].

The dataset is organized into a structured hierarchy, facilitating the process of loading and preprocessing. Prior to training our Convolutional Neural Network (CNN) model, we meticulously divided the dataset into training, validation, and test sets. This ensures a comprehensive evaluation of the model's performance across different subsets of the data.

In the subsequent sections, we provide a detailed introduction to our model architecture, training procedure, and evaluation metrics, offering a comprehensive understanding of our methodology and results.

III. Experiment

A. Preprocessing

Prior to training our Convolutional Neural Network (CNN), a crucial preprocessing stage is undertaken to ensure the input data is appropriately formatted and conducive to effective model training. This section outlines the key preprocessing steps applied to the GTZAN dataset.

First, we need to use the librosa library to load the GTZAN dataset, so as to achieve efficient organization and access of audio files. To facilitate the CNN training process, genre labels associated with each audio track undergo label encoding. The LabelEncoder from the scikit-learn library is applied, converting genre labels into numerical representations.

Critical to our model's success is the extraction of Mel-frequency cepstral coefficients (MFCCs) as audio features. Leveraging the capabilities of the Librosa library, this step ensures that the CNN receives essential information for discerning patterns within the audio data.

Also a well-organized dataset is pivotal for robust model training. The train_test_split function from scikit-learn partitions the preprocessed dataset into training, validation, and test sets. This division ensures a diverse range of data for training and comprehensive evaluation. To align with the input requirements of our CNN architecture, we reshape the feature matrices representing audio data. An additional dimension is introduced, ensuring compatibility with the Convolutional 1D layers in our model.

The following figures illustrate the waveform and visualization of Mel-frequency cepstral coefficients (MFCC) for a segment of audio from the GTZAN dataset.

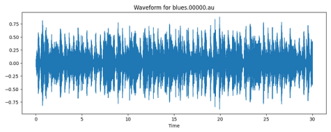

Fig. 2. The waveform of a segment of audio

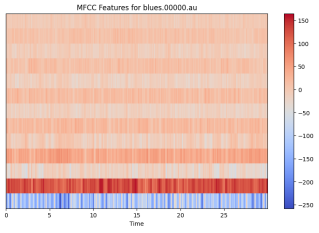

Fig. 3. Visualization of the Mel-frequency cepstral coefficients

B. The framework of the neural network

In this section, we explore the architecture of our neural network, explaining the layers and configurations of our framework.

Our neural network is carefully designed using the TensorFlow and Keras libraries, which are widely recognized for implementing advanced deep learning models in the Python ecosystem. Utilizing the sequential model enables a systematic layering approach, facilitating the smooth integration of various components within the neural network.

At the heart of our neural network architecture lies the application of Convolutional 1D layers for spatial feature extraction. The initial convolutional layer incorporates 64 filters, each with a kernel size of 3, complemented by a rectified linear unit (ReLU) activation function. Following this, a MaxPooling 1D layer, characterized by a pool size of 2, is introduced to efficiently down-sample spatial dimensions. Expanding on this foundation, the second convolutional layer features 128 filters with a kernel size of 3, followed by another MaxPooling

1D layer, further enhancing the refinement of feature extraction.The output is then flattened into a one-dimensional array, traversing a Dropout layer with a 30% dropout rate to mitigate overfitting. The subsequent dense layers contribute to global pattern recognition, with the penultimate layer hosting 128 neurons activated by a ReLU function. The final dense layer, activated by softmax, produces the output essential for music genre classification.

Configuring our model demands thoughtful considerations. We have opted for the Adam optimizer due to its proven efficiency, employing a learning rate set at 0.0001 to fine-tune the training dynamics. In light of our multi-class classification objective, we embrace sparse categorical cross-entropy as our chosen loss function, tailored to handle categorical data efficiently.

In the context of our multi-class classification task, the choice of an appropriate loss function is crucial.We choose the sparse categorical cross-entropy loss, which is well-suited for situations where each data instance is associated with only one class. This loss function mathematically evaluates the dissimilarity between the true class distribution and the predicted distribution, and it is calculated as the negative logarithm of the predicted probability assigned to the true class. The formula for sparse categorical cross-entropy is expressed as [12]:

$$H\left(y,\hat{y}\right)=-\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C}y_{i,j}\cdot\log\left(\hat{y}_{i,j}\right) \tag{1}$$

Here, 'y' represents the true class distribution, '$\hat{y}$' represents the predicted distribution, 'N' is the total number of instances, 'C' is the number of classes, 'i' indexes the instances, and 'j' indexes the classes. This formula calculates the negative logarithm of the predicted probability assigned to the true class for each instance, and the result is averaged over all instances and classes. This mathematical expression guides the model by quantifying the dissimilarity and steering the optimization process towards accurate music genre classification.

Our model undergoes extensive training, covering 1000 epochs on the training set. Validation data is essential for monitoring performance and ensuring the model's ability to generalize effectively. To prevent overfitting, we incorporate an EarlyStopping callback, which stops training if there's no improvement in the validation loss over 20 consecutive epochs. This strategic measure enhances the model's robustness, preventing it from learning noise or specificities present in the training data that may not generalize well to new, unseen data.

IV. RESULTS AND DISCUSSION

In this section, we present the results obtained from our proposed Convolutional Neural Network (CNN) model for music genre classification. We conduct a comprehensive analysis of the model's performance on the GTZAN dataset, examining key metrics and discussing the implications of our findings.

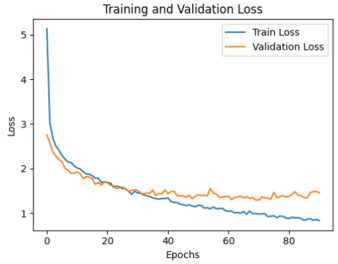

Fig. 4. Training and Validation Loss

We now shift our focus to a detailed examination of the model's training dynamics. The plotted curves showcase the evolution of the loss function over epochs. It can be seen that with the increase of epoch, the loss on both the training set and the validation set is decreasing, and under the effect of early stopping, the model ends training early to avoid the problem of overfitting. This convergence in loss values signifies the model's ability to learn and generalize well, showcasing its effectiveness in music genre classification.

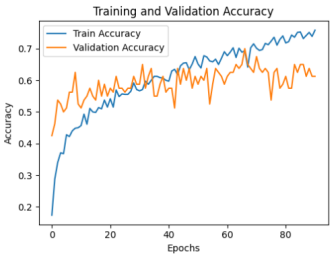

Fig. 5. Training and Validation Accuracy

The above graph illustrates the accuracy trends on both the training and validation sets. It is evident that with increasing epochs, the accuracy improves for both sets, showcasing the model's learning capabilities. However, it's noteworthy that the training set exhibits higher accuracy compared to the validation set, indicating that the model might have overfit to some extent on the training data. Nonetheless, the introduction of early stopping has played a crucial role, preventing the model from continuing training when there's no improvement in the validation set accuracy, effectively mitigating the risk of overfitting. The relatively gradual increase in accuracy on the validation set suggests that the model generalizes well to unseen data, achieving a balance between accuracy and robustness.

Here are the detailed outcomes from our experiments, specifically focusing on accuracy:

- Train accuracy: 75.8%
- Validation accuracy: 61.25%
- Test accuracy: 55.5%

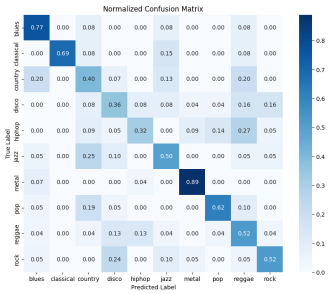

Fig. 6. Normalized Confusion Matrix

Figure 6 illustrates the Normalized Confusion Matrix, providing valuable insights into the model's performance across various music genres. The matrix offers a detailed breakdown of predicted genre labels against the actual genres present in the GTZAN dataset.

Analyzing the matrix, we observe higher accuracy in certain genres, notably blues, classical, and metal, as indicated by the diagonal elements. The color-coding of the matrix provides a visual representation, with darker shades indicating higher accuracy rates. Specifically, diagonal elements represent accurate classifications, while off-diagonal elements indicate instances where the model may have confused certain genres, leading to misclassifications. This visualization aids in identifying genres with strong classification performance, as well as those that pose challenges for the model.

In summary, the Normalized Confusion Matrix serves as a valuable tool for assessing the model's proficiency in genre classification, offering a nuanced perspective on its performance across the diverse range of music genres present in the GTZAN dataset.

Next, we will discuss the selection of parameters in our model and provide specific explanations for choosing these parameters.

The selection of the number of epochs is a crucial aspect in training a neural network. It determines the number of times the entire training dataset is presented to the network for learning. The choice of this parameter involves finding a balance between allowing the model to learn sufficiently and avoiding overfitting.

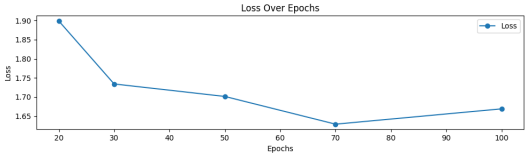

Fig. 7. Loss Over Epochs

In our experiments, we conducted tests with varying numbers of epochs, specifically 20, 30, 50, 70, and 100. Figure 7 illustrates the model's performance in terms of loss across these different epoch values. It is evident that as the number of epochs increases, the loss generally decreases. While there is a slight upward trend observed around 100 epochs, this can be considered a reasonable fluctuation. We will provide further explanations in the following sections.

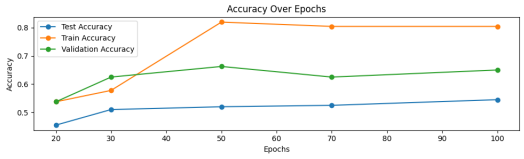

Fig. 8. Accuracy Over Epochs

Figure 8 demonstrates how accuracy changes on the training, validation, and test sets as the number of epochs varies. As mentioned earlier, there was a slight increase in loss around 100 epochs. However, by examining this graph, we can observe that the accuracy on all three sets continues to improve. Therefore, we believe that the occurrence of a slight increase in loss falls within a reasonable range.

Therefore, we ultimately decided to set the number of epochs to 1000, incorporating early stopping based on accuracy to prevent overfitting. Specifically, if there is no improvement in accuracy on the validation set for 20 consecutive epochs, the training process is halted. This careful strategy not only ensures effective training but also guards against overfitting, enhancing the model's generalization capabilities.

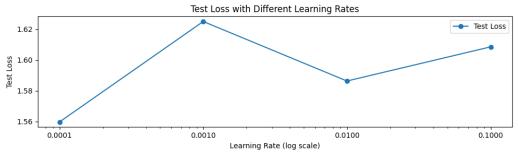

Fig. 9. Test Loss with Different Learning Rates

In Figure 9, we visualize the effects of varying learning rates on the model's loss. It is apparent that the minimum loss is attained when the learning rate is set to 0.0001. The decision to avoid selecting an even smaller value is driven by the acknowledgment that a further reduction in the learning rate might demand an increase in epochs for the model to converge. This trade-off aims to strike a balance between model efficiency and convergence requirements.

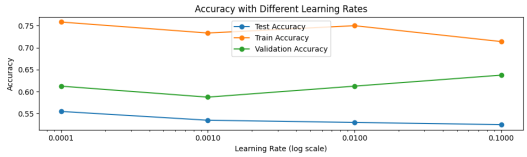

Fig. 10. Accuracy with Different Learning Rates

Figure 10 provides insights into the impact of different learning rates on the accuracy of the training, validation, and test sets. Once again, we observe that the highest accuracy is achieved when the learning rate is set to 0.0001. Taking a comprehensive perspective, we decide to set the learning rate

at 0.0001, considering both the minimization of loss and the maximization of accuracy across the datasets.

In addressing the concern of overfitting, we employ dropout as a regularization technique within our neural network. Dropout works by randomly excluding neurons during training, effectively disregarding their outputs [13]. This strategic dropout mechanism prevents the model from relying too heavily on specific neurons, promoting a more robust architecture capable of generalizing well to unseen data. In our model, a Dropout layer is introduced after the flattening layer, following the MaxPooling layer, with a dropout rate set at 30%. This thoughtful integration of dropout enhances the model's ability to generalize to new data, contributing to improved overall performance while mitigating overfitting risks [14].

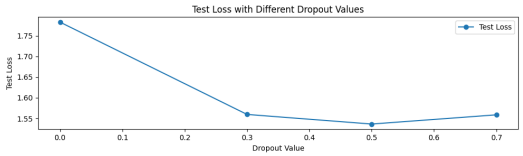

Fig. 11. Test Loss with Different Dropout Values

Figure 11 depicts the variation in loss under different dropout values. When dropout is not applied, the loss is initially at its maximum, decreasing as the dropout value increases. The minimum loss is observed when the dropout value is 0.5, followed by a gradual increase as the dropout value continues to rise.

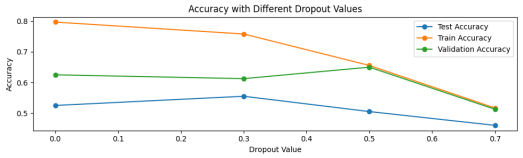

Fig. 12. Accuracy with Different Dropout Values

Describing the variation in accuracy under different dropout values is challenging. Figure 12 illustrates accuracy with different dropout values. While it was mentioned earlier that the minimum loss occurs when the dropout value is 0.5, the graph suggests that the best accuracy on the validation set aligns with this dropout value. Nevertheless, there is a significant decline in accuracy on the test set under the same condition. Taking a comprehensive view, we ultimately opted for a dropout value of 0.3 in our model.

V. SUMMARY

In conclusion, our Convolutional Neural Network (CNN) model, tailored for music genre classification, exhibits robust performance. By employing a thoughtfully designed architecture, leveraging Librosa for preprocessing, and fine-tuning key parameters, the model achieves promising accuracy. The selection of dropout values further enhances generalization, striking a balance between training and test set accuracy. This study offers valuable insights into the effective design of CNNs for audio data classification.

ACKNOWLEDGMENT

Through a semester of Big Data Analysis, I have acquired valuable professional knowledge and practical skills. The assignments throughout the course, covering topics like PCA (Principal Component Analysis), pattern mining, music classification, KMeans clustering, and DBSCAN algorithm, provided hands-on experience that deepened my understanding of these concepts. Among these, music classification captured my interest, leading me to choose it as the focus for my final project.

While I acknowledge the limitations of the model, particularly in achieving higher classification accuracy, I am eager to explore ways to enhance and improve in the future. I express gratitude to Professor Zou for the lessons throughout the semester.

REFERENCES

[1] McKinney M, Breebaart J. Features for audio and music classification[J]. 2003.

[2] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Advances in neural information processing systems, 2012, 25.

[3] Rajanna A R, Aryafar K, Shokoufandeh A, et al. Deep neural networks: A case study for music genre classification[C] 2015 IEEE 14th international conference on machine learning and applications (ICMLA). IEEE, 2015: 655-660.

[4] Choi K, Fazekas G, Sandler M, et al. Transfer learning for music classification and regression tasks[J]. arXiv preprint arXiv:1703.09179, 2017.

[5] Logan B. Mel frequency cepstral coefficients for music modeling[C] Ismir. 2000, 270(1): 11.

[6] Molau S, Pitz M, Schluter R, et al. Computing mel-frequency cepstral coefficients on the power spectrum[C]//2001 IEEE international conference on acoustics, speech, and signal processing. Proceedings (cat. No. 01CH37221). IEEE, 2001, 1: 73-76.

[7] Li Z, Liu F, Yang W, et al. A survey of convolutional neural networks: analysis, applications, and prospects[J]. IEEE transactions on neural networks and learning systems, 2021.

[8] O'Shea K, Nash R. An introduction to convolutional neural networks[J]. arXiv preprint arXiv:1511.08458, 2015.

[9] Gu J, Wang Z, Kuen J, et al. Recent advances in convolutional neural networks[J]. Pattern recognition, 2018, 77: 354-377.

[10] Sturm B L. An analysis of the GTZAN music genre dataset[C]//Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies. 2012: 7-12.

[11] Sturm B L. The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use[J]. arXiv preprint arXiv:1306.1461, 2013.

[12] Sharma A K, Aggarwal G, Bhardwaj S, et al. Classification of Indian classical music with time-series matching deep learning approach[J]. IEEE Access, 2021, 9: 102041-102052.

[13] Baldi P, Sadowski P J. Understanding dropout[J]. Advances in neural information processing systems, 2013, 26.

[14] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. The journal of machine learning research, 2014, 15(1): 1929-1958.