Music classification based on convolutional neural network

ZhuYunTao
1024040923
Nanjing University of Posts and Telecommunications
School of Computer Science
Nanjing,China

***Abstract*—With the rapid development of deep learning technology, Convolutional Neural Networks (CNNs) have shown significant performance improvements in music classification tasks. This article introduces how to use convolutional neural networks for effective music classification, which has practical significance. In Section II of the article, the five layer structure of convolutional neural networks is detailed to illustrate how CNN simulates the human brain to process data, as well as the feasibility and efficiency of CNN in image and audio processing. The data processing step of converting music audio into Mel spectral features is essential for using convolutional neural networks for music classification. At the same time, in order to increase the data volume, data augmentation was performed on the original data by doubling the data volume. Experiments have shown that this operation can significantly improve the accuracy of predictions and reduce overfitting. When building the training model, two convolutional layers and two pooling layers were constructed to extract higher-level and more useful features through two rounds of feature dimensionality reduction, while removing useless features. After training the model, corresponding indicators were used to evaluate it, and the final results showed a high classification accuracy of 0.94. This reflects the powerful potential of convolutional neural networks in music classification.**

Keywords—music classification, convolutional Neural Networks, data, model, accuracy

I. INTRODUCTION

Music is an art form and cultural activity, and its medium is the regularly organized and regulated sound waves. Its basic elements include strength, tonality, duration, timbre, etc. The common "formal elements" of music are formed by the combination of these basic elements, such as rhythm, melody, harmony, as well as strength, speed, mode, form, texture, etc. The formal elements that make up music are the means of expression in music. Different types of music may emphasize or ignore certain elements. The creation, expression, meaning, and even definition of music vary depending on cultural and social backgrounds. For a long time, people have been trying to understand sound and the difference between one song and another. How to visualize sound. What makes one tone different from another.

The classification of music genres has become a current research hotspot, and the steps of music genre classification can be roughly divided into two parts: feature extraction and machine learning [1] Feature extraction plays a crucial role in the classification of music genres, and its effectiveness and efficiency greatly affect classification accuracy Traditional feature parameters include pitch, timbre, rhythm, spectrogram, Mel spectrogram, linear prediction coefficient, Mel scale frequency cepstral coefficients (MFCC), short-term features, etc The traditional music genre classification models include K-nearest neighbor (KNN) [2] model, support vector machine (SVM) [3] model, and Gaussian mixture model (GMM) [4]

A neural network is a mathematical or computational model that mimics the structure and function of biological neural networks (the central nervous system of animals, especially the brain). A neural network is composed of a large number of artificial neurons, which are connected in different ways to construct different networks. CNN is one of them, along with GAN (Generative Adversarial Network), RNN (Recurrent Neural Network), etc. Neural networks can have simple decision-making and judgment abilities similar to humans, and can provide better results in image and speech recognition. This article investigates the applicability of CNN for music classification. Specifically, spectral features are used as input representations and CNN is trained to classify music into predefined categories.

II. CONVOLUTIONAL NEURAL NETWORK

CNN， Convolutional Neural Network is a deep learning model that performs well in fields such as image recognition, video analysis, and natural language processing. CNN extracts local features of image data by using convolutional layers, then reduces the spatial dimension of features through pooling layers, and finally performs classification or regression tasks through fully connected layers. CNN is mainly used to solve problems involving large amounts of nonlinear and high-dimensional data, especially in the field of image processing. It can recognize and classify objects, faces, scenes, etc. in images. In addition, CNN has also been applied to various tasks such as speech recognition, natural language processing, medical image analysis, recommendation systems, etc. The structure of neural is showed in figure 1 below.

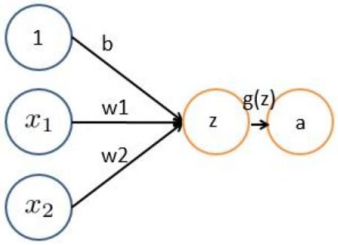

Figure 1 The neural structure of CNN

A. Neuron

A neural network is composed of a large number of interconnected neurons. After each neuron receives a linear combination of inputs, it initially only undergoes simple linear weighting, and later adds a nonlinear activation function to each neuron to perform nonlinear transformation and output. The connection between every two neurons represents a weighted value, which is called a weight. Different weights and activation functions can result in different outputs of the neural network.A neuron usually has the following important components:

1）Input: Neurons receive input signals from the previous layer. The input signal is usually obtained from the output of other neurons or raw input data. Each input signal will be assigned a weight value. The input can be represented as a one-dimensional vector $(x_1, x_2, ..., x_n)$, where each input signal xi represents a data point from the previous layer.

2）Weights: Each input signal has a corresponding weight, which determines the degree to which the input signal affects the final output. Weights are key parameters for neural network learning, and through the training process, the neural network continuously adjusts these weights to better fit the data. If the input signal is $(x_1, x_2, ..., x_n)$, then the weight corresponding to each input signal is（$w_1, w_2, ..., w_n$）.

3）Weighted summation: The neuron multiplies the input signal with the corresponding weights and sums these products. This process can be expressed as:

$$sum = w_1x_1 + w_2x_2 + ... + w_nx_n + b$$

Among them, b is a bias term that allows the neuron output to produce a non-zero value even without an input signal, which helps the model fit data more flexibly.

4）Activation Function: The weighted sum result is passed to an activation function. The function of activation is to introduce nonlinearity, enabling neural networks to handle more complex data patterns. If there is no activation function, the output of the neural network is only a linear combination and cannot represent complex decision boundaries.

5）Output: The output of the activation function is the final result of the neuron. This result will be passed on to the neurons in the next layer as input to the next layer.

Identify applicable funding agency here. If none, delete this text box.

B. Layer of CNN

Organizing a large number of neurons together forms a neural network.Convolutional neural networks typically consist of five layers: input layer, convolutional layer, activation layer, pooling layer, and fully connected layer.(figure 2)

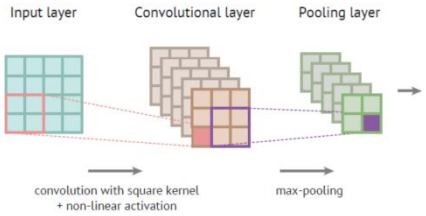

Figure 2 The Structure of Convolutional Neural Networks and the Training Process of Data in the Network

1）The input layer mainly preprocesses the raw image data.

2）Convolutional layers are the most important layers in convolutional neural networks. Convolutional kernels can convolve a certain region of feature maps into the input of the next layer of feature maps based on a certain algorithm. As the convolution kernel continues to move, the feature map of the input layer can be convolved into the input image of the next layer.

3）The activation layer nonlinearly maps the output result of the convolutional layer.

4）The pooling layer is sandwiched between consecutive convolutional layers to compress data volume and parameters, reducing overfitting.

5）The fully connected layer is typically located at the tail of a convolutional neural network and is used to connect all neurons between two layers of the network.

III. RELATED WORK

This experiment uses Python code to complete music classification work, which includes reading the dataset, data preprocessing (data augmentation, Mel spectral feature extraction), dataset partitioning, model construction, model training and evaluation. In this process, calling powerful library functions in Python is indispensable. The main libraries used are as follows:

A. Librosa

Librosa is a Python library specifically designed for audio and music signal analysis. It provides a range of functions, including audio feature extraction, audio visualization, rhythm analysis, audio processing, and more.

1）Audio feature extraction: Librosa can be used to extract audio features, such as Mel frequency cepstral coefficients (MFCC), spectral features, chromatic features, etc., for audio classification and analysis.

2）Audio visualization: The Librosa library can be used to draw audio waveforms, spectrograms, chromatograms, etc. to better understand the characteristics of audio data.Figure 3

shows the Log mel spectral features extracted from the audio using the librosa library.

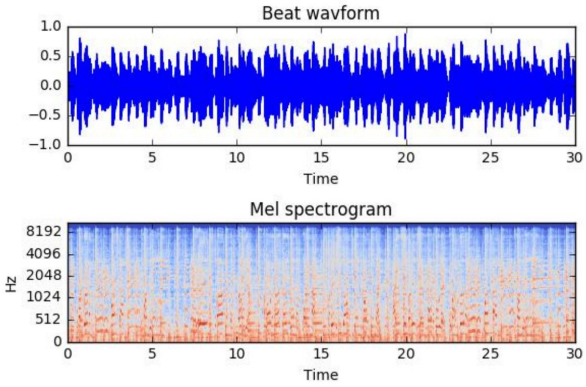

Figure 3 the Log mel spectral features extracted from the audio

3）Audio analysis: Librosa supports audio analysis tasks such as rhythm analysis, pitch estimation, and audio clustering.

4）Audio processing: The Librosa library can be used for audio processing, such as noise reduction, variable speed, pitch transformation, etc. This experiment uses Librosa to adjust the pitch of audio and change the sound speed of audio, thereby achieving data augmentation.

B. Tensorflow

TensorFlow is an open-source machine learning framework primarily used for building, training, and deploying various machine learning and deep learning models. TensorFlow provides flexible APIs (such as Keras) that enable developers to easily build various neural network architectures, including: CNN,RNN,GAN;Support automatic differentiation function, which can automatically calculate the gradient of model parameters,simplifying the implementation of backpropagation algorithm.This is crucial for optimizing model parameters to minimize the loss function;Using Data Flow Graph for numerical calculations enables efficient execution of complex computing tasks on both CPU and GPU; Support distributed training, able to train models in parallel on multiple computing nodes; Supports multiple platforms, including Windows Linux、macOS,And mobile platforms enable developers to deploy and run models in different environments.

C. Sklearn

Scikit learn (commonly referred to as sklearn) is one of the most popular machine learning libraries in the Python ecosystem. It is built on top of other scientific computing libraries such as NumPy, SciPy, and matplotlib, providing simple and efficient data analysis and machine learning tools. Its main functions include classification (SVM, decision tree, random forest, K-NN, Naive Bayes), Regression (linear regression, Ridge regression, Lasso regression, SVR), Clustering (K-Means, DBSCAN, Hierarchical clustering), Dimensionality Reduction (PCA, LDA, t-SNE). In addition, the sklearn library can also be used for feature extraction, data processing, and model evaluation. In the implementation of music classification, the train_test_stplit function of the library is used to partition the dataset into training set, validation set, and test set. At the same time, the library is also used to generate classification reports and confusion matrices after model training, in order to evaluate classification accuracy.

D. Matplotlib

Matplotlib is one of the most popular and powerful plotting libraries in Python, widely used for data visualization and graphic drawing. Matplotlib supports multiple types of 2D and 3D charts, including but not limited to: Line Plot, Scatter Plot, Bar Chart, etc; Matplotlib provides rich customization options, allowing users to finely adjust various aspects of the chart; It seamlessly integrates with numerous Python libraries, enhancing its functionality and application scope; Support exporting charts to multiple formats for easy sharing and publishing.

IV. MUSIC CLASSIFICATION

The dataset used for music classification consists of ten folders, each representing a category that stores all the audio in that category. The music categories include: blues、classical、country、disco、hiphop、jazz、metal、pop、reggae、rock。Each category folder contains 100 music files, which means there are a total of 1000 music data.

A. Data preprocessing

After importing the dataset, the first step is to perform data preprocessing. Data preprocessing consists of two steps: data augmentation and spectral feature extraction

1）Data augmentation: In deep learning, data augmentation is the process of changing input data in a certain way to generate more training samples, thereby improving the model's generalization ability and effectiveness. Data augmentation can reduce the excessive dependence of models on certain features, thereby avoiding overfitting and enhancing the accuracy of classification.

When training a model, it is required to have a sufficient number of samples. The more samples there are, the better the performance of the trained model and the stronger its generalization ability. However, in practice, if the sample size is insufficient or the sample quality is not good enough, it is necessary to perform data augmentation on the sample to improve its quality.

For example, in image classification tasks, simple operations such as translation, scaling, and color transformation can be performed on the input image. These operations do not change the category of the image, but can increase the number of training samples. These enhanced samples can help the model better learn and understand the features of the image, improving the model's generalization ability and accuracy.

In this work, it was found that the classification accuracy of the trained model was relatively low. To address this issue, the data has been enhanced. The specific method is to add noise to each audio data, with a noise factor of a random number between [0,0.2]. The original audio data and the audio data with added noise coexist, doubling the data volume.

Afterwards, increase the pitch of all audio by two units and increase the speed of sound to 1.1 times its original value.

2）Extract Mel spectral features: Mel spectrum is based on the Mel scale, which is a nonlinear frequency scale that simulates the sensitivity of the human ear to different frequencies.

The human ear is more sensitive to low frequencies, while its ability to perceive high frequencies gradually decreases. The Mel scale maps the original linear frequency (Hz) to a non-linear Mel scale, which is more in line with the auditory characteristics of the human ear. Extracting Mel spectra can help models capture features that are more relevant to human auditory perception, making the model more "intelligent". The original audio signal is a time-domain signal (waveform), and directly using waveforms may not effectively capture the frequency and energy distribution of the audio. Mel spectrum is a frequency domain feature calculated based on short-time Fourier transform (STFT), which displays the changes of audio in time and frequency, and is suitable for describing the frequency structure of audio. Mel spectrum preserves the main characteristic information of audio, such as frequency distribution and energy, while removing details unrelated to classification tasks. This makes the model input more concise and reduces the complexity of learning irrelevant information.

When extracting Mel spectral features, the final result is a two-dimensional matrix (time × frequency). However, due to the varying lengths of different audio, the length (number of columns) of the obtained matrix timeline will also differ. This brings up a problem: deep learning models require fixed size inputs. Therefore, in order to standardize the size of input features, we need to fill or truncate the Mel spectrum. If the audio length is insufficient (the number of columns in the Mel spectrum matrix is less than the target length), we fill in zero values at the tail (or head) to make up for it.If the audio is too long (the number of columns in the Mel spectrum matrix is greater than the target length), we only keep the front part and truncate the excess part.

3）The dataset is divided into training set, validation set, and testing set. The ratio is 8:1:1.

B. Building and Training Models

1）Building CNN Models:A deep learning model was constructed using Keras,which employs multiple convolutional layers (Conv2D) and pooling layers (MaxPooling2D) for feature extraction; Reduce overfitting through Batch Normalization and Dropout; Use fully connected layers (Dense) to complete classification tasks; Adapt the feature dimensions of audio input.The following is the construction and functional analysis of each layer:

The input layer defines the input shape of the model, including the frequency dimension, time step, and channel dimension of the Mel spectrum. The input shape is fixed and can match the Mel spectral features extracted earlier. Thus, batch normalization can be performed on the input data.

The model consists of two convolutional blocks. Block 1 has convolutional layers and max pooling layers, where the convolutional layers extract local features (such as frequency patterns of audio signals). The max pooling layer reduces the width and height of the feature map by half, reduces the size of the feature map (dimensionality reduction), and preserves important features. Thus improving computational efficiency and preventing overfitting. Block 2 has a second convolutional layer that uses 128 filters to extract higher-level features. Same as the first convolutional layer, use ReLU activation function and 3×3 convolution kernel. Pooling further reduces the size of the feature map, just like in block 1. Then normalize the convolutional output and use dropout to randomly discard 50% of the neurons to reduce overfitting.

The flattening and fully connected layers use the Flatten() function to flatten the multi-dimensional feature map output by convolution into a one-dimensional vector, preparing for the fully connected layer. The activation function of the fully connected layer is ReLU. Normalize the flattened features and randomly discard some neurons to reduce overfitting. And use L2 regularization to further limit the size of weights and prevent overfitting.

The final output layer needs to provide the number of categories as a parameter to convert the output into a probability distribution (normalized with each value between 0 and 1 and a total of 1). To classify the input samples in this way

The general structure of the CNN model is shown in the figure 4 and figure 5.

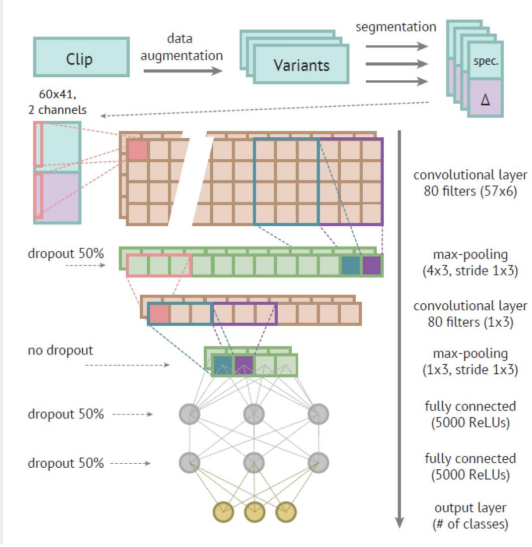

Figure 4 the structure diagram of the model

Model: "vggish"

Layer (type)	Output Shape	Param #
input_batch_norm (BatchNormalization)	(None, 60, 41, 1)	4
block1_conv1 (Conv2D)	(None, 60, 41, 64)	640
block1_pool (MaxPooling2D)	(None, 30, 20, 64)	0
block2_conv1 (Conv2D)	(None, 30, 20, 128)	73,856
block2_pool (MaxPooling2D)	(None, 15, 10, 128)	0
block2_batch_norm (BatchNormalization)	(None, 15, 10, 128)	512
block2_dropout (Dropout)	(None, 15, 10, 128)	0
flatten (Flatten)	(None, 19200)	0
embedding_3 (Dense)	(None, 128)	2,457,728
embedding_batch_norm_1 (BatchNormalization)	(None, 128)	512
embedding_dropout_1 (Dropout)	(None, 128)	0
classification_1 (Dense)	(None, 32)	4,128
prediction (Dense)	(None, 10)	330

Total params: 2,537,710 (9.68 MB)
Trainable params: 2,537,196 (9.68 MB)
Non-trainable params: 514 (2.01 KB)

Figure 5 The output of the code model. summary() is used to represent the structure of the model and the shape of the output data for each structure

2）Training Models: Before model training, callback functions were first set, including early stop and learning rate adjustment. The early stop function stops training when the validation set loss does not decrease for five consecutive epochs; The ReduceLOCnPlate function reduces the learning rate by half when the validation set loss does not decrease for three consecutive epochs.The function of early stopping is to prevent overfitting and save training time. The function of ReduceLONPlateau is to dynamically adjust the learning rate, prevent the learning rate from being too large or too small, and improve training efficiency.

Then, the model is trained with the hyperparameter epochs set to 50, indicating a maximum of 50 rounds of training; Batch_Size is set to 32, indicating that 32 samples are used for each weight update (small batch). Simultaneously call callback functions to dynamically adjust the training process. After training, return a history object to record metrics such as loss and accuracy for each epoch during the training process.

C. Evaluation

This classification uses classification reports and confusion matrices to evaluate the model.The classification report displays each category and the overall precision、recall、f1-score, and the number of cases that meet the category (depicted in Figure 6). The confusion matrix is a two-dimensional table with rows representing actual categories and columns representing predicted categories. The diagonal part of the confusion matrix in Figure 7 is darker in color, indicating that the predicted type roughly matches the actual type. The overall performance of the model is good. The music types in the figure correspond to blues,classical,country,disco,hiphop,jazz, metal,pop,reggae,rock, It can be seen that pop music has the lowest accuracy in classification, followed by hip-hop music.

	precision	recall	f1-score	support
0	1.00	0.89	0.94	46
1	0.88	1.00	0.94	45
2	0.88	0.96	0.92	46
3	0.95	0.98	0.97	43
4	1.00	0.97	0.98	33
5	0.90	0.84	0.87	44
6	0.97	1.00	0.99	34
7	0.88	0.93	0.90	30
8	0.95	0.88	0.91	42
9	0.97	0.92	0.94	37
accuracy			0.94	400
macro avg	0.94	0.94	0.94	400
weighted avg	0.94	0.94	0.93	400

Figure 6 The classification report of music classification

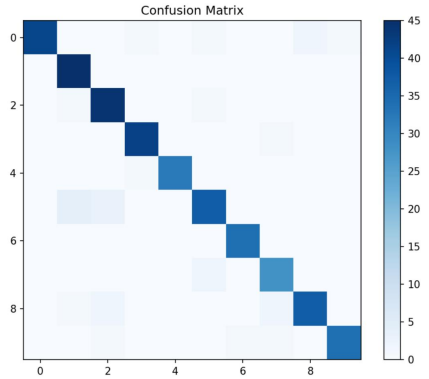

Figure 7 Confusion matrix between predicted type and actual type

V. SUMMARY

Convolutional neural networks have demonstrated strong potential in the field of music classification, significantly improving classification performance through automatic feature extraction and deep level model structures. However, in the face of the complexity and diversity of music data, further optimization of model architecture and training methods is still needed. With the continuous advancement of technology, CNN based music classification methods are expected to play a greater role in practical applications.

REFERENCES

[1] Wold E, Blum T, Keislar D, et al. Content-based classification, search, and retrieval of audio. IEEE Multimedia, 1996, 3(3): 27-36. DOI:10.1109/93.556537J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] Hearst MA, Dumais ST, Osuna E, et al. Support vector machines. IEEE Intelligent Systems and Their Applications, 1998, 13(4): 18-28. DOI:10.1109/5254.708428

[3] Kaur C, Kumar R. Study and analysis of feature based automatic music genre classification using Gaussian mixture model. Proceedings of the 2017 International Conference on Inventive Computing and Informatics (ICICI). Coimbatore: IEEE, 2017. 465–468.