Homework 5: Music Genre Classification

資訊111 F74071263 倪皓城

Prerequisites

Audio features

blues、classical、country、disco、hophop、jazz、metal、pop、reggas、rock

Environments

python 3.9.6

require packages

1. TensorFlow
2. Librosa
3. Numpy
4. Scikit-learn
5. JSON
6. Pytdub

Execution

> because I only submitted execution file(data preprocessing file is too large), so please execute the following instruction step by step

Data preprocessing

```
python preprocessing.py
```

Model training

```
python model.py
```

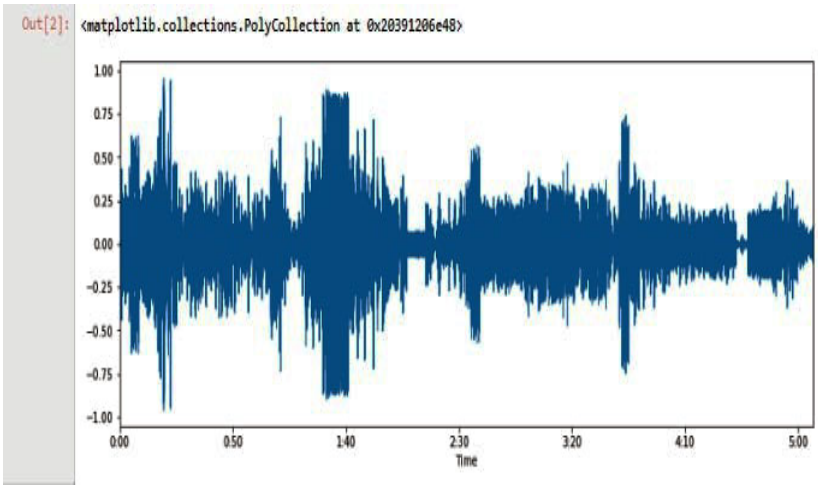
Prediction

> ex: python predict.py test_song.wav

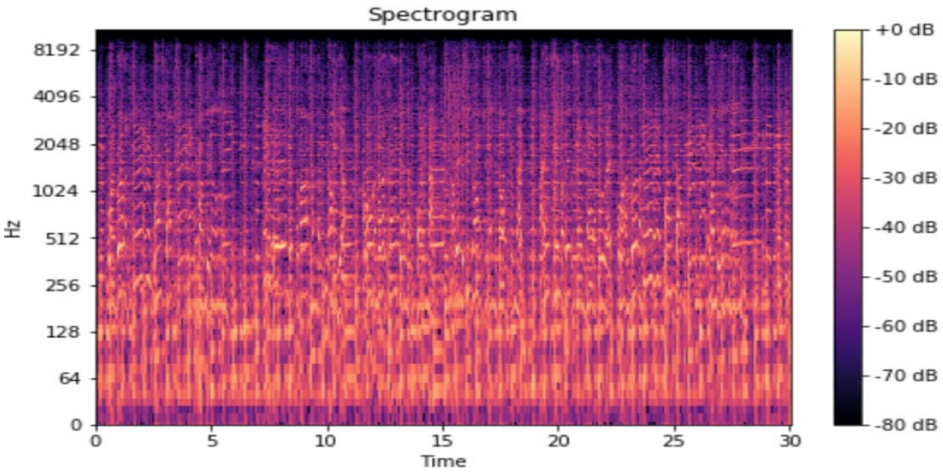
```
python predict.py "File Path"
```

Theory

For audio and music feature extraction in machine learning, we usually take mel-frequency cepstral coefficients(MFCCs) as extraction from song or audio.



But MFCC feature extraction is a way to extract only relevant information from an audio. In other words, the digital format not give us too much information about audio or song.Hences, we used Fast Fourier Transform(FFT) to represent the audio in **frequency domain**. The following figure displays the audio data in frequency and time doamins,called **Sepectogram**



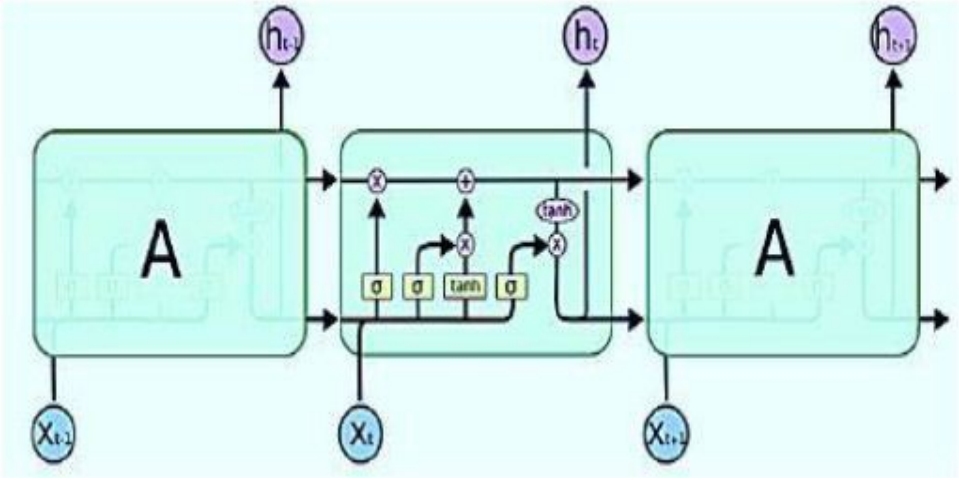
Data Preprocessing

The feature extraction will be done by using MFCCs. Librosa is python package used to extract the features from each of the audio segments. we label each segments to its correct genres and dump into json file for next model training step.

Model Training

Long short-term memory (LSTM) networks

There are several ways to solve this classification problem such as CNN networks. Here we used LSTM model which is a subdivision of RNN networks. In the past,we knows that RNN can used the past data as information to predict the new output. However, LSTM is modified networks overcoming the problem of **long term dependencies problems**.



Implements

LSTM is deep learning model for training **Time Series** data. We created 4 layers LSTM network using TensorFlow, including two hidden layers.

We initialize the model as sequential, adding one input layer with 64 as number of neurons in that layer,one hidden layer,one dense LSTM layer,and an output layer with 10 neurons for the **10 genres**. The size of input layer depends on the size of MFCC coefficient.

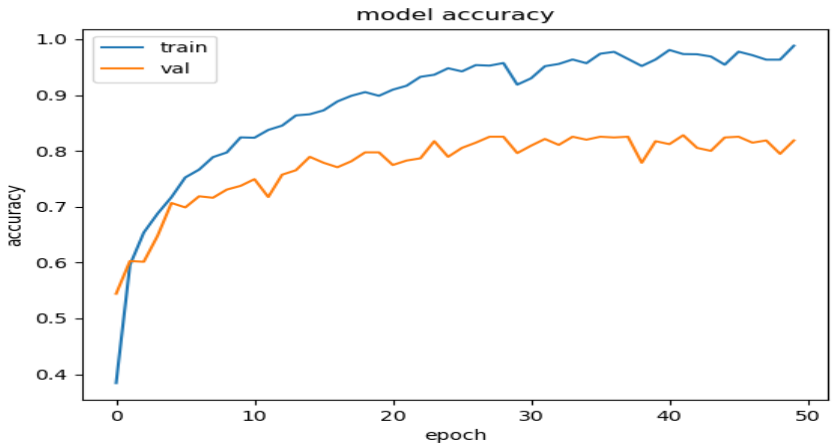
```
def build_model(input_shape):

    model = tf.keras.Sequential()

    model.add(tf.keras.layers.LSTM(
        64, input_shape=input_shape, return_sequences=True))
    model.add(tf.keras.layers.LSTM(64))

    model.add(tf.keras.layers.Dense(64, activation="relu"))
    # model.add(tf.keras.layers.Dropout(0.3))

    model.add(tf.keras.layers.Dense(10, activation="softmax"))

    return model
```

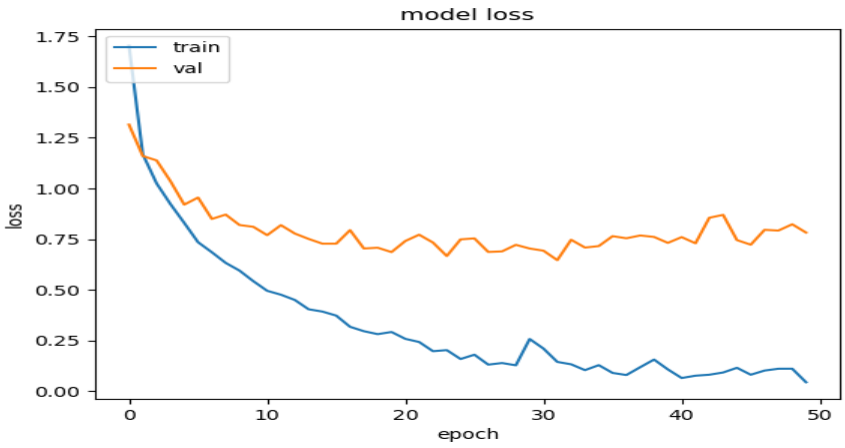
Accuracy/Loss

After servel time of model training, epochs=50 is most stable accuracy/loss in training preocess.

validation accuracy: 0.824800144958496 validation loss: 0.73860031786463

Accuracy





Loss