



딥러닝 기반 음악 생성

전산물리학

2023013472 채석현

CONTENTS

1

소개 및 배경이론



2

데이터 수집 및 전처리



3

모델 설계 및 학습



4

음악 생성



5

결과 및 개선방안



소개 및 배경 이론

이 프로젝트는 딥러닝을 이용해 비슷한 취향의 음악을 생성하는 시스템을 개발하는 것을 목표로 한다. 다양한 악기와 밴드 사운드를 포함한 음악을 자동으로 생성하는 것을 목표로 한다.

딥러닝은 인간의 학습 방식을 모방한 인공지능 기술로, 음악 생성 모델은 LSTM, GRU 등의 순환 신경망을 사용해 음악의 패턴을 학습하고 새로운 음악을 만든다.

데이터 수집 및 전처리

이 단계에서는 다양한 악기와 밴드 사운드를 포함한 음악 데이터를 수집한다.

그리고 나서 수집된 오디오 파일을 스펙트로그램으로 변환하여 딥러닝 모델에 입력한다. 여기서, 스펙트로그램은 오디오 신호의 주파수 성분을 시각적으로 표현한 것으로, 모델이 음악의 패턴을 학습하는 데 중요한 역할을 한다. 변환된 스펙트로그램은 고정된 크기로 자르거나 패딩하여 일관된 입력 형식을 유지한다.

데이터 수집 및 전처리

```
import os
import librosa
import numpy as np

# 오디오 파일을 스펙트로그램으로 변환
def audio_to_spectrogram(file_path, n_fft=2048, hop_length=512, n_mels=128):
    y, sr = librosa.load(file_path)
    S = librosa.feature.melspectrogram(y=y, sr=sr, n_fft=n_fft, hop_length=hop_length, n_mels=n_mels)
    S_db = librosa.power_to_db(S, ref=np.max)
    return S_db

# 스펙트로그램 크기 고정
def fix_spectrogram_shape(S_db, max_frames=2048):
    if S_db.shape[1] > max_frames:
        return S_db[:, :max_frames]
    else:
        padding = max_frames - S_db.shape[1]
        return np.pad(S_db, ((0, 0), (0, padding)), mode='constant')

# 스펙트로그램 저장
def save_spectrogram(S_db, save_path):
    np.save(save_path, S_db)

# 모든 오디오 파일에 대해 스펙트로그램 생성 및 저장
def process_all_audio_files(audio_dir, output_dir, max_frames=2048):
    for file_name in os.listdir(audio_dir):
        if file_name.endswith(".mp3") or file_name.endswith(".wav"):
            file_path = os.path.join(audio_dir, file_name)
            S_db = audio_to_spectrogram(file_path)
            S_db_fixed = fix_spectrogram_shape(S_db, max_frames=max_frames)
            save_path = os.path.join(output_dir, file_name.replace(".mp3", "").replace(".wav", "") + '.npy')
            save_spectrogram(S_db_fixed, save_path)

if __name__ == "__main__":
    audio_dir = "data/audio_files"
    output_dir = "data/spectrograms"
    os.makedirs(output_dir, exist_ok=True)
    process_all_audio_files(audio_dir, output_dir, max_frames=2048)
```

모델 설계 및 학습

모델은 Bidirectional LSTM을 사용하여 음악의 시간적 패턴을 학습한다. LSTM 레이어는 음악의 순차적 특성을 잘 반영하며, Dropout 레이어를 추가하여 과적합을 방지한다. 모델은 Adam 옵티마이저와 MSE 손실 함수를 사용하여 학습한다. 학습 데이터는 훈련과 검증 데이터로 나누어, 모델의 성능을 평가하며 최적의 하이퍼파라미터를 찾는다. 이 때, 학습은 여러 epoch에 걸쳐 진행된다.

모델 설계 및 학습

```
import os
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Conv1D, Dense, Dropout, Bidirectional, Input, Flatten, Reshape
from tensorflow.keras.optimizers import Adam

# 데이터 로드
def load_data(data_dir):
    data = []
    for file_name in os.listdir(data_dir):
        if file_name.endswith(".npy"):
            file_path = os.path.join(data_dir, file_name)
            data.append(np.load(file_path))
    data = np.array(data)
    return data

# 모델 정의
def build_model(input_shape):
    model = Sequential()
    model.add(Input(shape=input_shape))
    model.add(Bidirectional(LSTM(128, return_sequences=True)))
    model.add(Dropout(0.3))
    model.add(Bidirectional(LSTM(128, return_sequences=True)))
    model.add(Dropout(0.3))
    model.add(Conv1D(64, kernel_size=3, padding='same', activation='relu'))
    model.add(Dropout(0.3))
    model.add(Conv1D(32, kernel_size=3, padding='same', activation='relu'))
    model.add(Dropout(0.3))
    model.add(Flatten())
    model.add(Dense(input_shape[0] * input_shape[1], activation='sigmoid'))
    model.add(Reshape(input_shape))
    return model

if __name__ == "__main__":
    data_dir = "data/spectrograms"
    data = load_data(data_dir)

    # 데이터 형태 확인
    if len(data) == 0:
        raise ValueError("No data found. Ensure that the spectrogram files are correctly saved in the data/spectrograms directory.")
    if len(data.shape) != 3:
        raise ValueError(f"Unexpected data shape: {data.shape}. Expected a 3D array.")

    # 데이터 형태 출력
    print(f"Data shape: {data.shape}")

    input_shape = (data.shape[1], data.shape[2])
    model = build_model(input_shape)
    model.compile(optimizer=Adam(learning_rate=0.001), loss='mse')

    model.fit(data, data, epochs=100, batch_size=4, validation_split=0.2)
    model.save("models/music_generator_model.h5")
```

음악 생성

음악 생성을 위해 학습된 모델을 사용한다. 초기 시드를 설정하고, 모델이 이를 기반으로 새로운 음악 시퀀스를 생성한다. 생성된 시퀀스는 스펙트로그램 형태로 출력되며, 이를 오디오 신호로 변환하여 WAV 파일로 저장한다. 생성된 음악의 길이는 설정된 시퀀스 길이에 따라 조정되며, 다양한 악기와 밴드 사운드를 포함한 음악을 생성한다.

음악 생성

```
import numpy as np
import tensorflow as tf
import librosa
import soundfile as sf

# 모델 로드
def load_model_with_custom_objects(model_path):
    return tf.keras.models.load_model(model_path, custom_objects={"mse": tf.keras.losses.MeanSquaredError()})

# 새로운 음악 생성
def generate_music(model, seed, length=2048):
    generated = seed
    for _ in range(length):
        # 입력 데이터의 형태를 올바르게 맞춤
        prediction = model.predict(generated[:, -1, :].reshape(1, -1, generated.shape[2]))
        generated = np.concatenate((generated, prediction), axis=1)
    return generated

# 스펙트로그램을 오디오로 변환
def spectrogram_to_audio(S_db, hop_length=512, gain=1.0):
    S = librosa.db_to_amplitude(S_db)
    y = librosa.istft(S, hop_length=hop_length)
    y = y * gain # 볼륨 조정
    return y

if __name__ == "__main__":
    model_path = "models/music_generator_model.h5"
    model = load_model_with_custom_objects(model_path)

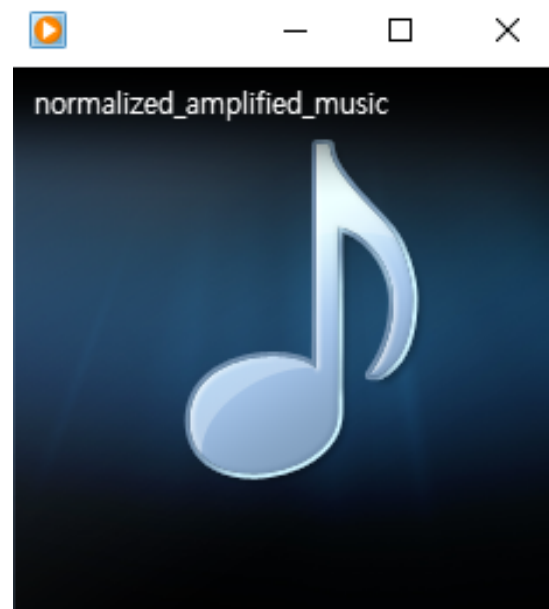
    seed = np.random.rand(1, 1, 2048) # 초기 시드의 크기를 모델 입력 크기에 맞춤
    generated_music = generate_music(model, seed, length=2048)

    # 마지막 차원을 제거하여 2D 스펙트로그램으로 변환
    generated_music = generated_music[0, :, :]

    # 스펙트로그램을 오디오로 변환
    gain = 5.0 # 볼륨을 조정
    generated_audio = spectrogram_to_audio(generated_music, gain=gain)

    # WAV 파일로 저장
    wav_path = "generated_music.wav"
    sf.write(wav_path, generated_audio, 22050)
```

결과 및 평가, 그리고 개선방안



음악 생성 모델의 성능은 손실 값과 생성된 음악의 품질로 평가되는데, 현재 모델은 음악의 구조와 패턴을 잘 학습하지만, 일부 한계가 있었다. 생성된 음악의 품질을 높이기 위해 더 많은 데이터를 수집하고, 모델 아키텍처를 개선할 필요가 있다.

향후 연구 방향으로서는 데이터 증대, 모델의 복잡도 증가, 실시간 음악 생성 기능 추가 등이 있다. 이러한 개선을 통해 더 자연스럽고 풍부한 음악을 생성할 수 있을 것으로 기대된다.

사용된 데이터들..

1. gun's N roses - sweet child o mine
2. radiohead - creep
3. oasis - don't look back in anger
4. queen - bohemian rhapsody
5. coldplay - viva la vida
6. coldplay - a sky full of stars



감사합니다

2023013472
채석현