CNN and K-NN for Music Recommendation Model

A Research Paper submitted to the Department of Engineering and Society

Presented to the Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science, School of Engineering

Tho Vu
Spring 2024

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Advisor
Rosanne Vrugtman, Department of Computer Science
Briana Morrison, Department of Computer Science

CNN and K-NN for Music Recommendation Model

CS4991 Capstone Report, 2024

Tho Vu
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
thv7pas@virginia.edu

ABSTRACT
Music is an important factor in many peoples' lives, and with the wide array of music genres, finding music based on personal preferences can be time consuming and tedious. To address this problem, I worked with three other students to develop a machine learning model that can take in one song the person likes and output ten songs with similar features. The model utilizes a Convolutional Neural Network (CNN) for feature extraction, TensorFlow, and a database of songs. Our model cannot be gauged with accuracy values but based on the distance from the recommended song to the original on a k-nearest neighbor (K-NN) graph. Future work for this project could include having a more extensive database of songs, having the capacity to recommend more than ten songs at once.

1. INTRODUCTION

Without music, the world would be a silent place, without rhythms and melodies. Research has revealed a strong connection between music and emotions; different types of music can impact our emotions differently. Jazz music soothes people, pop music boosts their energy, rhythmic drumming aids meditation, and classical music enhances memory recall. As such, different genres of music can produce various types of emotions and make a positive impact on individuals' mental health. For those with specific musical preferences and purposes, a digital music classifier can be invaluable, since it enables listeners to continuously enjoy their preferred songs (BW Online Bureau, 2019). This is crucial because music can help reduce anxiety, decrease blood pressure and pain, and improve sleep quality, mood, and memory. This project can be useful for people who crave playlists that fit their interests and it can also help keep people engaged with music to improve mood and health (Johns Hopkins Medicine, 2022).

2. RELATED WORKS

Most major music streaming apps use either a content-based or collaborative recommendation system. Content-based recommendation systems use patterns in the songs in combination with user preferences. Collaborative-based filtering uses similarities between user preferences to recommend music. Research done by Ning & Li (2020) found that content-based filtering cannot recommend potential new interests that might develop over time. By combining user collected data such as browsing history or purchase history, the recommendation algorithm can be more dynamic (Ning & Li, 2020).

A research project by Schedl, et. al (2018), discusses some of the challenges of music recommender systems (MRSs). Some of these challenges include the cold-start problem, which is the lack of data for a new user or item. Another problem is grouping songs with

similar characteristics with each other (Schedl, et. al, 2018). This highlights how the field of music recommendation has some challenges to overcome and that it continues to improve.

3. PROJECT DESIGN

Our project has two main components: the first component consists of cleaning and analyzing our data set to create a model to classify the data set, using a convolutional neural network The second component makes use of the features extracted from the CNN of the first component. Then, it takes a music soundtrack input and uses the K-NN algorithm based on the features of the inputted song and returns a list of ten songs that have the most similar features. Using this two-prong method will allow the model to generate a playlist that can be read from unlabeled data, then classify them based on certain characteristics and be used in a K-NN algorithm.

3.1 Creating the CNN

First, we converted our data set of WAV files into spectrogram images. Based on the spectrogram images, we used convolutional neural networks that extracted unlabeled features from the images. We chose to use CNNs, since we were trying to extract the features from a spectrogram image. In addition, a CNN will allow the model to analyze the image with unlabeled features. However, a major obstacle we ran into was the techniques in which we augmented the data to avoid over-fitting. Initially, we ran the model with no data augmentation, since traditional data augmentation techniques such as random rotations, zooms, and contrast modifications could alter the data in a way that made it hard for the CNN to effectively learn the features that each genre possessed. However, that proved to be ineffective as the model was quickly overfitting within the first 10 epochs. As such, we carefully picked out data augmentation techniques that add variety to the relatively small dataset without losing the features that made each genre unique. These techniques included random noise, horizontal shifts, and brightness modifications. That way, the model was able to perform better on unseen data, and avoided overfitting as much as the initial model did.

3.2 Training the CNN

The dataset we used for training only had 1000 songs. After doing an 80-20 split for the training and testing datasets, that left us with 800 songs to train the CNN. The initial plan of building a custom CNN model was put aside due to small amounts of training data. Thus, we utilized transfer learning in our project since it saved us a lot of training time, and provided better performance than training with the ground-up. Our project utilized the Keras family of models, and each of them was tested on accuracy.

The models performed best when their pre-trained layers were frozen. Our hypothesis is that training with unfrozen layers destroys their feature-learning capabilities and means that the model was essentially guessing about the genre. This hypothesis is backed by the fact that the accuracy of the model no matter the epoch hovered around 10%, which is the probability of guessing the genre correctly when given ten genres. Nevertheless, we proceeded with frozen models and added a few custom layers on top to adapt to our dataset. We added a global average pooling layer, making the model more robust to spatial variations in the input. This was particularly important as the training dataset contained a 30-second snippet.

A dense layer was also added, since its high number of neurons allows it to learn complex patterns in the data. And finally, a dropout layer was added to alleviate overfitting. The models we tested included Xception,

ResNet50V2, and VGG19. Once we finished fine-tuning the model, we reran with the output layer removed, which returned the penultimate dense layer that included the features that were extracted. Once that was done, a n-dimension plane was created to be used for the next step.

3.3 K-Nearest Neighbors

Once our CNN model was ready, we utilized another dataset of unlabeled songs to make use of our feature extractor. Each song was put through the same pipeline as the training dataset was, and their features were extracted. With the n-dimension plane, the model will be able to receive an input of a wav file. Using the same method as component 1, the model will convert the wav file into a spectrogram, extract the features, and place the soundtrack onto the n-feature dimension plane accordingly. K-NNs are normally used in classification tasks. In our case, it was used to find k songs that were the closest to the input song. With the target data point on the plane, the model will be able to make use of the 10-nearest neighbors algorithm to determine which ten WAV files have the closest characteristics and generate a playlist

4. RESULTS

For each model, we measured the loss, accuracy, precision, and recall values. Of the three models we trained and tested, ResNet50V2 came out to be the most accurate model with about 70% accuracy, followed by VGG19 with 62% accuracy, and lastly Xception with 59% accuracy. Based on the results, we chose ResNet50V2 as the model that would create the song recommendation playlist. To gather actual results from our model, we gathered the top 100 most popular songs in the United States as our dataset and inputted *Levitating* by Dua Lipa. Although it is not possible to accurately judge how close a song is to another due to the subjectivity of the matter, we can still measure the distance from *Levitating* to the other songs. Some of the results included *Baby Don't Hurt Me* by Davide Guetta, *Dance the Night Away* by Dua Lipa, and *Houdini* by Dua Lipa, with a distance of 13.49, 13.53, and 15.23, respectively.

5. CONCLUSION

The model created has achieved affirmative results that can classify music fairly well. With this model, users will be able to upload soundtracks of music they enjoy, and then receive a playlist of ten songs that have similar characteristics. This model has successfully achieved a high accuracy rate, with the use of region-based CNN and k-NN techniques. The model will allow people to have easier access to songs of their liking without having to listen to the same song on repeat. Completion of the model also allows for the promotion of health benefits, because users can select songs of a specific genre that they like and use the model to find similar songs in the same genre.

6. FUTURE WORK

The project was conducted with limited computing resources. These included not having a GPU that could quickly test to process training and evaluation of our model. Future work will need to include enough computing resources and increase the number of songs, since we have relatively smaller datasets of 1,000 songs. Adding more songs would allow the model to produce a more advanced model and improve performance to satisfy users' tastes.

We now have 10,000 features without labels to create playlists. Analyzing these features and grouping them by genres or some explicit characteristics that people can recognize easily such as upbeats, energy, and rhythms would make our model more useful. Our model has been developed and tested locally, but considering the benefits of the model,

publishing it through web platforms such as Heroku, Google Cloud Platform, or GitHub Pages would allow many people to use it and find preferred songs that match their tastes.

Additionally, instead of using a pre-trained network for image classification from Tensorflow (a framework used for various tasks such as image recognition, reinforcement learning, voice and speech recognition, and Natural Language Processing), we can consider developing our own network to inspire other research and projects.

7. ACKNOWLEDGMENTS

This project was developed in collaboration with Mohsen Alghannam, Sofia Yang, and Claire Yoon.

REFERENCES

BW Online Bureau. (2019, March 27). The effects of music genres on the human brain. https://bwhealthcareworld.businessworld.in/article/The-Effects-Of-Music-Genres-On-The-Human-Brain/27-03-2019-168501/

Johns Hopkins Medicine. (2022, April 13). Keep your brain young with music. https://www.hopkinsmedicine.org/health/wellness-and-prevention/keep-your-brain-young-with-music

Ning, H., & Li, Q. (2020, December 18). Personalized Music recommendation simulation based on improved collaborative filtering algorithm. Complexity. https://www.hindawi.com/journals/complexity/2020/6643888/

Schedl, M., Zamani, H., Chen, C.-W., Deldjoo, Y., & Elahi, M. (2018). Current challenges and visions in Music Recommender Systems Research. International Journal of Multimedia Information Retrieval, 7(2), 95–116. https://doi.org/10.1007/s13735-018-0154-2