Comparing CNNs and RNNs for Music Genre Recognition

Anonymous Author(s)
Affiliation
Address
email

Abstract

Previous work has demonstrated the strength of Convolution Recurrent Neural Networks in musical classification problems with their ability to draw patterns in frequency and how they change over time by combining the demonstrated image recognition powers of a CNN with an RNN's power to recognize change over time by giving a network long term memory. Combining these two networks leads to substantially improved performance, however at this is at the cost of increased training times. Splitting these back into two separate networks we can compare metrics between the two and see that the Convolution layers in the complete CRNN is doing most of the heavy lifting, getting passable metrics with slightly reduced model size and training times. A CRNN is still by far more successful however in situations where a lightweight solution is required these results show that a CNN is still a viable option.

1 Introduction

Musical classification is a task that weighs heavily on the current music marketplace. Streaming platforms often offer users suggested tracks or artists or auto fill playlists with related songs. Certain solutions to these problems involve estimating song qualities by user feedback, ie. users that listen to this X artist often like Y artists so we'll suggest artist Y to users that listen to artist X. Another solution is to classify songs manually by certain characteristics such as their genre or tempo, and create recommendations through songs that share similar characteristics. But with the plethora of new music being created everyday it could become a daunting task to manually attach such tags to new works of music. A solution being widely studied at the moment is the use of deep learning to make predictions about music to classify it and produce recommendations.

1.1 Related Works

A paper published in 2019 (1) demonstrated the power of a convolution recurrent neural network, or CRNN, in artist classification. The network they developed showed strong results in distinguishing between artists from songs it had never heard before, demonstrating its ability to recognize key features about an artists style and musical vocabulary especially as all the artists in the data set used for this study made music of the same genre and thus were similar enough that it isn't uncommon for humans to mix them up. The idea behind using a CRNN is to combine the strengths of convolutional networks, which have shown strength in image classification problems, and recurrent networks which excel at finding patterns over the time dimension. The convolution layers are able to pull out key and timbrel information from a spectrogram and the recurrent layers are able to look at how said frequency information changes over time.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

1.2 The Goal

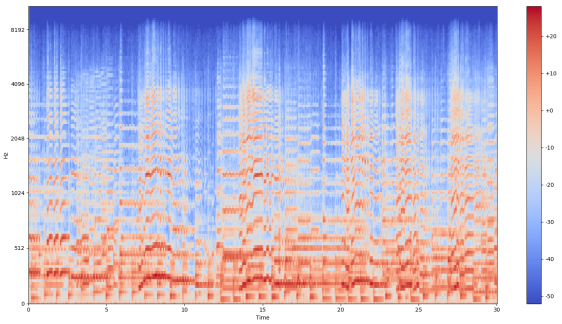
Initially when starting work for this project, the goal was simply to replicate the methodology of the work outlined above and use a CRNN for genre classification, however upon construction of the model I immediately started to run into problems related to disk space. I didn't have enough storage left on my disk to run a model containing both the convolution and recurrent layers at the same time. The obvious fix would simply be to clear up some disk space however this small setback had me wonder whether the whole model was really necessary for the task or if either the CNN or RNN models alone would suffice for this problem. The new task was to train both networks on the same data and compare the results between the two and determine which layers were playing the bigger role in producing the results seen in the previous work or whether they were all entirely necessary. Saving space may not be the biggest deal in the modern age when we can store dramatically more data than ever before, however if space can be saved and similar results can be achieved that allows for solutions which can scale better and be employed in stricter scenarios which is always welcome.

2 Methodology

2.1 Data and Data Representation

The data for this project came from the GATZAN music genre classification dataset on kaggle, containing 10 distinct genres and 100 30 second wav files per genre. In order to feed this data into the networks we need a way to represent it that is capable of expressing the things we want to extract from it. Musical genres are classified by the common stylistic choices from with songs within the genre, such as instrument choice, melodic structure and chord types, rhythmic structure, and tempo among many other things. These characteristics are found within the time and frequency dimensions thus our representation should contain information on both of those axis, which leads us to the spectrogram. A spectrogram can be obtained by taking the Short Time Fourier Transform, or STFT, of an audio signal resulting in a matrix which we can plot to visualize the strength of different frequencies within the signal over a specified time period. This representation is good for our CNN as the convolutional layers purpose is to find patterns within two dimensional data, however it can be improved. As our perception of sound is logarithmic, we have more trouble distinguishing small changes in frequency and higher pitches. The structure of music theory acknowledges this fact and bases notes off of perceived pitch change rather than some linear pattern of increasing frequency, thus the difference in frequency between a low C note and a low D note would be small compared to the same two notes a few octaves higher, however the perceived change would be about equal. Thus in order to normalize the data and hopefully see better performance from our network we transform the frequency dimension of our spectrograms from hertz to into the mel scale which produces a more linear output. This practice has been shown in previous work to improve performance in classification tasks.

Figure 1: Melspectrogram of 45th jazz training file

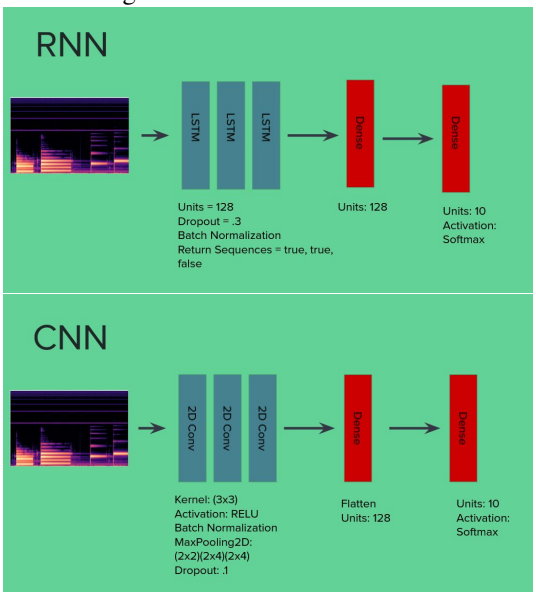


2.2 Model Architecture

For the CNN architecture I adapted the CRNN found in the previous work removing the recurrent layers but maintaining the same structure as their convolution component. The network consists of

convolution layers with 3x3 kernals, Elu activation, Batch normalization, max pooling and dropout. This output is then flattened and sent through a fully connected layer with softmax activation to give a probability distribution of the genres as the models output. The RNN looks similar but uses 3 LSTM layers with dropout and batch normalization instead of the convolution and pooling layers.

Figure 2/3: Model Architecture



The input data for both networks was processed the same way. The code parses one genre at a time storing each files melspectrogram and its expected output into a tuple into a list which I used pickle to save to disk to avoid having to reparse the dataset for every test. This process was repeated for 1, 5, 15, and 30 second clips. The last 10 songs of each genre were saved to a separate list to test the model with after training. Both models were compiled with categorical cross entropy and adam optimizer, then trained on each duration for 30 epochs. After training, the final 100 clips were fed back into the model to get predictions to measure the efficacy of our model. The category with the greatest probability as the output of the network was selected as the prediction and then dividing the number of correct predictions by the total, 100, predictions made gives us our models accuracy.

2.3 Results

Assuming the model finds no pattern and guesses at random we would expect to see an accuracy of about 10% as there are 10 potential categories, so I was a little worried when my first few trys training the RNN resulted in values around 11 to 12% accuracy on the training data. After tweaking some parameters I was able to get slightly better results:

Table 1: RNN Accuracy

1 second	5 seconds	15 seconds	30 seconds
14%	21%	15%	14%

While I don't doubt that further tweaking could show further improvement. These results suggest that an RNN, while somewhat capable of recognizing pattern within genre, is clearly not the greatest choice for this task. One thing to note was that the over multiple training sessions the 5 second training data provided the best results with accuracy decreasing as time increased or decreased suggesting that the RNN has trouble handling a large amount of time steps.

On the other hand, I did not do much tweaking to the CNN model as it was producing much better results out of the gate. Unlike the RNN, accuracy increased dramatically with the length of the clips the model was trained on. The CNN's ability to scale well with more data suggests that it may be possible to get even greater results out of higher resolution spectrograms, thus in the future I'd like to see the results of testing the same model with more mel bins, shorter time steps, and possibly increased sampling rate. The current results were as follows:

Table 2: CNN Accuracy

1 second	5 seconds	15 seconds	30 seconds
17%	21%	35%	42%

While these predictions could likely improve by tweaking some conditions to the model such as reducing the learning rate and training the model for more epochs, they are significantly better than what was achieved by the RNN and took the same amount of time to train proving the CNN's dominance in classification problems. One thing to note about these results was that they were very inconsistent and retraining the model often resulted in large fluctuations in accuracy. This is likely due to the model trying to classify between 10 different categories with not enough data and not enough time spent training. Training the model for 50 to 100 epochs resulted in much more stable results but took to long to do for all different time windows.

3 Conclusion

While neither the CNN or RNN get results as accurate as the CRNN a CNN still gets viable results in music classification tasks, providing a lighter weight alternative when a desired model is needed to run under stricter conditions such as a small amount of disk space or time for training. On the other hand the RNN was unsuccessful in providing usable results. The RNNs purpose in the CRNN is really to aid the pattern recognition and abstraction done by the Convolution layers before it and thus is not great for this task on its own.

link to source code on github: https://github.com/jArnke/JakeYenneyCSE190FinalProject

References

[1] Nasrullah, Z. & Zhao, Y. (2019) Music Artist Classification with Convolutional Recurrent Neural Networks University of Toronto.