Neural Network Music Genre Classification

Nikki Pelchat
Software Systems Engineering
University of Regina
Regina, Canada
pelchat.nikki@gmail.com

Craig M Gelowitz
Software Systems Engineering
University of Regina
Regina, Canada
craig.gelowitz@uregina.ca

***Abstract*—Music genre classification utilizing neural networks has achieved some limited success in recent years. Differences in song libraries, machine learning techniques, input formats, and types of neural networks implemented have all had varying levels of success. This paper reviews some of the machine learning techniques utilized in this area. It also presents some initial research work on music genre classification. The research uses images of spectrograms generated from time-slices of songs as the input into a neural network to classify the songs into their respective musical genres.**

***Keywords*—neural network, music, spectrogram, classification, music genre, convolutional neural network, deep learning**

I. Introduction

A. Machine Learning and Neural Networks

Machine learning has become very popular in recent years. Depending on the type of application and the dataset available, certain types of machine learning techniques are more appropriate than others for different applications. There are generally four main types of learning algorithms in machine learning. The main types of learning algorithms include supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning [1]. Supervised learning utilizes a fully labelled dataset to build a mathematical model whereas unsupervised learning attempts to extract useful features from an unlabelled dataset without any specific target in mind. Correspondingly, semi-supervised learning utilizes a dataset that contains both labelled and unlabelled data. Reinforcement learning takes a different approach through the use of a feedback mechanism. In reinforcement learning, the learning is accomplished through a process of being rewarded for correct actions or predictions. For example, reinforcement learning is often used in games where the intent is to minimize risk and maximize reward.

A neural network (NN) is a technique of machine learning that is generally effective at extracting critical features from complex datasets and deriving a function or model that expresses those features [2]. The NN utilizes a training dataset to first train a model. After the model is trained, the NN can then be applied to new or previously unseen data-points and classify the data based on the previously trained model.

A convolutional neural network (CNN) is a type of neutral network that is intended to process multi-dimensional arrays such as images [2]. A CNN can be used for both binary classification and multi-classification tasks where these classifiers differ only in the number of output classes. For example, a dataset of animal images can be utilized to train an image classifier. The CNN is provided a vector of pixel values from an image that is accompanied with the correct output class label the vector describes (cat, dog, bird, etc.). When an image is provided as an input to the CNN for training, it will attempt to classify the image with an output class. For every training image, the classification is compared with the expected output class label. The weights throughout the network are then iteratively updated through a backpropagation process in an attempt to improve the accuracy of the CNN during training. Each iteration helps form the CNN model which defines the training dataset's features and its ability to correctly classify images.

There are several other types of NNs that have been developed for different problem sets and application types [3]. For example, generative adversarial networks (GANs) have been used to generate brand-new image content that has similar characteristics to a given training dataset [4]. An example of this might be producing a realistic new image of a cat or even a human face.

Auto-encoders can be used to reduce the dimensionality of data into a smaller latent space while preserving imperative features. Training an auto-encoder includes shrinking input down to a compressed representation and then decompressing the data with the aim to recover the original data [5]. Once trained, auto-encoders have been used to predict missing or corrupted values of images [3].

Another common NN is the recurrent neural network (RNN). RNNs are used when sequential context is important. This means the output from a previous iteration may influence the current output [6]. Common application examples of RNNs are for text data and speech data because the order of words is important with respect to the meaning of the sentence.

B. Music classification

Humans are particularly good at listening to short samples of songs with the ability to distinguish the artist, the song title and even the genre. Emulating these abilities has been attempted through a number of NN approaches and they have shown varying levels of success [7]. A popular application example for

978-1-7281-0319-8/19/$31.00 ©2019 IEEE

automatically determining both artist and song title from music data is the mobile device application Shazam. Shazam is primarily known for being able to determine the song title and the artist from only a few seconds of a song. Shazam has also been working on being able to classify different aspects of music data including genre, instruments used, mood description of the composition, and whether a given user might like a given song [8]. Shazam uses a technique that they refer to as a song's signature. Shazam defines a song's signature as the large peaks in amplitude taken from the song's spectrogram. "It's like collecting the locations of the highest mountain peaks in a region; instead of (*latitude, longitude, altitude*), we have (*time, frequency, amplitude*) for these prominent peaks" [8]. Tim O'Brien from Shazam put together a neural network which consisted of two fully-connected layers with a final classification output layer of genre labels. "This resembles a very 'vanilla' multi-class classifier model". He achieved test accuracy in the low 90% range. Combining his NN with Sharath Pingula's (a Shazam employee) track-level collaborative filtering features, he was able to slightly improve the model.

Others have had success in music genre recognition using spectrograms. Yandre Costa, Luis Oliveira et al [9] worked on music genre classification using different types of inputs including spectrograms. Using 900 songs from the Latin Music Database, they divided them into 10 music genres equally. From each of the 900 songs, they extracted three 30-second segments from the beginning, middle, and end of the songs. From the spectrograms they mathematically extracted several features where each 30 second spectrogram was represented as ten 28-dimensional feature vectors. This ultimately resulted in 30 vectors for a single music piece. With those vectors as inputs, they trained an NN and then compared their work with Lopes et al [10]. Lopes had previously presented a method to limit the number of training datapoints to only include the ones that had shown better discrimination with respect to the output class. The final accuracy Lopes achieved was 60% whereas Yandre Costa, Luis Oliveira et al. saw an improvement of 7% having achieved a 67% recognition rate.

Despois [11] also used spectrograms as input into a neural network. Despois used a music library which consisted of 2000 labelled songs but reduced the number of genres to Hardcore, Dubstep, Electro, Classical, Soundtrack, and Rap. The sub-genres were included in the main genre class. He then converted the songs into a spectrogram and sliced the spectrogram up into 128x128 pixel images which equated to 2.56 seconds of the song. He used these spectrogram snippets as input into his CNN. The architecture of the CNN consisted of four convolutional neural network layers, a fully connected layer, and a softmax function to classify the results into the genre classes specified. The test set accuracy was not provided but the reported validation set accuracy was 90%. He also mentioned some future research and tweaks to the code that could be done which might vary the accuracy of the NN. It included changes to the function that divides the dataset into training, validation and test sections, which could potentially bring down the accuracy.

II. Implementation Work and Initial Results

A neural network similar to Despois [11] has been implemented in this work with some alterations. The differences include an increase in the number of music genre classes as well as the use of a different music dataset. It also included a change to the activation function as well as an increase in the number of training spectrogram slices per genre used for training the neural network.

In this work, a music library of 1880 songs categorized into seven genres was used. The preparation of the music dataset included transforming the stereo channels into one mono channel and utilizing the SoX (Sound eXchange) [12] command-line music application utility to convert the music data into a spectrogram. An example of a spectrogram of Pop music is shown at the top of Fig. 1.

The next step in preparing the dataset included slicing up the larger spectrograms into 128 pixel wide PNGs, which represented 2.56 seconds of a given song. An example of the spectrogram sliced into 128x128 pixel images are shown in the bottom of Fig. 1.

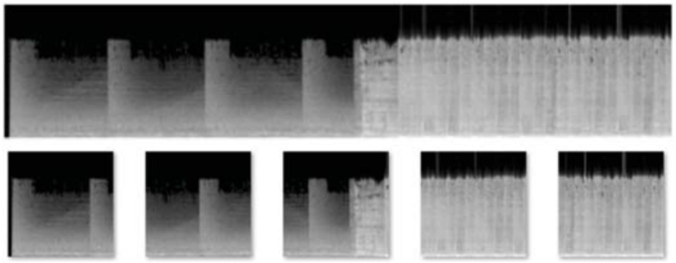

Fig. 1. (Top) 13 second spectrogram; (bottom) divided into 2.56s segments

The dataset consists of 1880 different songs that are each three minutes in length. They are divided into 2.56 second segment spectrograms to make approximately 132,000 labelled spectrogram snippets. The labelled spectrogram inputs of the dataset were split into 70% training data, 20% validation data, and 10% test data. This transformation is illustrated in Fig. 2.

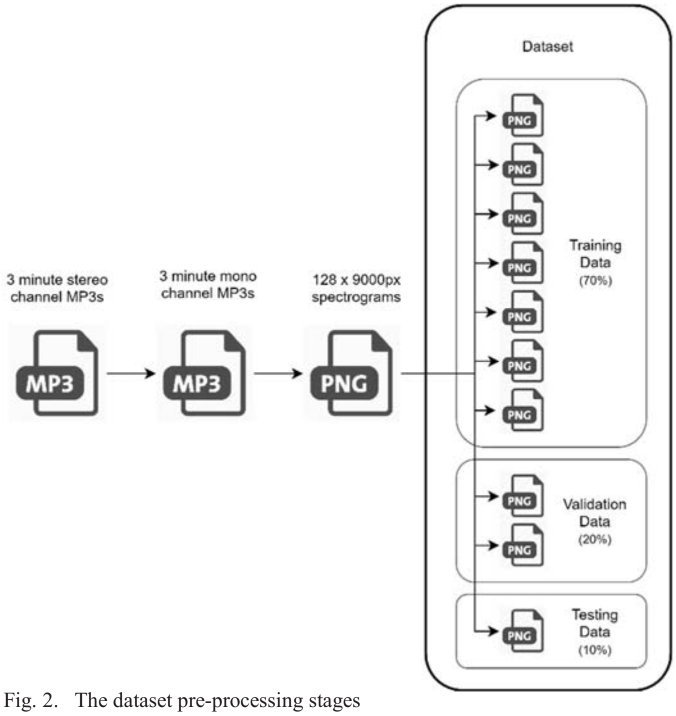
Dataset
3 minute stereo channel MP3s
3 minute mono channel MP3s
128 x 9000px spectrograms
MP3
MP3
PNG
Training Data (70%)
Validation Data (20%)
Testing Data (10%)

Fig. 2. The dataset pre-processing stages

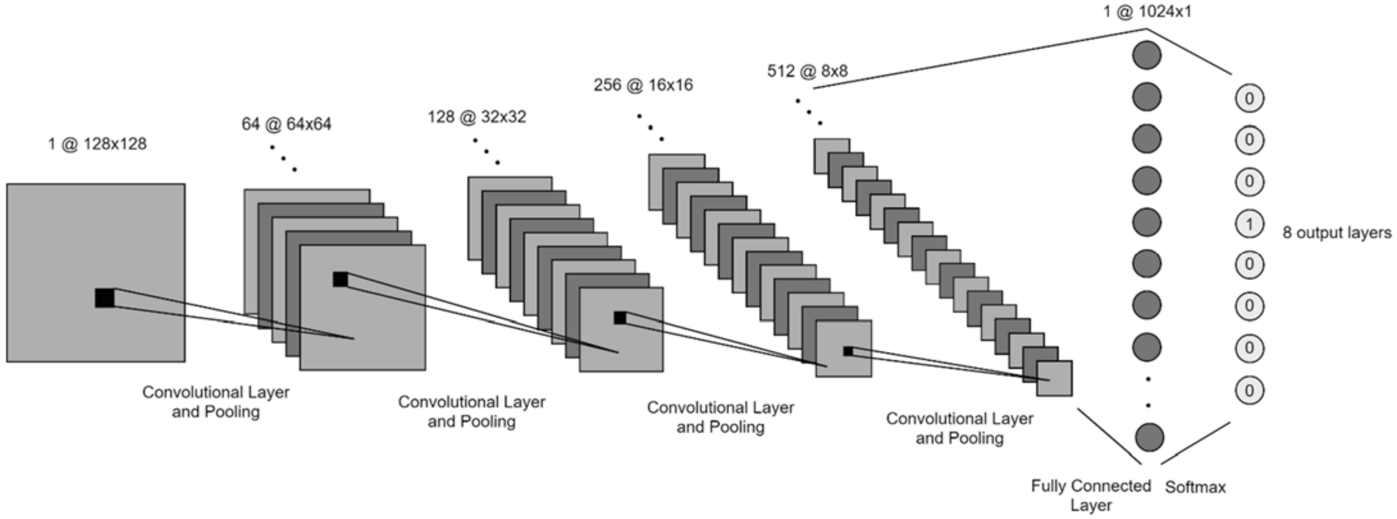
1 @ 128x128
64 @ 64x64
128 @ 32x32
256 @ 16x16
512 @ 8x8
1 @ 1024x1
0
0
0
1
0
0
0
0
8 output layers
Convolutional Layer and Pooling
Convolutional Layer and Pooling
Convolutional Layer and Pooling
Convolutional Layer and Pooling
Fully Connected Layer
Softmax

Fig. 3. The implemented convolutional neural network architecture

The neural network implemented was a deep convolutional neural network using TensorFlow. The CNN architecture is illustrated in Fig 3. All the weights were initialized using Xavier initialization and the input vector was a 128x128 pixel spectrogram. The first four layers are convolutional layers with a kernel size of 2x2 with a stride of two and a max pooling layer after each successive layer. After the first four layers, there is a fully connected layer where each output of the last layer is fed into each input of the fully connected layer. This produces a 1024 vector of numbers. A softmax layer is then applied to get 7 outputs which represent each genre.

The activation function being used throughout the network is a Rectified Linear Unit (ReLU). The optimizer used was RMSProps. To curb overfitting, a dropout with probability of 0.5 was implemented immediately after the fully connected layer during training.

The CNN was initially trained using 27 genres across 995 songs with no changes in architecture from Despois [11], a test accuracy of 47% was observed. It appeared the implementation of the CNN was overfit as the accuracy for the training data was 95% versus the test data at only 47%.

Following the initial results, modifications were made to limit the number of genres to 7 (Hip Hop, Electronic, Rock, Pop, R&B, Alternative, and Country). This was done by consolidating numerous sub-genres into the main 7 genres. In addition, the dataset was almost doubled to 1880 songs, equating to 132,000 total spectrogram slices. With the previous mentioned changes, test accuracy was determined to be 62%. Next, the activation function used in the convolutional layers was changed from an Exponential Linear Unit (ELU) to a ReLU which increased accuracy by 5%. After training the CNN architecture in Fig. 3 with the modifications mentioned above, the test accuracy for this work improved to 67%. This is in contrast to the reported validation accuracy by Despois [11] of 90%, since the test accuracy was not provided in the article.

III. Conclusions and Future Work

This paper has reviewed some of the research done in machine learning with respect to music and genre classification. It has also presented some preliminary research and implementation work for musical genre classification. The research implementation included taking songs, converting them into short time segments and representing the time segments by their respective spectrogram images. Each of these spectrograms were labelled by music genre and then used as inputs into a neural network. The neural network had four convolutional layers followed by a fully connected layer and then a softmax function at the end to calculate the probability of each genre detected and then returns a one-hot array of genre classifications. The initial results obtained were 67% accurate on the testing data.

Some future research work with respect to modifications to the algorithms include removing the very last portion of the final layer to observe what probabilities were calculated by the NN before converting to a one-hot array of values. It will also include changing the optimizer function, the initialization of the weights, and varying the activation function to better understand the effect of each of these modifications.

With respect to the dataset, future work will include determining the effect of increasing/decreasing the size of the dataset including the size of the slices, the number of slices and the number of songs. It will also include determining if it is more or less advantageous to use music data from the middle of a song rather than the beginning or the ending of a song as well as determining the effect of classifying only the high, mid, or low end of the frequencies in the songs.

For testing, the current test accuracy is being verified on 2.56 seconds of the song only. Future work will include different approaches to testing that may include increasing the verification number and length for making a final determination.

Other opportunities for future work may include using a binary classifier for determining if a given user might like a song based on the user's song library or listening preferences. Other future work may also include replacing the current CNN architecture with a recurrent neural network. An RNN may use Musical Instrument Digital Interface (MIDI) files as input and would consider the sequential context of the data fed into the network for a given song and genre.

References

[1] S. Gollapudi, "Practial Machine Learning," Packt Publishing Ltd., 2016.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning.(Report)," Nature, vol. 521, no. 7553, pp. 436, May 2015, 2015.

[3] J. Shah, "Neural Networks for Beginners: Popular Types and Applications," https://blog.statsbot.co/neural-networks-for-beginners-d99f2235efca, [January 12, 2019, 2017].

[4] O. Mogren, "C-RNN-GAN: Continuous recurrent neural networks with adversarial training," CoRR, vol. abs/1611.09904, 2016.

[5] Y. Chen, "Towards Explaining Neural Networks," Faculty of Science, Utrecht University, Utrecht, Netherlands, 2017.

[6] Tomas Mikolov, Martin Karafiat, Lukas Burget et al., "Recurrent neural network based language model," INTERSPEECH, vol. 2, pp. 4, 2010.

[7] Y. M. G. Costa, L. S. Oliveria, and C. N. S. Jr., "An Evaluation of Convolutional Neural Networks for Music Classification Using Spectrograms," 52, https://www.sciencedirect.com/science/article/pii/S1568494616306421 , [December 16, 2018, 2017].

[8] T. O'Brien, "Learning to understand music from Shazam," https://blog.shazam.com/learning-to-understand-music-from-shazam-56a60788b62f, [December 19, 2018, 2017].

[9] Y. M. G. Costa, L. S. Oliveria, A. L. Koerich et al., "Music genre recognition using spectrograms," 2011 18th International Conference on Systems, Signals and Image Processing, pp. 1-4, 2011, 2011.

[10] M. Lopes, F. Gouyon, A. L. Koerich et al., Selection of Training Instances for Music Genre Classification, p.^pp. 4569-4572, 2010.

[11] J. Despois. "Finding the genre of a song with Deep Learning - A.I. Odyssey part. 1," December 27, 2018; https://hackernoon.com/finding-the-genre-of-a-song-with-deep-learning-da8f59a61194.

[12] L. Norskog, "SoX - Sound eXchange," Sourceforge, 2015.