Song Classification Using Neural Network

Anmol Rattan Singh, Shivang Sachdev, Purti Shah

Guided By: Jaychand Upadhyay

Department Of Information Technology
Xavier Institute Of Engineering
Mahim, Mumbai

***Abstract*—This paper deals with the classification of songs into pre defined categories on the basis of musical features extracted from the mp3 files by using echonest API. Features such as loudness, tempo and timbre are extracted and fed into a multi-layer feed-forward neural network for classification. The dataset used for training of the network is acquired from the freely available Million Song Dataset [2]. The neural network model uses error-back propagation to dynamically adjust weights during training process. It uses python dictionary function to store those weights within each neuron and later uses them for classification of test data**

Keywords—song; classification; genre; neural network; timbre (key words)

I. INTRODUCTION

Machine learning is the branch of computer science that has to do with building algorithms that are guided by data. Rather than relying on human programmers to provide explicit instructions, machine learning algorithms use training sets of real-world data to infer models that are more accurate and sophisticated than humans could devise on their own.
Within the field of machine learning, neural networks are a subset of algorithms built around a model of artificial neurons spread across three or more layers.

Neural networks have been used for a wide array of classification tasks. Artificial neural networks are relatively crude electronic networks of neurons based on the neural structure of the brain. Such systems "learn" (i.e. progressively improve performance on) tasks by considering examples, generally without task-specific programming.An ANN is based on a collection of connected units or nodes called artificial neurons (a simplified version of biological neurons in an animal brain). Each connection (a simplified version of a synapse) between artificial neurons can transmit a signal from one to another. The artificial neuron that receives the signal can process it and then signal artificial neurons connected to it.
Neural networks are well-suited to identifying non-linear patterns, as in patterns where there isn't a direct, one-to-one relationship between the input and the output. Instead, the networks identify patterns between combinations of inputs and a given output.

Due to the high dimensionality of musical data, it is impossible for simple classifiers such as k-NN and SVM to provide a satisfactory result.

Music genres are high-level descriptors used by the listeners, dealers, organizations and institutes to describe and organize their audio collections. Classifying songs into genres is an important application of research currently being carried out in the field of Music Information Retrieval (MIR). Despite having considerable about of data at hand, the task cannot be carried out intelligently and has to rely on metadata tags for proper classification. Upon the tag being absent, the song does not get classified and hence remains an outlier. Manual classification of a song poses a conundrum regarding the accuracy of the classification. Since very few genres have clear definitions and there is often significant overlap among them. Also, classifications tend to be by artist or album rather than by individual recordings, and metadata found in mp3 tags tend to have unreliable annotations.

Further the result would be subjective to the understanding of various genres and their signatures possessed by a user. The task of manually classifying a song is tedious and can be better handled by an intelligent system.

We propose a model of artificial multilayer feedforward neural network to automate the task of music classification. Song features such as loudness, tempo and timbre values can be extracted from mp3 files dynamically and fed into a trained neural network for the purpose of classification into any one of the pre defined genres.

This paper is organized as follows: Section II reviews upon the previous work done in the field of song classification,

Section III explains the process of feature extraction and list of all features being extracted for classification purpose, Section IV describes the dataset used for the purpose of training the neural network, Section V explains the system architecture of the neural network model, Section VI displays the preliminary experimentation results of training and testing the network.

II. PREVIOUS WORK

A. *Mel Frequency Cepstral Coefficients (MFCC)based.*

A number of classification models have been used to classify music based on it their MFCC features. Traditional models have shown more promise when dealing with MFCC data to classify songs. K-means, k-nearest neighbour and traditional clustering have proved more accurate than smarter models that involve supervised and/or unsupervised learning, namely neural networks and deep belief networks. .

B. *Harmonic and drum component based*

An alternate but lesser known approach to genre classification came from extraction harmonic and drum components of a song. The model was highly successful in classifying songs into genres that characteristically used heavy percussion-type instruments such metal and rock. Support Vector Machines proved successful to a great extent in dealing with harmonic data due to their superiority in supervised learning over other intelligent models.

C. *Instrument Signature Sound based*

It was argued and generalized that a specific genre only employs a specific subset of instruments that lend it its characteristic sound signature. The approach failed with newer upcoming genres that fused two primitive genres to create a fusion and also with electronically produced music. The accuracy of classification was also fairly limited if the number of instruments used was exceeding 5.

D. *Mood based*

Mood based classification of songs is done using a combination of timbre-based features and tempo-based features. The limitation of using tempo as a vector for mood classification is the ambiguity regarding the lyrical content of the song. Mood based classifiers do not take into account the actual lyrics of the song which leads to some sad songs being categorized as happy due to their upbeat tempo.

E. *Lyrics based*

Mood based classification of songs is done using a combination of timbre-based features and tempo-based features. The limitation of using tempo as a vector for mood classification is the ambiguity regarding the lyrical content of the song. Mood based classifiers do not take into account the actual lyrics of the song which leads to some sad songs being categorized as happy due to their upbeat tempo and vice versa.

III. FEATURE EXTRACTION

For each segment of the track the various extracted audio features will be as follows - a 12 dimensional vector of chroma features, a 12 dimensional vector of 'MFCC-like' timbre features and various measures of the loudness of the segment, including loudness at the segment start and maximum loudness. The extracted data will also include a 25 dimensional vector for each included segment, consisting of the 12 timbre features, 12 chroma features and loudness at start of segment concatenated in that order. Other features such as time-key and duration of the song will also be available but will be immaterial to the project.

A. *Loudness*

Loudness is an important factor if we look at the songs of different genres. Comparing a classical or a jazz song with that
of the Rap or Pop, the later is found to have greater loudness
then the earlier one. The loudness is measured in decibels (dB)
and it can be correlated to the amplitude of the sound wave.
Here loudness is determined by combining each segment obtained from the json file created using Echonest API and then determining the average and variance of the whole song
considering each and every segment of the song.

B. *Tempo/Beats Periodicity*

The first thing that comes to every music lover is the beats of a particular song. The beats complexity and periodicity gives the song its beauty. Moreover and technically, it has been observed that most of the songs in a particular genre have nearly the same beat periodicity. That is why first feature included in the set is beats per minute. Along with beats per minute (bpm) average, its variance and skewness are also determined so that entire model of a song can be approximated well.

$$bpm = \frac{Total\_beats}{Duration\_of\_song} \quad \ldots\ldots\ldots(1)$$

C. *Timbre*

Timbral Texture features These features are used to differentiate mixture of sounds that possibly have similar pitch and rhythm [8]. The features used to represent timbral texture are based on standard features proposed for music-speech discrimination [18]. To extract the timbral features, audio signals are first divided into frames by applying a windowing function at fixed intervals. The window

function of this research is hamming window which helps to remove the edge effects. Timbral texture features in Fig.2 have been computed and later we calculated different statistical values like mean, standard deviation, skewness, kurtosis, and covariance matrix from feature values. The mean (μ) and standard deviation (σ) for frame-wise feature values (Xn) in a N -frame song are given by

$$Mean(\mu) = \frac{1}{N}\sum_{n=1}^{N} X_n \quad \text{.....(2)}$$

$$Std(\sigma) = \frac{1}{N}\sum_{n=1}^{N} (X_n - \mu^2) \quad \text{.....(3)}$$

IV. THE DATASET

The Million Song Dataset is a freely-available collection of audio features and metadata for a million contemporary popular music tracks. Attractive features of the Million Song Database include the range of existing resources to which it is linked, and the fact that it is the largest current research dataset in our field. The core of the dataset comes from The Echo Nest API. This online resource provides metadata and audio analysis for millions of tracks and powers many music applications on the web, smart phones, etc.

The MSD contains audio features and metadata for a million contemporary popular music tracks. It contains: • 280 GB of data • 1, 000, 000 songs/files • 44, 745 unique artists • 7, 643 unique terms (Echo Nest tags) • 2, 321 unique musicbrainz tags • 43, 943 artists with at least one term • 2, 201, 916 asymmetric similarity relationships • 515, 576 dated tracks starting from 1922.

The data is stored using HDF5 format 2 to efficiently handle the heterogeneous types of information such as audio features in variable array lengths, names as strings, longitude/latitude, similar artists, etc. The main acoustic features are pitches, timbre and loudness, as defined by the Echo Nest Analyze API. The API provides these for every "segment", which are generally delimited by note onsets, or other discontinuities in the signal. The API also estimates the tatums, beats, bars (usually groups of 3 or 4 beats) and sections.

VI. SYSTEM ARCHITECTURE

Our approach is to use machine learning to develop a musical style recognizer. Recognition can be viewed naturally as a supervised learning task. The learner is given examples of music in the specified style and then is deployed to classify songs based on the supervised learning. However, musical style recognition is different from traditional supervised learning tasks due to the free-form nature of the input, and the depth of structure. Considerable care is required in the design of the learning model such as choosing an appropriate learning rate. We have chosen back-propagation neural networks for our learning approach for a number of reasons. They have an excellent track record in complex recognition tasks in problems such as face recognition. They are capable of inducing the hidden features of a domain that may be elusive to a rule-based approach. Their flexible design allows us to encode musical structure in the model. They have been successful in other musical tasks, such as composition.

A supervised learning is performed using sigmoidal activation function. The neural network model used here is based on the Feedforward Multi-Layer Perceptron model. This model can be said to be working similar to our brain. Using an optimized learning rate, the model uses supervised learning over a dataset of 50,000+ songs which is divided into 10 genres.

The neural network model will consist of three layers, input layer, hidden layer and output layer. The input layer will consist of 26 neurons, which will receive input of 26 different song parameters. The parameters will consist of loudness, temp, 12 inputs consisting of average timbre of songs divided into segments and another 12 of timbre variance of the the same segments.

The neural network is designed and implemented in python. Python, due to its advanced machine learning capabilities, allows to make extremely flexible and robust neural network designs. The training and testing dataset is a CSV file acquired from the million song dataset website. It is called the derived genre dataset. It consists of all the aforementioned audio features of 53,960 songs spread over 10 genres. The CSV file is fed into the neural network and network trains in a supervised manner. Later the model is deployed to classify songs according to the audio features fed into it after acquiring them from arbitrary songs.

Various in-built dictionary functions act as a storage medium for various weights as a neuron comes across various auditory cortex in training phase. The cortex such as loudness and tempo are dealt differently than those of timbre due to their unrelated nature to each other and their individual influence over song genre.

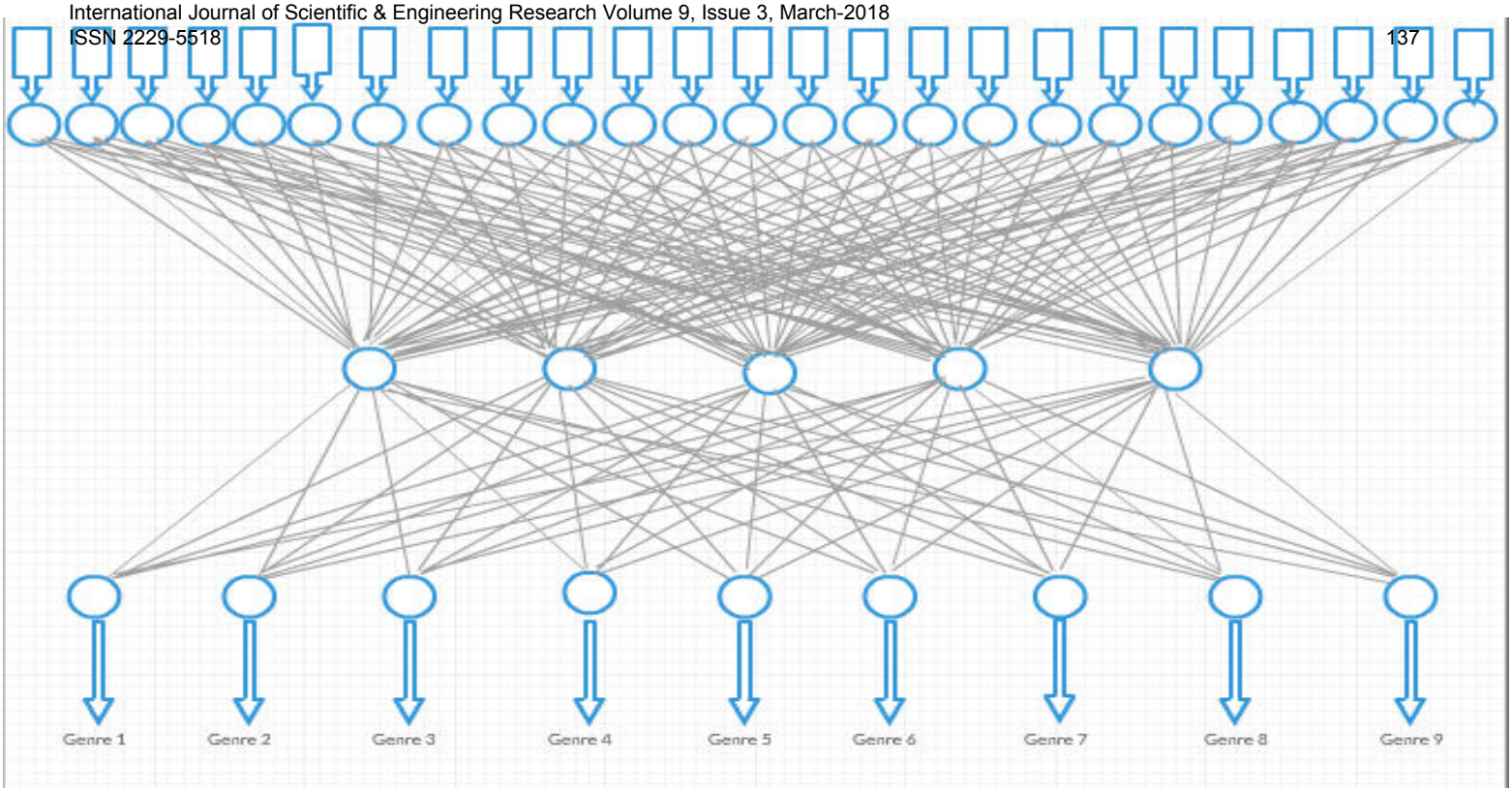

Figure 1: Neural network model connection

The network learns via a sigmoidal function.

$$f(x) = \frac{1}{1+e^{-x}} \text{ .....(4)}$$

It iterates the input through the neural network over a number of epochs, each time reducing the error in its computation and essentially learning the right weights attached with each neuron, associated with each input.

The network starts off with random weights attached to each neuron. The net input of each neuron is calculated as

$$f_{net} = w_t * i_t \text{ .....(5)}$$

After the output of a neuron is calculated, it acts as input value for the next neuron. Finally, a result is obtained from the output layer. After the output is obtained, the error function is calculated.

$$\Delta_t = r_t - o_t \text{ .....(6)}$$

Where r_t is the expected result and o_t is the obtained result.

The error function is obtained by multiplying the calculated error with the derivative of sigmoidal function and the learning rate of the neural network.

The value obtained is the change in weight for a neuron. Multiple epochs help the network get tuned better for any kind of data as input.

The learning rate is how quickly a network abandons old beliefs for new ones.

The learning rate is a relatively small constant that indicates the relative change in weights. If the learning rate is too low, the network will learn very slowly, and if the learning rate is a little too high, the network may oscillate around minimum point, overshooting the lowest point with each weight adjustment, but never actually reaching it. Usually the learning rate is very small, with 0.01 not an uncommon number.

VI. EXPERIMENTATION AND RESULTS

Learning Rate	Epoch	Mean Accuracy (%)
0.05	100	62.865
0.05	200	68.723
0.1	100	68.544
0.1	150	70.233
0.1	200	73.437
0.2	100	71.683
0.2	150	74.861
0.2	200	76.588
0.3	100	72.379
0.3	200	76.923
0.3	300	82.899
0.3	400	84.165
0.3	500	84.532

Table 1: Test results for ANN

Starting with a very low learning rate of 0.05, the accuracy achieved was really low (as seen from the table). This can be attributed to the fact that the gradient descent curve was not steep enough. With epochs as low as 100, the model was not able to learn fast enough. Hence, increasing the epochs to 200, we see a sharp increase in accuracy of 6%.

Consequently, we get the same accuracy by increasing the earning rate to 0.1. This signifies that with a higher learning rate, the change in weights was considerable thus making the gradient descent curve steeper and hence enhancing learning. Moreover by keeping the learning rate constant and increasing the number of epochs, the accuracy increased significantly.

This also points to the fact that there is considerable learning to do for the model before the sigmoidal curve becomes constant and the weights reach their saturation point.

Jumping forward to the last few observations in the results, with a learning rate of 0.3 we observed the model was changing weights very quickly hence until 400 epochs, the accuracy increased steadily with an average of 2.5% per 100 epochs. Beyond that, the increase in accuracy fell to 0.4%.

This can mean one of two things. One is that the model is stuck in a local minima of the gradient descent curve and is unable to train further. This problem is consistent with simple architecture models that are fed high dimensionality datasets with overwhelming number of data points to process. The proximity of data points fools the network into learning and remembering patterns rigidly, thus losing out on adaptability and generalization abilities of the network.

Another reason could be that the network is oscillating around global minima,thus is unable to converge onto it. The change in weights needed near the lowest point of the curve is very small, hence a higher learning rate does not bode well for such a scenario. Near the minima, the model needs to learn very fine patterns hence requiring a lower learning rate. A higher learning rate of 0.3 affects accuracy at the lowest point of the gradient descent curve, making the model unable to learn past a certain point.

To overcome this problem, an algorithm is proposed that involves reducing the learning rate gradually the number of epochs progress. The algorithm is called a "Time Based Learning Rate Decay" algorithm. Although it is not a very popular choice in supervised learning, the algorithm ensures the model to converge nearer the answer if not on it. Applying the algorithm, the highest accuracy achieved was 93.62%,thus proving our case.

VII. CONCLUSION

The propose work uses a total of twenty-six features and applied upon the ten genres that are Classic Rock & Pop,Folk, Punk, Pop, Hip-Hop,Jazz, Metal, Rock, Dance Electronic and Soul Raggae songs. The results obtained with an overall accuracy of 85%. The structure designed can be implemented on distributed or multi-processor environment to determine the results faster i.e., on real or run time only. A small in decrease in the accuracy can be because of the fact many genres have overlapping similarity of features of natural songs and speechiness patterns. This work can be successfully implemented for other genres around the world as well. It can also be used be used in the music selling websites allowing them to reduce their overhead of classifying and increasing the success rate among their customer because manual work may still have confusion among the operator about the different genres and also require them to have prior knowledge about it. With this no prior knowledge is needed. This work can also be extended for songs which are a mixture of two genres like Jazz-Rock, Rock-Rap,Classical-Rock etc

VIII. REFERENCES

[1] Anshuman Goel,Mohd. Sheezan,Sarfaraz Masood, Aadam Saleem. 'Genre Classification of Songs Using Neural Network". 2014 5th international conference on computer and comm. Technology.

[2] Thierry Bertin-Mahieux, Daniel P.W. Ellis
Brian Whitman, Paul Lamere. "THE MILLION SONG DATASET"

[3] Babu Kaji Baniya*, Deepak Ghimire, Joonwhoan Lee
Division of Computer Science and Engineering
Chonbuk National University, South Korea. "Automatic Music Genre Classification Using Timbral Texture and Rhythmic Content Features"

[4] Vikramjit Mitra and Chia J. Wang, "A Neural Network based Audio Content Classification", Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA, August 12-17, 2007.

[5] Silla, C. N., Alessandro L. Koerich, and Celso AA Kaestner. "Feature selection in automatic music genre classification", Multimedia, 2008. ISM 2008. Tenth IEEE International Symposium on. IEEE, 2008.

[6] Fu, Zhouyu, et al. "Learning naive Bayes classifiers for music classification and retrieval", Pattern Recognition (ICPR), 2010 20th International Conference on. IEEE, 2010.

[7] Miyoshi, Masato, et al. "Feature selection method for music mood score detection", Modeling, Simulation and Applied Optimization (ICMSAO), 2011 4th International Conference on. IEEE, 2011.

[8] Susheel Sharma1, Rakesh Singh Jadon, "Mood Based Music Classification"

[9] "Music Genre Classification", Archit Rathore - 12152, Margaux Dorido - EXY1420, Indian Institute of Technology, Kanpur, Department of Computer Science and Engineering