Music Classification using DNN's

Amlan Kar, Chaitanya Ahuja
Advised by
Prof. Amitabha Mukherjee

April 18, 2015

Abstract

Music Classification is a harder task as compared to speaker classification, due to presence of polyphonic sounds in the in signal. Hence signal-processing techniques fail to produce usable competent results. We attempt to deploy a deep neural network for extracting features from different genres and artists of music. This system would be then used for classifying unknown music signals. As opposed to a traditional neural network classification by soft-max margin , we just plan to extract features from the hidden layers and use them as the basis of classification for classification techniques like Random forests.

1 Introduction

Music Classification is a well known problem and has been researched fairly well in literature. From techniques involving traditional **signal processin**g approaches involving handcrafted features (FFT, Cepstrum) and modern learning algorithms like **Random Forests(RFs) and Deep Neural Networks(DNNs)** [1], music classification reduces to a problem of good feature extraction. **AdaBoost and Aggregation** of features is used in [2, 3] for the same purpose.

Stochastic Gradient Descent (SGD) has been a prime contender for training neural networks for the past 3 decades, but its inherent time complexity to train the network for large datasets can make it impractical for a large number of situations. As a remedy to this, Rectified linear units (ReLUs) can be used as shown in [1], but they tend to overfit the data. We follow a traditional approach in terms of the selection of the iteration method and work with SGD and activation functions as biased sigmoids.

Figure 1: Abstraction of the problem statement

Our basic problem statement, as demonstrated in a block diagram in Figure 1, is creating a model which is able to classify music clips based on genres and artists. This problem stands as a classic problem in the Music Information Retrieval domain and we have attempted to experiment with a few techniques.

The rest of the report is organized as follows. Section 2 starts of with the basic theoretical concepts involving neural networks and classifiers. In section 3, we discuss the methodologies and experiments tried in this project, while section 4 connotes to the results obtained for these methods. The project has been finally concluded in Section 5.

2 Theory

2.1 DNN

Neural networks are a collection of perceptrons, which when stacked up in form of layers, leads to a network. If this network contains more that 2 layers (Input and Output), it is called a Deep Neural Network(DNN). A toy image is shown in Figure 2. To calculate the outputs on each layer, we need the weight matrix. A weight matrix W consists of each weight between the 2 layers. Also, we need the bias vector $\mathbf{b}$ which is basically the weights of the bias terms. Ofcourse, we also need the activation functions $f(x)$ which in this case is the sigmoid function $\left[\frac{1}{1+\exp(-x)}\right]$ for all the nodes. So for input $\mathbf{x}$, the output at the first layer would be

$$\text{out} = \frac{1}{1 - \exp^{-(W\mathbf{x}+\mathbf{b})}} \tag{1}$$

Using these basic equations and SGD and soft-max margin, the neural network is trained to give a classification and feature extraction model.

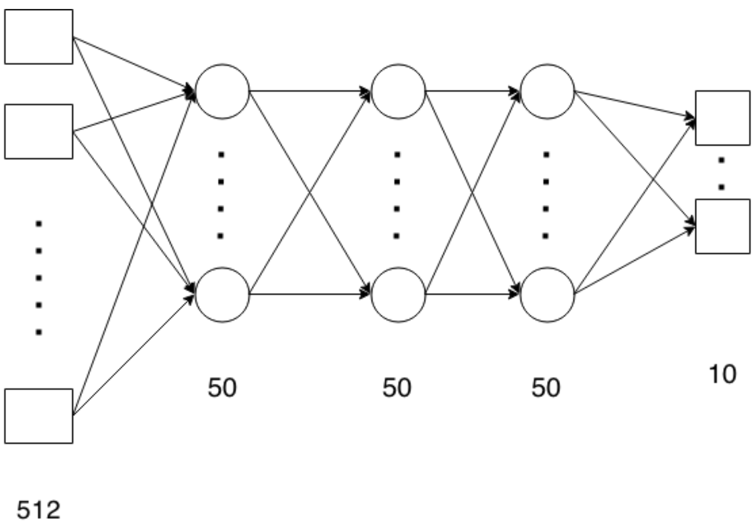

Figure 2: Structure of the neural net used in the project

2.2 Dropout

Dropout is a neat method to prevent overfitting in neural networks as shown in [4]. Basically, dropout gives a probability p to each node which determines its presence during a training epoch. This ensures that the weights do not get too tuned to the data. This is not applicable during testing, and each node is present normally during testing. Figure 3 gives a sketch of the dropout method.

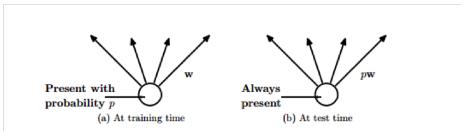

Figure 3: Abstraction of the concept of dropout

2.3 RF

Random forests are a basically a collection of weak learner set of decision trees as explained in [5]. At random, data points are selected with replacement to form a bag. This bag is used for training a decision tree model. Repeating this process for n number of trees, we get a set of decision trees which is formally known as a random forest. The classification of any input comes as a result to maximum pooling of the output classification. Figure 4 represents the bagging of the training data.

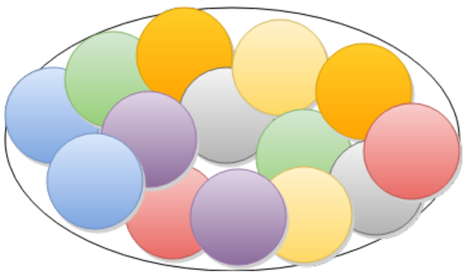

Figure 4: Bagging of training set in Random Forests

2.4 HMM

A Hidden Markov Model is a special bayes net with properties that make it particularly applicable to temporal data modelling [6]. They have been widely used in temporal pattern recognition problems viz. speech recognition, gesture recognition, part-of-speech tagging etc. Using HMMs to generate features doesn't seem to be a very well-researched topic and we have made an attempt towards it in our project.

3 Methodology and Experiments

Moving on to the methodology used on the problem statement, we tried 3 different approaches which are explained as follows. Also Figure 5 gives the gist of the whole flow of methods.

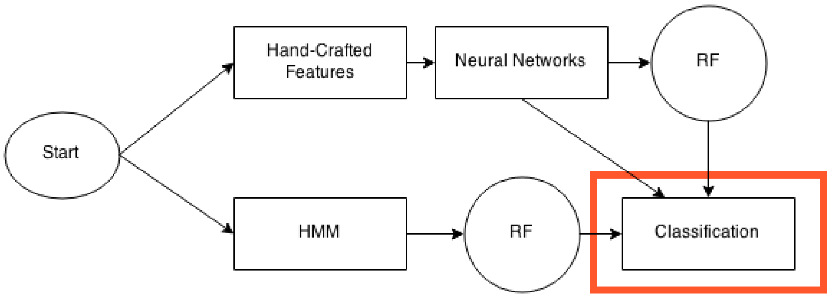

Figure 5: Abstraction of the concept of dropout

3.1 Case 1

We trained a neural network of the form 513-50-50-50-10 which is shown in Figure 2[1]
Input: FFT of a 512 point window of 30s audio clip.
Output: Probabilities of the classification output.
Activation function: Sigmoid function $\frac{1}{1-\exp^{-(Wx+b)}}$

3.2 Case 2

Use the first hidden layer to train a RF classifier to predict classes
Input: FFT of a 512 point window of 30s audio clip.
Output: Probabilities of the classification output.
Activation function: Sigmoid function $\frac{1}{1-\exp^{-(Wx+b)}}$

3.3 Case 3

Use MFCC features to train an HMM model. The features acquired from this model are then used to train an RF classifier.

3.4 Dataset

For Genres we used GZTAN [7] dataset containing 10 genres with 100 30s audio clips. **For Artists we created a similar dataset using music of 10 artists of the genre (blues).**

[1] Code Adopted from: https://github.com/sidsig/ICASSP-MLP-Code

4 Results

After training the neural nets on the training data(90%) for 50 and 500 epochs, we tested the results on the testing data (remaining 10%). The HMM model was also trained on the same datasets to maintain comparability. The benchmark in Table 1 has been taken from [1] and Accuracy column has been calculated by certain modifications to the code provided on *https://github.com/sidsig/ICASSP-MLP-Code*

Method	Accuracy	Benchmark
DNN-50 epochs	0.48	NA
DNN-500 epochs	0.56	NA
DNN-RF-50 epochs	**0.62**	0.718
DNN-RF-500 epochs	**0.63**	0.656
HMM-RF	0.42	NA

Table 1: Genre Classification on GZTAN dataset

Method	Accuracy
DNN-50	**0.7573**
DNN-RF-50	**0.8738**

Table 2: Artist Classification on Self-Created dataset [2]

4.1 Confusion Matrices

There was a sudden rise in accuracies of the artist classification, which was surprising. Hence we decided to find the confusion matrices to analyze the false positives of the classifier. Figure 6 and 7 correspond to the confusion matrices of genres and artists respectively.

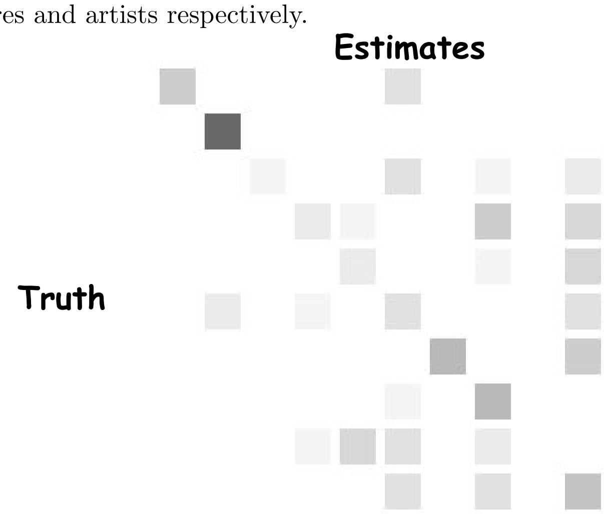

Figure 6: Genre Confusion Matrix. Higher intensities imply higher value

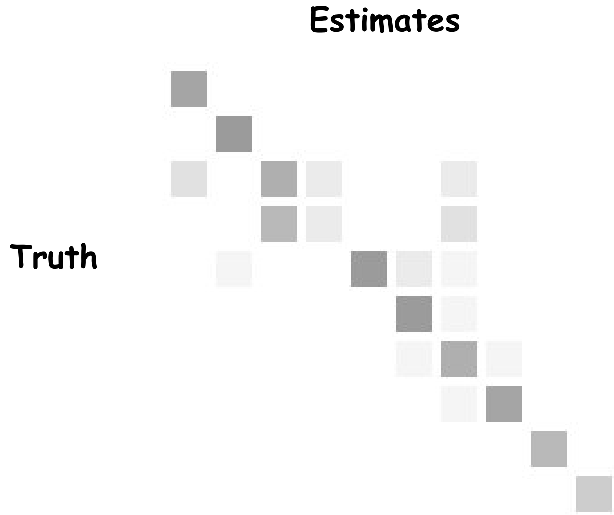

Figure 7: Artist Confusion Matrix. Higher intensities imply higher value

Clearly, in Figure 7 the 4^{th} artist (Cream) has been classified as 3^{rd} (Clapton) for more that its actual class. Though, this was expected as both the bands have the same lead guitarist. Hence the model was able to successfully extract the guitar music features but got confused with the band it belonged to. Perhaps, Clapton has more dominant features of the guitarist, hence the testing yields results pointing towards Clapton rather than cream.

5 Conclusion

1. The accuracies obtained for a simple DNN is surprisingly well, because it is able to perform at **around 50% accuracie**s on 513 points of data.
2. After **aggregation of different frames** of a given audio and using maximum pooling in an RF classifier, the accuracy boosts up by a significant percentage.
3. Neural Network Models have been successful in extracting music features as it was demonstrated by the confusion matrices in Section 4.1
4. HMM modelling have been shown to extract useful features with speech signals [6], but they do not seem to work well for music features.

6 Future Work

1. Dimensionality Reduction on feature vectors
2. Using HMM features on Neural Net
3. Expanding artist classification problem to multiple genres
4. Using autoencoders for feature generation
5. Extend the better feature extraction to transcription of music notes

7 References

References

[1] Siddharth Sigtia and Simon Dixon. Improved music feature learning with deep neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 6959–6963. IEEE, 2014.

[2] James Bergstra, Norman Casagrande, Dumitru Erhan, Douglas Eck, and Balázs Kégl. Aggregate features and adaboost for music classification. *Machine learning*, 65(2-3):473–484, 2006.

[3] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *Speech and Audio Processing, IEEE transactions on*, 10(5):293–302, 2002.

[4] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[5] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[6] R Gajšek, F Mihelič, and S Dobrišek. Speaker state recognition using an hmm-based feature extraction method. *Computer Speech & Language*, 27(1):135–150, 2013.

[7] Thierry Bertin-Mahieux, Daniel PW Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *ISMIR 2011: Proceedings of the 12th International Society for Music Information Retrieval Conference, October 24-28, 2011, Miami, Florida*, pages 591–596. University of Miami, 2011.