Music genre classification using Deep Neural Network

Konstantin Sozinov
Royal Institute of Technology
sozinov@kth.se

Albina Shilo
Royal Institute of Technology
shilo@kth.se

Abstract

Music genre classification is a challenging task of Music Information Retrieval. Many attempts is made to solve the task and good result is achieved, for example, with Gaussian mixture model classifier and Convolutional Neural Network. In this project we apply conventional Deep Neural Network feeding MFCC to classify 10 music genres. The experiments revealed that error rate strongly depends on number of classifying genres. This conclusion is also supported by visualization made with t-Distributed Stochastic Neighbor Embedding which illustrate that MFCC features of genres have vague boundaries. In addition, an analysis of potential improvements is presented.

1 Introduction

Genre classification task becomes more and more popular nowadays increasing its applications. Mainly, it is used to accelerate the classification process and substitute expensive manual labeling. However, the musical genre classification is a challenging task due to fuzzy nature of genre boundaries.

There are many works related to the task were presented recently. Tzanetakis and Cook in [9] used the parametric Gaussian mixture model (GMM) classifier and nonparametric K nearest neighbors (KNN) classifier to classify 10 genres. The main result achieved is **60%** accuracy for non real-time models and **40%** accuracy for real-time Gaussian model classifier.

Recently, deep learning method became very popular solution for the classification problem. One of the deep learning approaches, which is successfully used in image classification, is Convolutional Neural Network (CNN). CNN is used to classify genres in [7] by Li, Chan and Chun. In their work many experiments are performed in order to analyze influence of different parameters on CNN model performance. They concluded that when number of the genres in classification increase, it is harder to train a CNN and achieve a good result. For example, training on 3-genre dataset converges faster than on 6-genre dataset. Therefore, they used ensemble training to improve performance of the classification of 10 genres. The best result is **84%** accuracy on the test data before the majority voting and the accuracy increases after taking the majority voting.

A very successful approach that was proposed in [2], is transfer learning. Authors introduced a pretrained convnet feature, a concatenated feature vector using activations of feature maps of multiple layers in a trained CNN. By using the pretrained features they were able to achieve a performance of 89,8% on the same dataset that was used in this paper (10 different music genres) which is a impressive result.

Another deep learning approach which can be used for music classification task is Deep Neural Network (DNN). Feng in [4] compare regular Neural Network (NN) and Deep Belief Network (DBN) with pretraining using Restricted Boltzman Machine (RBM). Experiments revile the same performance (around **70%** accuracy for 3 genres) for both types but DBN becomes worse when number of genres increase due to overfitting. This article is the closest paper to our chosen project topic.

The main task of the project is to evaluate a simple DNN on a task of music genre classification using with different number of genres (10, 3 and 2 genres). The project is constructed as follows: Dataset section provides information about the dataset is used to train and test the network. Method section consists of a feature extraction procedure from audio files and a description of network architecture including details and reasoning of chosen ReLU activation function and optimization of cost function. Visualization of features boundaries in Experiment section describes the difficulty of genre classification task. Main findings of experiments with different numbers of genres described in Classification results subsection of Experiment section. Final conclusion is placed in Discussion and Conclusion section.

2 Dataset

GTZAN Genre Collection dataset is used for the experiment in this project. The dataset consist of 1000 audio samples (30 seconds long each). There are 10 genres, which are represented by 100 tracks. The genres are reggae, classical, country, jazz, metal, pop, disco, hip-hop, rock and blues. Tracks are 22050Hz Mono 16-bit audio files in Au file format [8]. Tracks are converted to Wave form Audio File Format (WAV) in order to take advantage of a software library providing feature extraction described in more detail in the next section.
For the experiment performance evaluation the whole dataset is divided into three sets: training 60%, validation 20% and test 20%.

3 Method

3.1 Feature selection: Mel frequency Cepstral Coefficient (MFCC)

In traditional machine learning problems where DNN model is used, the features are represented simply as $N \times M$ matrix, where N is the number of examples in the dataset and M is the dimensionality of the data. In order to extract the feature information from the audio files from the dataset, MFCC feature extraction procedure is used. According to [9] MFCC features perform better in music genre classification task than other feature types.

MFCC features are created in the following way: the original 30 sec audio of each song (sampled with frequency 22050 Hz) combined into frames, each frame is 12.8 sec and 4.8 sec shift. The Hamming windowing then applied in order to emphasize the main frequency of the signal. Further, Fourier transform with 512 samples is applied to the windowed signal and then mapped into Mel frequency bank, by calculating triangle filters based on Mel scale and grouping frequencies into 26 frames. Finally, Cosine transform is applied to Mel frequency bank (filter bank) and only first 13 coefficients are kept because higher coefficients representing higher frequencies contains less information, hence, can be neglected. All parameters for the MFCC extraction were not found during this project but rather assumed to be optimal according to following article [3].

At the end of the MFCC feature extraction chain one training song is represented by a matrix with a dimensionality of 4133×13. Next, if we concatenate all songs into one big training matrix it would result to a matrix of $(4133 * 60 * 10 \times 13) = 2479800 \times 13$, where 60 is the number of training songs for one label and 10 is the number of the labels. Each row in this big matrix corresponds to a frame extracted with the MFCC for one particular genre. This is a lot of information that can not be processed by a simple deep neural network that is used in this project. In order to perform the experiments, the amount of the frames in each song are reduced by following procedure: first each song is divided in 4 roughly equal parts, then from each part of the song only a small amount of the MFCC frames are taken and concatenated together into a smaller MFCC representation of the song. The procedure is illustrated on Figure 1. This would lead to a feature matrix (for one song) of $(138 \times 13) * 4 = 552 \times 13$ which approximately corresponds to a song with length of 4 seconds, cropped from different locations of the MFCC representation of the song. The reason why this technique was applied, is an experiment on college students provided by [9]: a human is able to classify a genre with 70 % accuracy after only listening to 3 seconds of the song, hence we decided to apply same technique in the experiments.

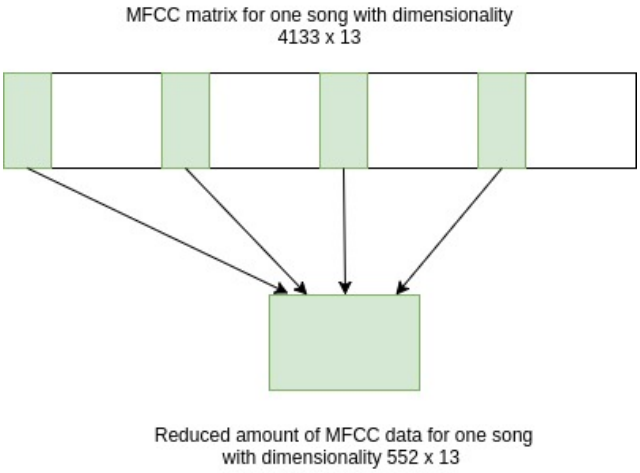


Figure 1: Technique applied to minimize amount of the MFCC frames for each song.

3.2 Network architecture

As a baseline for the classification of the given 10 genres, we choose a DNN with 4 hidden layers, 256 nodes in each layer and a softmax layer on the top of hidden layers that calculates the cross-entropy between the predictions and an 1-hot encoding of the labels (in this case music genres). In order to introduce nonlinearity in the model, a Rectifier Linear Unit (ReLU) activation function is used. It has been shown that in terms of the training time with gradient decent, ReLU is much faster than $f(x) = tanh(x)$ (Tanh) or $f(x) = (1 + e^{-x})^{-1}$ (Sigmoid) [6]. The cost function of the optimization is then defined as:

$$J(\beta, \lambda, \Theta) = \frac{1}{n} \sum_{i=1}^{n} l_{cross}(x_i, y_i, \Theta) + \lambda \sum_{i=1}^{k} ||W_i^2|| \tag{1}$$

where Θ is model used for training, β is a random subset of the data, called mini-batch, with size of 128 samples and λ is a regularization term to prevent the model from overfitting, in our case $\lambda = 0.001$. The optimization of the given cost function is done with mini-batch Gradient Decent and Momentum as an adaptive learning rate algorithm. To be able to train the model with high learning rate and be less careful about model parameter initialization, a technique called batch normalization [5] is used after each linear transformation of one mini-batch.

In order to get better classification results model's input should be pre-processed. As pre-processing procedure we used MFCC features normalization (get unit variance) and zero-centering (mean subtraction).

4 Experiment

In this section, we describe different experiments that are conducted for this project. Since MFCCs are a high dimensional input we decided first to visualize the MFCCs to get an understanding of how the features are distributed in a 2 dimensional space. After the visualization the comparison between classification of 10, 3, and 2 genres is performed.

4.1 Visualization with t-Distributed Stochastic Neighbor Embedding (t-SNE)

For better understanding of high dimensional data, a technique called t-SNE was applied on the MFCC representation of the 10 genres. The algorithm described in [10], visualizes high dimensional data by giving a location on the two or three-dimensional map. By observing scatter plots obtained from the t-SNE, an important conclusion can be made - if we consider all genres, the data is not linearly separable, see Figure 2. However, if we reduce the number of the genres to two, metal and classical, we can observe that genres contains completely different MFCC information and can be

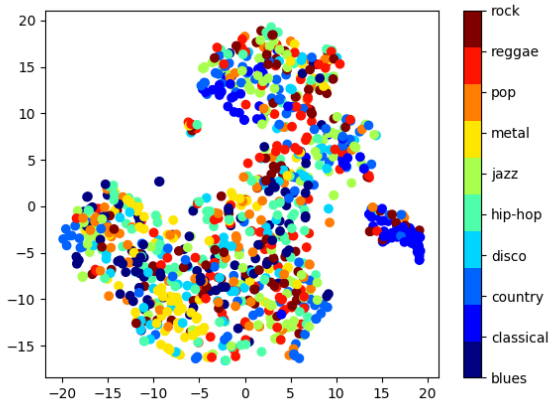


Figure 2: t-SNE for all genres

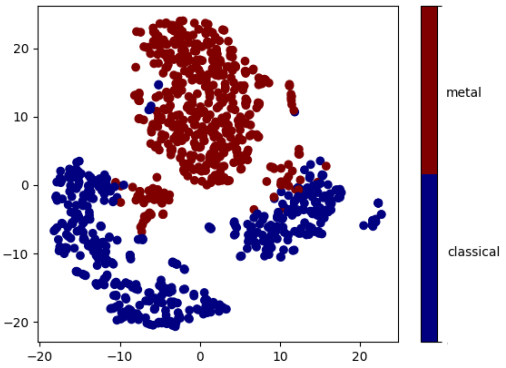


Figure 3: t-SNE for 2 labels: metal and classical

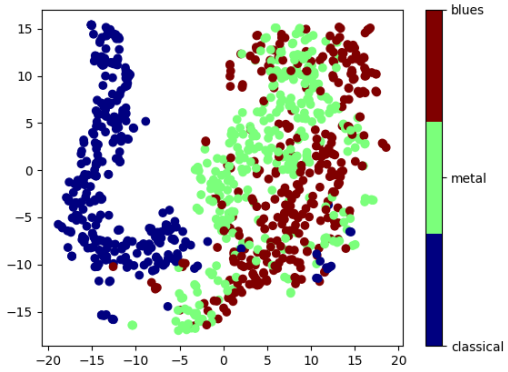
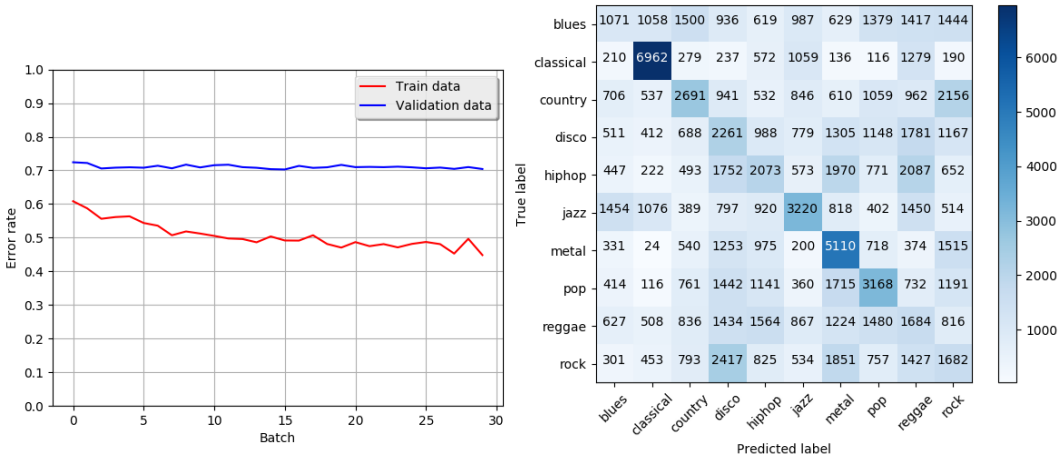


Figure 4: t-SNE for 3 labels: classical, metal and blues

separated easily, see Figure 3. If we increase number of genres to 3: classical, metal and blues the t-SNE representation of the MFCC data becomes more dense on borders but still relatively easy separable (Figure 4).

4.2 Classification results

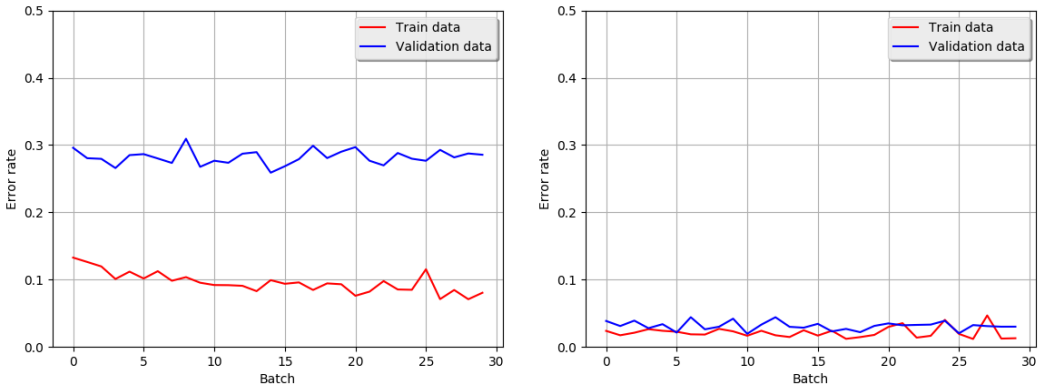
As described earlier in this section, first experiment consist of a challenging task of classification of 10 different genres. The model was trained for 30 epochs and evaluated at the end of every epoch by calculating error rate on the train and validation data, the evaluation of training a 10-class classifier showed in Figure 5a. The final test error rate with best model (lowest error on the validation data) is **72,9%** which is better than a chance but not great. The result of classification on frame-by-frame basis can be further evaluated with a confusion matrix, see Figure 5b, here we see that a lot of genres get many misclassifications, thus increasing final error rate. Next experiment tests dataset of 3 genres (classical, metal and blues) keeping parameters of model the same as first experiment. According to Figure 4 boundaries of the genres are more distinct comparing to 10 genres dataset. Therefore, classification of this 3 genres gives lower error rate, it is **25,8%**, also see Figure 6a. Finally, even better result is achieved when we classify two very different genres - metal and classic - with linearly separable boundaries (Figure 3). Error rate in this case only **4.7%**, see Figure 6b.



(a) Error rate for 10 genres

(b) Confusion matrix for classification on frame-by-frame basis for all genres

Figure 5: Evaluation of 10 genres classifier



(a) Error rate for 3 genres

(b) Error rate for 2 genres

Figure 6: Evaluation of 3 and 2 genres classifiers

5 Discussion & Conclusion

In this project we examined the performance of DNN on music genre classification with different number of genres in datasets. Results obtained under experiments, presented in Table 1, proves that classification of 10 different genres is a challenging task. The correlation between complexity of the classification task (amount genres) and classification error rate is clear, the error rate decreases dramatically if number of genres decreases. The results question the chose of a simple DNN as a model for the classification of music genres, taking a closer look on t-SNE visualization of the 2 genres, that can be found in Figure 3, shows that the dataset consisting only of 2 genres can be easily separated using for example a support vector machine. Furthermore using MFCC extraction with only 13 coefficients potentially could decrease the performance since important information for music (for example fundamental frequency) of the original audio is discarded. The results obtained with our DNN can not be called outstanding compared to, for example, a CNN model for this task [7], where error rate for the classification of 10 different genres was 16%. The reason of such result depends on many factors, first key factor is a model choice and its architecture. The baseline architecture that was used during the experiments was designed from scratch and not evaluated on a task of music genre classification, i.e. number of hidden layers, number of nodes in each hidden layer and activation function were chosen without a search for parameters that can achieve best results on the validation

Table 1: Classification error rate on the test data for different genre classification tasks

# Genres	Error rate
10	72,9%
3	25,8%
2	4.7%

set. Another important implementation detail is hyper-parameters that were used in the architecture. By performing a random search, using accuracy on the validation set as a benchmark, values for hyper-parameters (for example the amount of regularization - λ) can be optimized in order to boost performance on the test set [1].

Another possible bottleneck in used architecture could be a problem of overfitting. It can be noticed that on the error rate graph 5a for classification of 10 genres, the error rate for the validation data does not decrease across all the 30 epochs of training, which can be potentially an overfit to the train data.

A potential performance boost could be achieved by defining an ensemble of simple DNNs, which would classify genre of different part of the song (MFCC features) and taking a majority vote to assign a genre to the whole song.

References

[1] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13:281–305, February 2012.

[2] Keunwoo Choi, György Fazekas, Mark B. Sandler, and Kyunghyun Cho. Transfer learning for music classification and regression tasks. *CoRR*, abs/1703.09179, 2017.

[3] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366, Aug 1980.

[4] Tao Feng. Deep learning for music genre classification. Available at https://courses.engr.illinois.edu/ece544na/fa2014/Tao_Feng.pdf. Last access: 14 May 2017.

[5] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[7] Tom Lh. Li, Antoni B. Chan, and Andy Hw. Chun. Automatic musical pattern feature extraction using convolutional neural network. In *In Proc. IMECS*, 2010.

[8] George Tzanetakis. Gtzan genre collection. Available at http://marsyasweb.appspot.com/download/data_sets/. Last access: 07 May 2017.

[9] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. In *IEEE Transactions on Speech and Audio Processing*, pages 293–302, 2002.

[10] Laurens van der Maaten. Accelerating t-sne using tree-based algorithms. *Journal of Machine Learning Research*, 15:3221–3245, 2014.