Music Genre Classification Using Feed-Forward Neural Networks

1. Introduction

I am very passionate about music and enjoy composing in my spare time. The world has an abundance of music genres, including classical, jazz, and pop, which I am very interested in exploring in my own compositions. Music genres tend to have properties that are unique to them, such as instrumentation, musical form, as well as rhythmic, harmonic, and melodic elements. However, compositions can have elements from numerous genres and placing a composition in a genre can be subjective, which creates a lot of uncertainty when trying to recognise the genre of a composition.

Being able to classify music into different genres would allow musicians to analyse key elements of the set of compositions from specific genre, without having to figure out and recognise the genre myself. This would help in composing pieces from a great variety of genres. Other more common applications of genre classification include music recommender systems, playlist generation and music searching. In this project I decided explore music genre classification problem using feed-forward neural networks.

The project report is organized as follows. In Section 2, I define the Machine Learning (ML) problem based on the application above and discuss the data points, features, and labels in the problem. Section 3 discusses the data set, models and loss functions used, and analyses the compatibility of the ML algorithms with the problem. In Sections 4 and 5, I analyse the results and summarise the main findings of the project, in addition to discussing potential improvements and expansions for future work.

2. Problem Formulation

This task can be modelled as a machine learning problem. The data points represent audio files of songs/compositions.

Sound/music will not be directly used as the features. Instead, I use extracted features from the audio signal data, building on work from the field of Music Information Retrieval (MIR) [1]. The features used in this ML problem are:

- Mel-frequency cepstral coefficients (MFCC),
- Zero-crossing rate (ZCR),
- Spectral centroid and roll-off, and
- Chroma features,

all of which are widely used features in e.g. speech recognition, audio similarity measurement and classification [2, Sec. 2.1].

The quantity of interest/label of the data points is the music genre of the pieces, which is a part of the audio file metadata. There are 16 genres: rock, experimental, electronic, hip-hop, folk, pop, instrumental, international, classical, jazz, old-time/historic, spoken, country, soul, blues, and easy-listening.

3. Methods

For this project, the data points with known labels (genres) were gathered using the Free Music Archive (FMA) data sets provided by Michaël Defferrard et al. and the Swiss Data Science Center. The data sets in FMA, comprised of 106574 licensed tracks in 161 genres, were specifically created for Music Information Retrieval and Music Analysis. There are five data sets in total, four of which are audio file data sets of various sizes and one set focused on metadata [3].

The data set used in this project is the csv-formatted "fma_metadata", which contains all metadata, including genre, of all individual audio tracks in FMA. Conveniently, it also contains the most common MIR features discussed in Section 2, all pre-processed and extracted from the audio data of all tracks using librosa [11], which will be used to train the candidate models. For this project, I will be using 25000 tracks across 16 genres mentioned in the previous section. The data set is split into two subsets. The first set contains 21250 (85%) data points for training and validation, and the second is the test set containing 3750 (15%) data points. The data set is unbalanced, in the sense that some genres may have more samples than others, which will be taken into consideration when analysing test results [4].

I will test and compare various feed-forward neural network architectures/models for the ML problem. K-fold cross validation was used, with k=6 [5, Ch. 6.2.2 & 6.3] on the first set to train all models and find the model with the best results on the validation set. Thus, each fold is around 14.2% of the whole data set, and the size of the training set in each fold is 70.8% of the whole data set. Then, The performance of the resulting hypothesis was assessed on the second set (test set). The original data set is thus split with a ratio of approx. 70%/15%/15%.

Before deciding on the FNN model/architecture to use, I first did some manual prune-testing on what model architectures could be most suitable for this ML problem. The models tested generally achieved an accuracy of 60-65%.

Based on the initial prune-testing (eliminating obviously weaker models), models reached their lowest validation loss at around 6-10 epochs and highest validation accuracy at 7-12 epochs. Overfitting was the most glaring issue with the ML problem, starting usually after 10-15 epochs (see figures 1 and 2 for examples). Having 1-4 (0-3 hidden and output layer) layers for the model gave the best performance without much difference between them, even when changing nodes in each layer. Having at least 5 hidden layers caused worse performance, overfitting very quickly already at higher validation errors. In addition, using an appropriate regularization layer benefited the architecture by reducing the effects of overfitting (although not entirely), increasing the accuracy by 1-2%. Also, using all the features listed in Section 2 improved accuracy by up to 1%, compared to only MFCC.

Finally, I compared two fully connected neural network models containing 2 and 3 layers using ReLU activation function and categorical cross-entropy as the loss function. The input data contained all features listed in Section 2. The input layer (1028 nodes) is the same for both models. The 2-layer model has one hidden layer (128 nodes) with regularization applied to it. The 3-layer model has two hidden layers, containing 256 and 64 nodes respectively, with regularization was applied to the second hidden layer. ReLU was used as the activation function at these layers for faster and better performance [7]. I chose to use dropout regularization, a technique in neural network models that randomly sets input features to 0 at a specified rate, reducing the effects of overfitting [10]. The output layer is also the same for both models, using softmax activation to construct a probability distribution of the genres (labels) for given outputs. Categorical cross-entropy is used as the loss function in conjunction with softmax to reduce complexity of calculating the derivative of the loss [6]. The model was implemented using scikit-learn, Tensorflow and Keras [8][9][12].

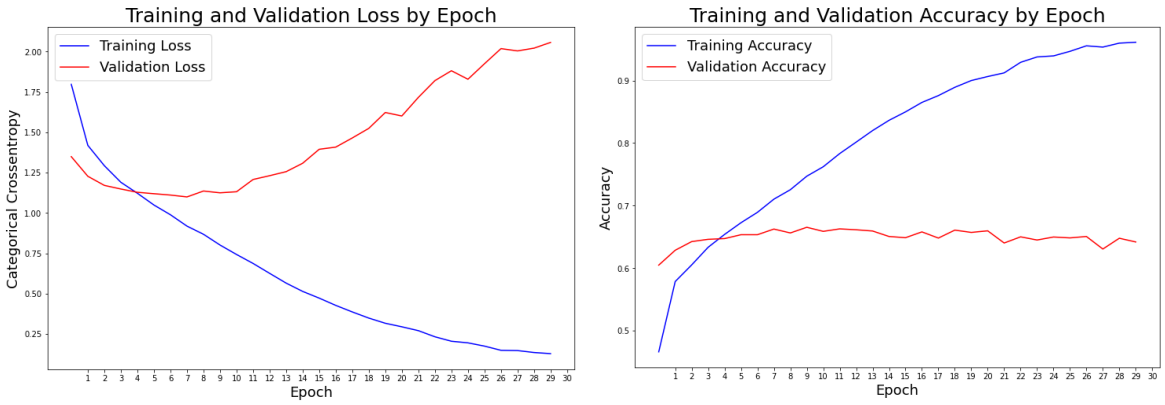

Figure 1 and 2: Feed-Forward Neural Networks, using 6-fold cross validation. The image on the left/right represents the typical training and validation loss/accuracy by epoch. Both training loss and accuracy improved over each epoch. However, after around 7 epochs, the validation loss starts to go up very quickly, indicating the model's susceptibility to overfitting. The validation accuracy starts to slightly drop off after 9 epochs. The graphs are from part of the cross-validation process for one of the tested models.

4. Results

Below is a table of the resulting training and validation errors of FNN models used in the k-fold cross validation after the initial pruning phase, for different choices of dropout rate, epochs, and number of layers:

Dropout	Epochs	Layers	Train loss	Train acc.	Val loss	Val acc.
0.3	7	2/3	0.851/0.845	0.725/0.731	1.109/1.143	0.653/0.649
0.3	10	2/3	0.687/0.632	0.776/0.796	1.126/1.190	0.655/0.648
0.3	14	2/3	0.494/0.390	0.839/0.875	1.194/1.391	0.652/0.641
0.4	7	2/3	0.920/0.919	0.704/0.709	1.107/1.138	0.651/0.650
0.4	10	2/3	0.769/0.711	0.750/0.772	1.112/1.184	0.655/0.647
0.4	14	2/3	0.585/0.467	0.808/0.849	1.161/1.343	0.653/0.645
0.5	7	2/3	0.979/0.996	0.687/0.691	1.108/1.139	0.649/0.648
0.5	10	2/3	0.853/0.818	0.722/0.743	1.103/1.178	0.656/0.646
0.5	14	2/3	0.687/0.588	0.774/0.812	1.139/1.301	0.658/0.648
Prune-testing findings						
Epochs over (>)30			<0.015	>0.999	>1.955	<0.636
Layers >5			<0.759	>0.773	>1.372	<0.631
MFCC only			<0.890	>0.717	>1.161	<0.642
L > 5, E > 30, MFCC			<0.179	>0.949	>2.658	<0.612

Table 1: Training and validation errors and accuracies for different choices of dropout rate, epochs, and number of layers. The model with smallest validation error is highlighted in red, and highest validation accuracy highlighted in blue. The prune-testing findings section shows training and validation errors for models with otherwise similar parameters as the selected models described in Section 3.1, but with one nonoptimal parameter. E.g. The row "Epochs >30" represents the error and accuracy of a model with all other parameter choices like those of the selected models, except for the number of epochs.

According to Table 1, using more epochs improves the training loss and accuracy, but negatively impacts validation loss and accuracy. The effects of higher epochs are of greater magnitude, the higher the number of layers is. On the other hand, a higher dropout worsens the training loss and accuracy and improves validation loss and accuracy. Both match my observations in the prune-testing phase. After 7-10 epochs the validation loss started to increase, whereas the training loss continued to decrease (see Figure 1), which indicates overfitting.

Overall, the models with 2 layers outperformed ones with 3 layers. The model with the smallest validation error has a dropout rate of 0.5 and ran for 10 epochs, and the one with the highest validation accuracy has a dropout rate of 0.5 and ran for 14 epochs. Thus, I assess the performance of the resulting hypothesis (layers = 2, dropout = 0.5, epochs = 10) by computing its average loss on the test set with 3750 data points. The resulting test error and accuracy were 1.090 and 0.660, respectively, which is an improvement on the validation error and accuracy.

5. Conclusions

I have studied 18 different models for predicting the music genre based on MFCC, ZCR, spectral centroid and roll-off and chroma features. The models are constructed as feed-forward neural networks with different choices for number of layers, dropout rate and number of epochs. I used k-fold cross validation to choose the best model (layers = 2, dropout = 0.5, epochs = 10), which resulted in the smallest validation error 1.103 with an accuracy of 0.656. The corresponding training error was 0.853, with an accuracy of 0.722, which are better than the corresponding validation scores. The use of validation sets was crucial to the training process, as the models were very susceptible to overfitting. I did not find a benchmark level that would allow to tell if an error on the level of 1.103 is acceptable.

The final hypothesis/best model has been tested on the test set with 3750 datapoints. The test set is gathered from the same original data set as the training and validation set. The test error and accuracy were 1.090 and 0.660, respectively, a slight improvement from the corresponding validation scores.

The project has a couple of avenues for future work or experimentation. One obvious expansion is increasing the data set and adding compositions from an even wider array of genres. Since the original dataset was unbalanced, it should be a priority to find more samples from genres that had less data to work with. Overfitting turned out to be a very central issue in the ML problem, so I should consider exploring the use of other regularization methods, and perhaps even further reducing the complexity of the network while avoiding oversimplification if possible, for example eliminating unnecessary features with principal component analysis [10]. Since the test scores were better than the validation scores, it is possible that the test set happened to be a better representation of the training data than validation, which may imply a problem with the way the data was split. Hence it is a good idea to try repeating the experiment on randomly shuffled data and performing the splits again, to make sure that the higher test scores were not due to lucky splits. One possible approach is to use stratified k-fold cross validation [15].

Other future endeavours include using Mel spectrograms as features for using convolutional neural networks in this task. Mel spectrograms are visual representations, images, of mel-frequency cepstrums, which represent the short-term power spectrums of sounds [14]. Essentially, the problem can be transformed into an image classification task, to which convolutional neural networks are particularly suited to [13]. Another experiment could be applying clustering methods to the ML problem to find certain patterns within the music, which could directly deduce what are the key elements within a specific genre.

6. References

[1] L. Smith, O. Campbell, M. Goldstein, Sudara, “Music Information Retrieval”, *musicinformationretrieval.com*, 2021. [Online]. Available: https://musicinformationretrieval.com/. [Accessed: 10-Mar-2021]

[2] J. Breebart and M. McKinney, "(PDF) Features for Audio Classification", *ResearchGate*, 2004. [Online]. Available: https://www.researchgate.net/publication/250008991_Features_for_Audio_Classification. [Accessed: 10-Mar-2021]

[3] M. Defferrard, K. Benzi, P. Vandergheynst and X. Bresson, "FMA: A Dataset For Music Analysis", *arXiv.org*, 2017. [Online]. Available: https://arxiv.org/abs/1612.01840v2. [Accessed: 11-Mar-2021]

[4] Michaël Defferrard, 2017. ”FMA: A Dataset For Music Analysis”, https://github.com/mdeff/fma. [Accessed: 11-Mar-2021]

[5] A. Jung, “Machine Learning. The Basics”, 2021. [Online]. Available: https://github.com/alexjungaalto/MachineLearningTheBasics/blob/master/MLBasicsBook.pdf. [Accessed: 23-Mar-2021]

[6] Paolo Perrotta, 2020. “Killer Combo: Softmax and Cross Entropy”, *GitConnected*. [Online]. Available: https://levelup.gitconnected.com/killer-combo-softmax-and-cross-entropy-5907442f60ba. [Accessed: 23-Mar-2021]

[7] "Rectifier", *En.wikipedia.org*, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Rectifier_(neural_networks). [Accessed: 23-Mar-2021]

[8] F. Pedregosa, et.al, “Scikit-learn: Machine Learning in Python“, *Journal of Machine Learning Research*, vol. 12, p. 2825--2830, 2011. [Accessed: 16-Mar-2021]

[9] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. [Accessed: 16-Mar-2021]

[10] Jason Brownlee, 2018. “How to Avoid Overfitting in Deep Learning Neural Networks”, *Machine Learning Mastery*. [Online]. Available: https://machinelearningmastery.com/introduction-to-regularization-to-reduce-overfitting-and-improve-generalization-error/. [Accessed: 23-Mar-2021]

[11] McFee, Brian, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. “librosa: Audio and music signal analysis in python.” In Proceedings of the 14th python in science conference, pp. 18-25. 2015.

[12] François Chollet et al. Keras. https://keras.io, 2015 [Accessed: 16-Mar-2021]

[13] "Convolutional neural network", *En.wikipedia.org*, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Convolutional_neural_network. [Accessed: 24-Mar-2021]

[14] Leland Roberts, 2020. “Understanding the Mel Spectrogram”, *Medium.com.* [Online]. Available: https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53. [Accessed: 24-Mar-2021]

[15] ”Would it be possible that test result is better than validation result?”, *stackoverflow.com*. [Online]. Available: https://stackoverflow.com/questions/49205773/would-it-be-possible-that-test-result-is-better-than-validation-result. [Accessed: 28-03-2021]