Using Neural Networks to Apply Intelligent Genre-Specific Equalization to Audio Masters

Bryce DuBray & Farzad Mohammadi

Overview

Recent trends in audio technology are pushing toward automating the mastering process, and as more artists are able to independently record music, the more appealing a streamlined, inexpensive mastering solution becomes. As audio engineers, we are direct witnesses of this trend affecting the independent music community and professional recording industry of Nashville alike. Skeptics question whether an algorithm can outperform a mastering engineer, but we believe using automation to at least augment mastering can importantly lower the barrier of entry to releasing professional-sounding music.[1]

Mastering is an umbrella term for a number of processes, but for our purposes it is best to focus on the process of equalization, since it is applied to nearly every modern master and has unique associations to genre (i.e. hip hop and EDM having more intense bass frequencies than country music). We plan to use a supervised parametric neural network that uses the relative amplitude of 5 bands of frequencies (low, low mid, mid, high mid, and high) of an audio file as inputs to predict the likelihood of that music being a particular genre (pop/rock, classical, hip hop, jazz, disco, and country). With this identification, another program could then adjust the relative amplitude of each frequency band to match a preset frequency response customized for the genre.

More frequency bands and more genres could be added to make this network more specific and accurate. However, this could also contribute to overfitting, as frequency response tends to increasingly overlap as intermediary genres are introduced (i.e. blues would likely overlap with rock and country). For now, using 6 mostly distinct genres will allow the network to train most accurately since there will be the most correlation between input to genre.

[1] Thomas Birtchnell. 2018. Listening Without Ears: Artificial Intelligence in Audio Mastering. *ACM* 50, 1, Article 2 (Nov. 2018), 16 pages. DOI: https://journals.sagepub.com/doi/pdf/10.1177/2053951718808553

Background

The term, “artificial intelligence”, was coined by John McCarthy who is a pioneer in the field. He defines artificial intelligence as the science and engineering of making intelligent machines, especially intelligent computer programs. The history of AI can be followed back to the hardware advances in both processors and memory such as those in today’s computer systems. The importance and uniqueness of artificial intelligence is best elaborated through the works of Alan Turing who created the “Turing Test” in 1950 which implies the possibility that a computer can behave intelligently.[2] Artificial Intelligence continued to be studied from these two great men and many more expanding into many other fields.

Audio Mastering has been automated for many years as a way to simplify alterations to audio and improve fidelity without human intervention. It has incorporated a number of different techniques, algorithms, and features. This includes the yule-walker approach, machine learning, and breaking down an audio’s spectral, dynamic, and temporal features.[3][4] LANDR, an automated mastering service, has been an important figure in audio engineering since the late 20th century working to provide audio engineers with the ability to automate their work.[5] They have been one of the most prominent figures in the audio engineering field.

In terms of this project, we have combined basic digital signal processing of audio with machine learning to create the basis for an Intelligent Equalizer (IntellEQ). We have constructed our coding project using Python as a neural net training tool and Matlab as a signal processor to analyze the frequency content of a piece of audio and map it to a particular genre. Additional genre-specific algorithms could then be applied to match the audio master to the sonic trends of its identified genre. LANDR, Smart EQ by Sonible, Izotope, and others are applications we have worked as audio engineers and are using their concepts as a basis to design this automated

[2] Chen Manni. 2020. *Intelligent Audio Mastering*. Master of Arts Thesis. School of Creative Media, University of Hongkong, Hong Kong, China. DOI: https://www.researchgate.net/publication/344127327_Intelligent_audio_mastering

[3] Zheng Ma, Joshua D. Reiss, and Dawn A.A. Black. 2013. In *Implementation of an intelligent equalization tool using Yule-Walker for music mixing and mastering* (AES 134th Convention), May 4 - 7, 2013, Rome, Italy. Audio Engineering Society, New York, NY, 1-10. DOI: https://www.academia.edu/31481046/Implementation_of_an_intelligent_equalization_tool_using_Yule_Walker_for_music_mixing_and_mastering?email_work_card=view-paper

[4] Jay Lebeouf and Stephen Pope. 2015. Automatic Learning and Control of Audio Algorithms By Audio Recognition. (May 2015). Patent No. US 9,031,243 B2, Filed Sept. 28, 2010, Issued May 12, 2015.

[5] Jonathan Sterne and Elena Razlogova. 2019. Machine Learning in Context, or Learning from LANDR: Artificial Intelligence and the Platformization of Music Mastering. *ACM* 51, 2, Article 2 (June 2019), 18 pages. DOI: https://journals.sagepub.com/doi/pdf/10.1177/2056305119847525

audio mastering application. Constructing our own automated audio application is a unique opportunity we have as audio engineering and computer science students, and is a good step in following this unique trend in the music industry.

Process

There are two major types of processing done to a piece of audio in mastering: dynamic and spectral. Dynamic processing is used to compress the maximum and minimum signal level of a piece of audio to fit its playback medium, so although some genres use significantly more compression than others, the dynamic range can often be more of an indicator of its release format than a definitive genre. An audio file's frequency response, however, can vary more consistently from genre to genre. Therefore, spectral processing, or equalization, is the more useful aspect of audio mastering that we can use to identify genre.

To make our dataset, we used the GTZAN genre dataset, a collection of 10 genres with 100 30-second song samples each.[6] The genres included are blues, classical, country, disco hip hop, jazz, metal, pop, reggae, and rock. We then created a MATLAB script to filter each song sample into five audio bands: low (20-250 Hz), low-mid (250-500 Hz), mid (500-2000 Hz), high-mid (2000-6000 Hz), and high (6000-20000 Hz). An RMS amplitude was calculated for each band along with the RMS amplitude of the original full-spectrum track, resulting in an input array of length 6 for our training data; the 5 RMS band measurements provide a rough analysis of frequency response, and the overall RMS measurement acts as somewhat of a bias for the data. Each associated genre was translated to an output array that could be used with a softmax activation function: [1,0,0,0,0,0] for rock, pop, blues, reggae, and metal, [0,1,0,0,0,0] for classical, [0,0,1,0,0,0] for country, etc. Rock, pop, blues, reggae, and metal were consolidated into one output category because of the often overlapping frequency response between genres due to similar recording techniques. This decision also helped provide more correlation for the neural net to recognize.

We built the neural network within PyCharm using the Tennis framework. The best training and testing results came from using a hidden layer of size 12 created with a tanh activation function, and using a softmax activation function on the output layer.

We used MATLAB to create the final app that allowed for arbitrary instances. The neural network was recreated in MATLAB functions, and the final optimal layer weights found during training in PyCharm were hard-coded into the function so the network would only need to

[6] Carlos N. Silla, Alessandro L. Koerich, and Celso A.A Kaestner. 2008. A Machine Learning Approach to Automatic Music Genre Classification. *Journal of the Brazilian Computer Society* 14, 7-18 (September 2008), 12 pages. DOI: https://doi.org/10.1007/BF03192561

forward-propagate. The app allows the user to select any .wav or .mp3 file from their machine as an input, then does the multiband processing on it to create the neural net input array. When the user presses “Run”, the neural net code executes and the input file is mapped to a genre that is then displayed.

Results

This network certainly proved difficult to train despite several approaches to improving its learning. Our best results produced roughly 27% error of 6000 training examples. Considering there are 6 total genre output mappings, randomly guessing one each iteration would potentially bring more accurate results, unfortunately. Obviously, this is not an acceptable result for an algorithm that would simulate the discernable ear of a skilled mastering engineer if it cannot even identify the genre of a quarter of the songs it is given. Adding a hidden layer was by far the greatest improvement on the network, and any following attempts at better learning were met with marginal, if any, improvements.

Adding multiple hidden layers did not improve the error rate, although this was to be expected since the effectiveness of additional hidden layers often drops drastically after the first. Implementing a dropout mask improved the error rate of the internal neuron mappings significantly, but the rounding of the softmax output activation function made these improvements insignificant as the overall accuracy remained the same at around 30%.

On the other hand, we did manage to successfully recreate the simple three-layer network in a standalone app using MATLAB. This allows us to use any arbitrary audio file to run through the algorithm, and can be used as a decent demonstration tool for the intended purpose of the application.

Conclusion & Future Work

As stated before, there were several techniques we used to try to improve the accuracy of our network. The first was increasing the number of hidden layers, although the effectiveness of hidden layers decreases dramatically after the addition of the first. Therefore, adding a second or third hidden layer did not improve performance by a large enough margin. The second technique we tried was adding dropouts to combat overfitting, but the dropout mask did not improve overall accuracy in the final output layer. A convolution layer was decided to be unnecessary for this network, since the amount of input neurons was so small.

In any neural network, overfitting will likely be a challenging obstacle that needs to be overcome. However, because additional hidden layers and dropout masks did not improve our network's learning, it is actually likely that the network *underfit* the data instead. It is highly likely that there just may not be enough correlation between frequency response and genre in our dataset to produce any meaningful results. After all, we are only using 5 frequency bands, and the frequency response of an audio file is impacted by much more than just the genre of music; where and how it was recorded, how it was mixed, and other factors can influence spectral content tremendously as well.

In hindsight, the best approach to improving this network would be to change the input data. Obviously, the more data the better, so collecting even more assorted music by genre could help. An even better solution could be to increase the number of frequency bands used in analysis, increasing the size of the input layer. With enough bands, a convolution layer could actually be implemented to specifically look for, say, a loud bass range and map that to an appropriate genre. Too many bands could lead to a problem, however; using too narrow of a filter could actually increase the correlative overlap from genre to genre. This is because a wide frequency range (i.e. low band or mid band) provides a more general look at overall musical content that could be tied to a genre, like hyped low end in hip hop or crisp high end in a pop song. Creating too narrow of a band could potentially only show frequency content in a song that would be tied to specific notes or its key instead, and that would be inconsistent with genre.

In conclusion, this project was a great opportunity to dive deeper into the emerging trends in the audio engineering field. More importantly, despite dissatisfying results, this project was also a good lesson in arguably the most crucial aspect of constructing a network: the dataset. Without proper architecture and loads of example data, the network simply won't find correlation enough to learn. However, we still consider IntellEQ an overall success in integrating machine learning concepts with digital signal processing.

Annotated Bibliography

[1]Carlos N. Silla, Alessandro L. Koerich, and Celso A.A Kaestner. 2008. A Machine Learning Approach to Automatic Music Genre Classification. *Journal of the Brazilian Computer Society* 14, 7-18 (September 2008), 12 pages. DOI: https://doi.org/10.1007/BF03192561

"A Machine Learning Approach to Automatic Music Genre Classification" follows an approach to classifying music genres through its spectral and temporal properties. The classification is completed through binary classifiers whose results are merged to produce a musical genre label. The experiment is conducted with a GTZAN Database containing 1,000 samples in wav and mp3 files in 10 different genres. This source goes into further detail in its method to automatically extract data from audio files and classify into different genres which will help us do the same for our project.

[2]Chen Manni. 2020. *Intelligent Audio Mastering*. Master of Arts Thesis. School of Creative Media, University of Hongkong, Hong Kong, China. DOI: https://www.researchgate.net/publication/344127327_Intelligent_audio_mastering

In "Intelligent Audio Mastering", Chen Manni analyzes intelligent audio mastering through three perspectives: AI technology, sound perception, and mastering in application. This source breaks down the importance of artificial intelligence then provides examples of intelligent audio mastering through Alan Turing, LANDR, DAWs, and Thomas Birtchnell. The author explains how automation imitates human operation of comparison and sound processing without human intervention. This source is useful as it supports and elaborates on the importance of intelligent machine learning for the audio field. It provides a number of different examples and resources to assist in the completion of our final project.

[3] Jay Lebeouf and Stephen Pope. 2015. Automatic Learning and Control of Audio Algorithms By Audio Recognition. (May 2015). Patent No. US 9,031,243 B2, Filed Sept. 28, 2010, Issued May 12, 2015.

This patent describes a sophisticated multi-stage audio analysis process that is used to attach high-level metadata to audio files. By doing so, audio can be intuitively and automatically categorized for use in further processing. The patent is assigned to the audio software company Izotope, Inc., which is well known for its intelligent mastering assistant plug-in suite Ozone.

This patent discloses useful methods for automatically categorizing audio files, for our case into genres, that can then be used to apply the appropriate processing.

[4] Jonathan Sterne and Elena Razlogova. 2019. Machine Learning in Context, or Learning from LANDR: Artificial Intelligence and the Platformization of Music Mastering. *ACM* 51, 2, Article 2 (June 2019), 18 pages. DOI: https://journals.sagepub.com/doi/pdf/10.1177/2056305119847525

“Machine Learning in Context” deconstructs the intelligent mastering service LANDR. The authors question the validity of the company’s claims that they fully automate the mastering process, suggesting that machine learning is only used in part of the process and conventional signal processing is truly the core of their service. Nevertheless, despite the ethical controversy of falsely claiming “AI” for its appeal as an industry buzzword, the article reveals useful methods for using machine learning to classify audio files. They claim LANDR simply uses machine learning to analyze sound and map it to a matrix of preset processing algorithms, instead of the full automation it claims. Even so, this methodology is useful for our research, because *complete* automation of mastering is not necessary for utility, and this partial automation is sufficient and achievable for our project.

[5] Thomas Birtchnell. 2018. Listening Without Ears: Artificial Intelligence in Audio Mastering. *ACM* 50, 1, Article 2 (Nov. 2018), 16 pages. DOI: https://journals.sagepub.com/doi/pdf/10.1177/2053951718808553

In “Listening without ears,” Birtchnell explores the potential role of automation in the audio mastering field. In interviews with professional mastering engineers, Birtchnell asks how they perceive AI in their field, whether it is viewed as a tool or a threat. From this data and his knowledge of the history of mastering, he predicts several different hybrid scenarios in which AI could affect the mastering “workflow.” These scenarios span from using AI as a reference tool, to humans acting as a “premium” upgrade from AI mastering, and AI simply being used as a data management tool outside of the creative part of mastering. This source is useful to our research in that it validates the importance of using machine learning to automate parts of the mastering process and combats skepticism by viewing its role through several lenses as an augmentation, not replacement, of human mastering.

[6]Zheng Ma, Joshua D. Reiss, and Dawn A.A. Black. 2013. In *Implementation of an intelligent equalization tool using Yule-Walker for music mixing and mastering* (AES 134th Convention), May 4 - 7, 2013, Rome, Italy. Audio Engineering Society, New York, NY, 1-10. DOI: https://www.academia.edu/31481046/Implementation_of_an_intelligent_equalization_tool_using_Yule_Walker_for_music_mixing_and_mastering?email_work_card=view-paper

“Implementation of an intelligent equalization tool” is a conventional paper for the AES 134th Convention where three individuals utilize a new approach for automatically equalizing an audio signal towards a target frequency spectrum. It is an algorithm based on the Yule-Walker method and designs a digital filter through the spectrum of a large dataset of commercial recordings.The coding process is implemented through C++ and MatLab. The Yule-Walker method fits an autoregressive model to the windowed input by minimizing the forward prediction error in the least squares sense. The algorithm and method construct a filter curve so variable smoothing is completed on desired filter magnitude responses. This source is useful as it assists in the automatization and alteration of an audio signal’s frequency spectrum by following an approach similar to ours.