2021-07-07

Indexing Mood

David Kleingeld

Email
dskleingeld@gmail.com

Contents

1 Introduction 1
2 Implementation 3
3 Results 4
4 Conclusion 6
A Run Instructions 8

1 Introduction

Music can evoke strong emotions when listened too. Many people like to select music based on their current mood. The field of music emotion recognition (MER) tries to find a way to predict the emotion a musical piece will evoke. Building on existing methods I present and evaluate a system to index the emotion or mood of a musical piece.

A musical piece, being an audio signal, is stored as a list of intensities for each timestamp. We can transform this into different representations such as intensity for various frequencies. We can reduce these representations into something 'simpler', a sound feature, such as *the average pitch* of a song. These features can tell us things about a song. Intuitively a song with a high *pitch* and *high beats per minute (bpm)* sounds more anxious then one with low *pitch*

and *bpm*. Two or three features are not enough, we need some nuance, for example Vivaldi's Summer contains quite some high bpm high pitch fragments however those sound exited rather then anxious. To get the needed nuance I use 10 different sound features. Instead of constructing rules based on intuition a neural network fits the features to emotions. To train it we use samples from the *1000 Songs for Emotional Analysis of Music*[5] dataset.

This dataset contains annotations in the valence arousal model. In this model every emotion is seen as a combination of *Valence*, positive versus negative emotion and *Arousal* the emotions intensity. It is often represented as a continues 2 dimensional space[1] as seen in fig. 1. The dataset motivates this model as it allowed rating songs in reference to other songs.

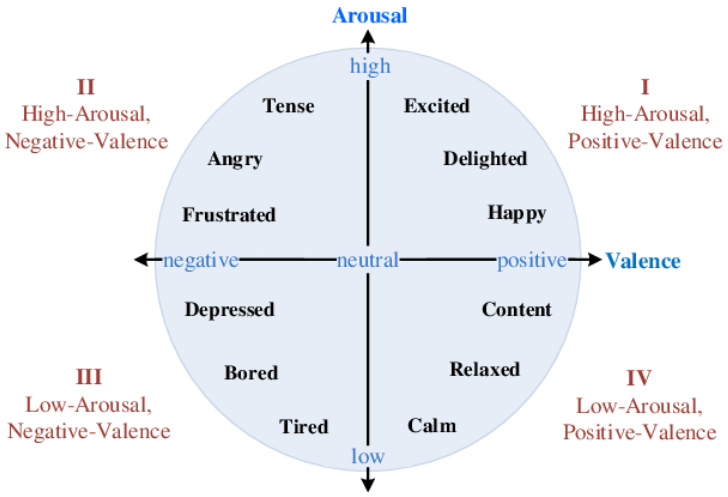

Figure 1: *A visual mapping between discreet emotions and valence-arousal space*

In the next section I will explain my implementation; introduce the used audio features, neural network architecture and going through the audio pipeline. Then I will go over the results, that is: the neural networks accuracy and the my own personal judgment listening to samples. Finally I will conclude whether this effort was successful and discuss what improvements could be made in future work.

[1] In reality this is more of a 1-d space as only the angle between valence and arousal seems to matters

2 Implementation

Audio features are the cornerstone of this implementation, I base my choice of features on the findings of Shlok Gilda et al[3]. They use recursive feature elimination to reduce a large amount of candidates based on established emotion recognition work to 11 features. From these I pick exclude one feature that was hard to extract given available libraries. These used features are:

1. Pitch
2. Spectral Rolloff
3. Mel Frequency Coefficents
4. Tempo
5. Root Mean Square Energy
6. Spectral Centroid
7. Beat Spectrum
8. Zero Crossing Rate
9. Short FFT
10. Kurtosis

Many of these features are sufficiently complicated that to explain them would need a report on its own. I will focus on how these are used to get to a single feature vector for each song. The feature extraction relies on the *librosa* Python library [4]. Excluding tempo all feature are either continues or return a sequence. I simply take the average over a 45s sample of audio.

The feature vector, a list of numbers, is fed into the neural network. It maps the features to a valance and an arousal. These are then mapped to an emotion following the model in fig. 1. After testing multiple models I settled on a fully connected network with 3 hidden layers, see fig. 2. All layers except the output use *relu* as activation. The output uses linear activation[2]. The loss function is the Mean Absolute Error as valance and arousal are two distinct dimensions. To find the best model the candidates where trained for 100 epochs using 20% of the data for validation. I measure for how many epochs the network can train before it stats to overfit then I train the final model on all the data for that many epochs.

The feature extraction works on the level of raw audio samples, it can not handle stereo or compensate for different sampling rates. To compensate each song is converted to mono and re-sampled to 44.1kHz. Each song is also reduced to a 45s segment starting 10 seconds from its beginning.

[2]because the valence-arousal model does not restrict values between zero and one and we do not want to bias to zero and one

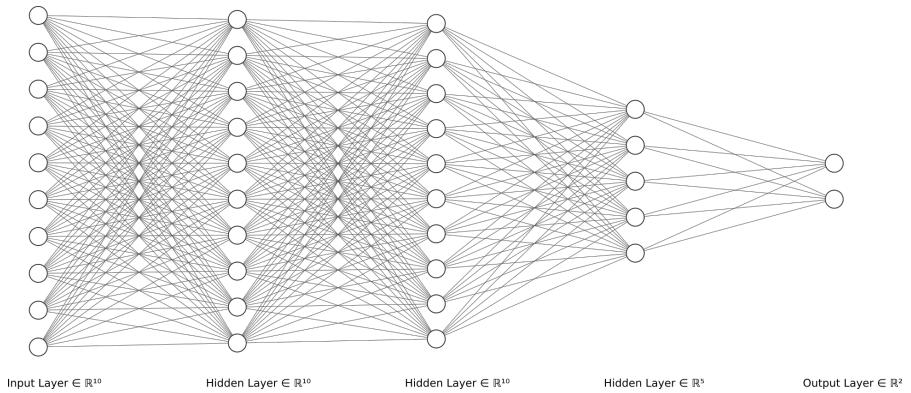

Figure 2: *Neural network architecture used to map features to valance arousal space*

3 Results

One way to evaluate the emotion estimation is by looking at the performance of the validation set during model selection. As the validation data is randomly sampled from the dataset it should give a good estimate of eventual performance without suffering of overfitting. We see the loss drop as we train longer in fig. 3. Initially the validation loss is significantly lower then the training loss. This could be interpreted as the validation data being easier to 'classify' then the training data. The chance of this happening on a randomly selected dataset is very small. We see the networks performance improves up to about 50 epochs without any signs of overfitting.

Now we know we can use the loss as a performance measure. The neural network gives us mean absolute error as loss, defined as:

$$\frac{(valence_{true} - valence_{predicted})^2 + (arousal_{true} - arousal_{predicted})^2}{2}$$

This indicates how close the predicted valence and arousal are to the data. They predicted values can be far off however if the ratios are close the detected emotion might still be the same. The loss of the neural network on the complete dataset was 12%, that is the valance and arousal deviated 12% from the dataset annotations on average.

A more robust way to check if the network functions is to spot check samples not in the dataset. For this I will use two audio samples in the public domain:

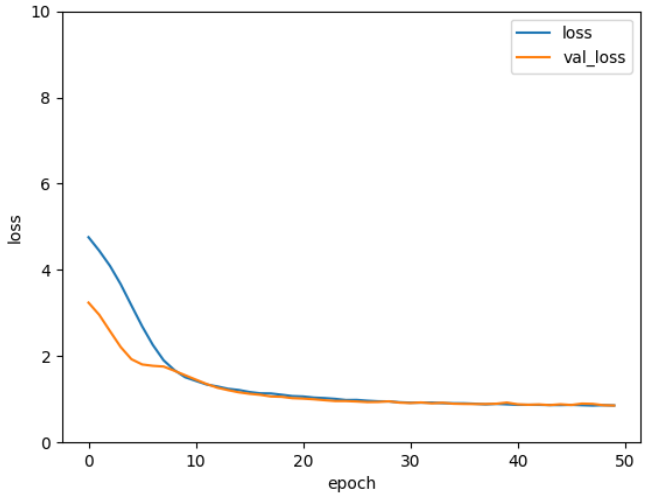

Figure 3: *"Training and Validation loss of the chosen model, notice how the validation data has a lower loss at the start"*

the first movement of Beethovens *Moonlight Sonate*[3] which I would classify as content and Vivaldi's *Storm*[4] which I classify as Tense. I also take the top hit on YouTube for *relaxing music*[5] ($283 \cdot 10^6$ views). The emotion prediction results are in table 1.

We see *Storm* is completely misclassified[6]. We combine the arousal and valence into a weight by adding their absolute values. This might indicate how certain the neural-net is. It is five times higher for the other two songs. Those are also classified better, still misclassified as Tired however Tired and Relaxed are quite close in my opinion.

[3] see: https://youtu.be/4Tr0otuiQuU?t=10

[4] see: https://youtu.be/NqAOGduIFbg?t=10

[5] see: https://youtu.be/1ZYbU82GVz4?t=10

[6] If you switch the values for valance and arousal you get a plausible classification, namely: Frustrated. I carefully checked the code-base for any mistaken switching and found none.

Table 1: *Emotion estimations for hand picked songs*

Song	Estimate	weight	Neural net output	
			arousal	valence
Moonlight Sonate	Tired	0.58	-0.34	-0.23
Storm	Content	0.10	-0.016	0.08
Relaxing music[a]	Tired	0.56	-0.32	-0.24

[a] full name: *Relaxing Sleep Music • Deep Sleeping Music, Relaxing Music, Stress Relief, Meditation Music (Flying)*

4 Conclusion

I build a system that estimates the emotion or mood of a musical piece. It performs well on the relatively small dataset but fails to generalize as seen with a quick test. The dataset only contains 1000 songs, from these I take only a single 45 second fragment. The small dataset is not enough to generalize even without overfitting.

The regression problem given to the neural network can be improved, instead of a separate valance and arousal the angle between them could be the networks prediction. The choice for valance arousal was made because of the availability of annotated data it might not be the right choice for emotion prediction. Professor of psychology Lisa Feldman Barrett[1] notes in her book on emotions:

> *A careful read of the literature reveals that no theory has ever hypothesized that emotions can sufficiently be reduced to or explained by valence and arousal. Instead, these theories hypothesize that valence and arousal are important (and perhaps necessary) descriptive features of all emotions.*

Going from audio features through arousal and valance to get to emotions is throwing away information. A neural network classifying emotions directly instead of regressing to arousal and valance is going to perform better. Further more we should ask if we really want the emotion of a song and not an other description. I rarely find myself looking for music with a specific emotion. Rather I might want *energetic* and *rough* music when feeling angry and *calm smooth* music when tired. These are descriptions that intuitively map to certain audio features.

This seems to be the way the state of the art is developing[7]. New far larger datasets are now available with over 18-thousand songs annotated with 56 moods and themes annotations including descriptions such as *uplifting* and *meditative*[2]. There has been a declining interest in emotion recognition since 2014, the inclusion of theme recognition could very well bring this around while providing more usable results.

[7]Which I only found out while writing this conclusion

A Run Instructions

Install

To install you will need: make, python version 3.7 or higher and a installation of pip for that python. Then run *make install* from the projects root. This assumes these versions of pip and python are called using pip3 and python3. You might need to change this in the makefile and loosen versions in requirements.txt depending on what is available for your installed python. To prevent messing with your current python environment I recommend doing this in a virtual env.

Use

The project download comes with the trained model and annotations database, ready to be used for emotion estimation. To estimate a songs emotion change the variable `path` in `estemate.py` to the location of the song for which you wish to estimate the emotion and call it using *python3 /src/audiovibe/estimate.py*. You may wish to use *test/cut45.sh* to cut songs to a 45 second segment (requires ffmpeg).

References

[1] Lisa Feldman Barrett and Christiana Westlin. 'Chapter 2 - Navigating the science of emotion'. In: *Emotion Measurement (Second Edition)*. Ed. by Herbert L. Meiselman. Second Edition. Woodhead Publishing, 2021, pp. 39–84. ISBN: 978-0-12-821125-0. DOI: https://doi.org/10.1016/B978-0-12-821124-3.00002-8. URL: https://www.sciencedirect.com/science/article/pii/B9780128211243000028.

[2] Dmitry Bogdanov et al. 'MediaEval 2019: Emotion and Theme Recognition in Music Using Jamendo'. 2019. URL: https://multimediaeval.github.io/2019-Emotion-and-Theme-Recognition-in-Music-Task/.

[3] Shlok Gilda et al. 'Smart music player integrating facial emotion recognition and music mood recommendation'. In: *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. 2017, pp. 154–158. DOI: 10.1109/WiSPNET.2017.8299738.

[4] Brian McFee et al. 'librosa: Audio and music signal analysis in python'. In: *Proceedings of the 14th python in science conference*. Vol. 8. 2015.

[5] Mohammad Soleymani et al. '1000 Songs for Emotional Analysis of Music'. In: *Proceedings of the 2nd ACM International Workshop on Crowdsourcing for Multimedia*. CrowdMM '13. Barcelona, Spain: Association for Computing Machinery, 2013, pp. 1–6. ISBN: 9781450323963. DOI: 10.1145/2506364.2506365. URL: https://doi.org/10.1145/2506364.2506365.