Real-Time Music Emotion Recognition

Jan Havránek

Machine Learning course 2020/2021

1 Motivation & goals

The aim of the project was to create an RGB light, which would adjust its color in reaction to the emotions of music played to it. Such task, which may seem to be very abstract in its nature, can be broken down to four tractable sub-tasks: representation of emotions in music, recognition of such emotions from audio, translation of emotions to colors, and preparation and integration of necessary hardware (along with controlling software).

As for representation of emotions, two main approaches are possible: discrete and dimensional. While discrete models use a predefined set of words to describe mood of a song, dimensional models represent emotions as N-dimensional continuous vectors [1]. In the case of this project, the latter approach was used, with two-dimensional valence-arousal (V-A) space, where the two values represent positivity/negativity of the song and its perceived energy, respectively.

For the two last points, I have utilized solutions from my previous hobby project. Therefore, the main remaining (and, by far, the most challenging) part of the project was to create a machine learning model capable of emotion recognition from audio in real time.

2 Dataset

The model was trained and evaluated on *1000 Songs Database* [2], which provides 45 s long music excerpts annotated with V-A values from range [-1, 1] every 0.5 s. Due to the later discovery of duplicates, the dataset actually comprises only 744 songs, split between training (619 songs) and evaluation (125 songs) set. For hyperparameter optimization, I further split the training set into training (495) and validation (124) set using stratified sampling by genre.

3 Feature extraction

Although *1000 Songs Database* provides extracted audio features, I decided to compute the features from scratch, so that I could apply the model on any song later. For this task I chose `pyAudioAnalysis` [3] package, with the use of which I extracted 68 low-level audio features (window size 50 ms, step 50 ms). This resulted in $68 \cdot 10$ values per annotated fragment.

4 Baseline model

As a baseline model, I chose Support Vector Regression (SVR). For this model, in order to reduce dimensionality, only mean and standard deviation of each feature in each annotated fragment were used. Based on the RMSE values on the validation

dataset, RBF kernel with regularization parameter $C = 0.1$ was chosen. Although far from perfect, the results obtained with such simple model were notably better than taking the mean values for arousal and valence on training set (see table 2).

5 Recurrent neural networks

5.1 Architecture

Two possible architectures utilizing RNNs were considered. The first one, *forked*, comprised one RNN followed by two separate fully connected networks (one for arousal and one for valence). The second one, *parallel*, utilized two completely separated RNN+FC sub-models. While the RMSE values did not differ much for the two, the forked architecture predicted highly correlated values for arousal and valence, which rendered it useless for the this application. Therefore, parallel architecture was chosen.

As for the RNN part of the architecture, GRU was chosen over LSTM, based on the RMSE on the validation dataset. One layer and unidirectional processing were kept, as altering those parameters increased the complexity of the model without improving the results. Only the output from the last time step was passed to the FC part.

The FC part in the final architecture had three layers with ReLU as an activation function after hidden layers. After the last layer, tanh activation was added, in order to get values from range [-1, 1]. Furthermore, dropout layer between RNN and FC parts was used.

5.2 Training process

The networks were trained with the use of Adam optimizer with learning rate 0.001 and weight decay 0.001 (based on preliminary tests, altering those values did not improve the results) and MSE as a loss function. As the model learned really quickly and overfitting on the validation set was apparent, the training was run for 5 epochs.

5.3 Hyperparameter optimization

Various values for GRU hidden size, dropout rate, sequence length (measured in feature extraction windows), and batch size were considered. Mean MSE of arousal and valence on the validation dataset was chosen as a comparison criterion. The final set of hyperparameters is described in table 1.

5.4 Overfitting

As mentioned before, overfitting was a prevalent problem in this project. Therefore, various strategies were employed to counteract it. Besides dropout layer and weight decay (L2 penalty), augmentations of the training dataset were also tested, but to no avail. Ensembles, which had been used on such problems before [4], gave better

Parameter	Final value
GRU hidden size	30
Sequence length	60
Dropout rate	0.25
Batch size	32

Table 1: Final hyperparameter values

results and were chosen for further optimization. The final ensemble comprised 5 parallel neural networks trained over 5 epochs, returning a simple mean of their results.

6 Results & discussion

For the final evaluation, RNN-based models were trained on union of training and validation set. SVR was trained on training set only, as the number of samples was already becoming prohibitive. The ensemble of neural networks gave the best result, as shown in table 2.

Despite its simple architecture, the ensemble performed with comparable RMSE to the submissions to *Mediaeval 2013 Emotion in Music* task (note that the target values for published results were scaled to [-0.5, 0.5] and thus the RMSE values are halved) [2]. Unfortunately, the results could not be compared to the newer publications, as they used smaller dataset [1].

Interestingly, neural networks alone (not in ensemble) did not present a significant improvement over SVR. I, personally, suspect that in the low-dimensional input space neural networks do not have a notable advantage against "traditional" models.

Nevertheless, even though the predicted values are not precise, they are good enough to distinguish general mood of a song. Therefore, the model could be utilized for real-time color-to-mood matching, which was the main goal of the project. A demonstration of the responsive light prototype is available here: `https://youtu.be/xi0QlckEt9w`.

References

[1] Miroslav Malik et al. "Stacked Convolutional and Recurrent Neural Networks for Music Emotion Recognition". In: *arXiv:1706.02292 [cs]* (June 7, 2017). arXiv: 1706.02292. URL: http://arxiv.org/abs/1706.02292 (visited on 01/19/2021).

[2] 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: ACM, 2013, pp. 1–6. ISBN: 978-1-4503-2396-3. DOI: 10.1145/2506364.2506365.

[3] Theodoros Giannakopoulos. *tyiannak/pyAudioAnalysis*. original-date: 2014-08-27T12:43:13Z. Jan. 19, 2021. URL: https://github.com/tyiannak/pyAudioAnalysis (visited on 01/19/2021).

[4] Mingxing Xu et al. "Multi-scale Approaches to the MediaEval 2015 "Emotion in Music" Task". In: (), p. 3. URL: http://ceur-ws.org/Vol-1436/Paper77.pdf.

Model	Arousal	Valence	N
Expected value	0.2870	0.2315	1
SVR	0.2211	0.2153	1
Single parallel NN	0.2179 ± 0.0070	0.2143 ± 0.0055	10
Ensemble	**0.2085 ± 0.0019**	**0.2080 ± 0.0014**	5

Table 2: Mean and standard deviation of RMSE on evaluation dataset