AN EFFICIENT DEEP LEARNING APPROACH TO STAGE LIGHTING CONTROL: PART 2 - MODEL DESCRIPTION *

Amerena Matteo, Cantoni Corrado, Piccirillo Jacopo*, Stucchi Gabriele*

Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano
Piazza Leonardo Da Vinci 32, 20122 Milano, Italy
[matteo.amerena, corrado.cantoni,
jacopo.piccirillo, gabriele.stucchi]@mail.polimi.it

ABSTRACT

Music Emotion Recognition is an emerging field of research, with an increasing interest in the field of computational musicology and music information retrieval. In this second part of the paper, we present in detail the neural network that we implemented in order to map a continuous stream of audio to the Russell's emotional plane, the core of the audio plugin that we developed as a part of our stage lighting control system. We propose a network based on stacked convolutional and recurrent neural networks, with a focus on maintaining a low amount of computational effort due to the reactivity requirement of the system. The model was trained on the MediaEval database and we were able to preserve results comparable to the state-of-the-art using a lightweight model. Our average execution time is considerably small and will allow us to integrate the trained model in our software for light control.

Index Terms— music emotion recognition, deep learning, stage lighting control, raw audio, neural network, lightweight

1. INTRODUCTION

The objective of this research is to produce a neural network able to automatically map music emotions expressed by a performance to stage lighting parameters in a reactive fashion. We were required not only to optimize the precision of the network, but also to minimize the time per prediction in order to have a fluent reactive system.

MER (Music Emotion Recognition) is a novel, rapidly growing, cross-disciplinary field of research revolving around a central issue: the conceptualization and modeling of a highly subjective topic such as the emotions expressed by music. A research conducted on social networks based on music such as Last.fm has shown how the emotions expressed by a musical piece classify themselves third in the tags searched by users, after genre and localization [1], confirming the increasing importance of such topic in the industry, e.g. in music recommendation systems. Musical content research is not the only field in which MER could be applied, as a mathematical abstraction and quantification of emotions in music opens the way to other kind of mappings and applications. The advent of the digital era and the exponential improvement in quality of the audiovisual content led to an increase in demand of systems able to improve the whole experience offered by a performer, including innovative and catching interaction between sound and video. Connecting the mood expressed by a musical piece to an interactive lighting system is a possible and quite new application, such as was done in [2].

The approach to solve the MER task may vary a lot, given the different emotional models used. There are two main categories in which these models are divided: *discrete models* and *dimensional models*. Basic emotions [3] and Geneva Emotional Music Scales (GEMS) [4] belong to the so called *discrete models*, in which songs are associated to a certain word or label. The main drawback of this kind of categorization lies in how each subject interprets the words used to describe a particular emotion. Alongside with the subjectivity of the interpretation of words also the mapping itself from a discrete to a continuous set would have been subject to an high degree of approximation, having to map a finite set to an infinite one. On the other hand, dimensional models try to represent emotions in a continuous N-dimensional space, thus turning the classification problem into a regression problem. The most common model is the Thayer's emotional plane [5] (also known as Russell's model), a 2-D plane where the x-axis corresponds to valence (positive-negative) and the y-axis to arousal (active-inactive). Therefore, the tuple (valence, arousal) can describe a range of emotions as previously described in the first part of this paper [6]. For our task, a 2-dimensional parametric model is the best solution, turning the task into the design of a mapping among continuous parameters. How we designed such mapping has been already discussed in the first part of this paper [6].

Although the most common approach to MER consists in manually extracting features, the choice of the feature set is crucial and should be oriented to fill a "semantic gap" between low-level information and higher level descriptors that we use to depict music [7]. It also requires heavy workload for their extraction [8]. The majority of music emotion recognition systems rely on a front-end that extracts hand-crafted features, most of the times derived from spectro-temporal signal representation. OpenSmile [9] can be taken as an example of such audio feature extractors. Instead, our approach consists in feeding the network with a continuous stream of audio, segmented in windows of 500ms: this way we did not tune our feature selection, since we kept as input the raw audio signal directly coming from the stage as in [8] [10].

Among all the different algorithms proposed in recent years, Malik et al. [11] were able to achieve better results by combining Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), maintaining a relatively low amount of parameters compared to state-of-the-art solutions. In our case, 1-D CNNs directly extract the features, avoiding all the pre-processing required to select or hand-craft them from scratch. A multi-view approach, taken from image processing, has shown improving performances during training [12]. Then, these features are aggregated and fed to a Gated recurrent unit (GRU), in order to exploit their capability

to understand sequential data. We chose a GRU over an Long Short Term Memory (LSTM) approach [8] for the lighter amount of computations needed by the first one.
In this paper we will see in detail the architecture of the model, how it performed and we will compare it to state-of-the-art works. This approach led us to reach results comparable to state-of-the-art networks, maintaining a fair balance between network complexity and computational effort, so that our model could be implemented in a live scenario. It is worth noticing also that we obtained fluctuations in metrics an order of magnitude greater than other approaches as we'll discuss in the results section.

The code is available at [1].

2. ARCHITECTURE

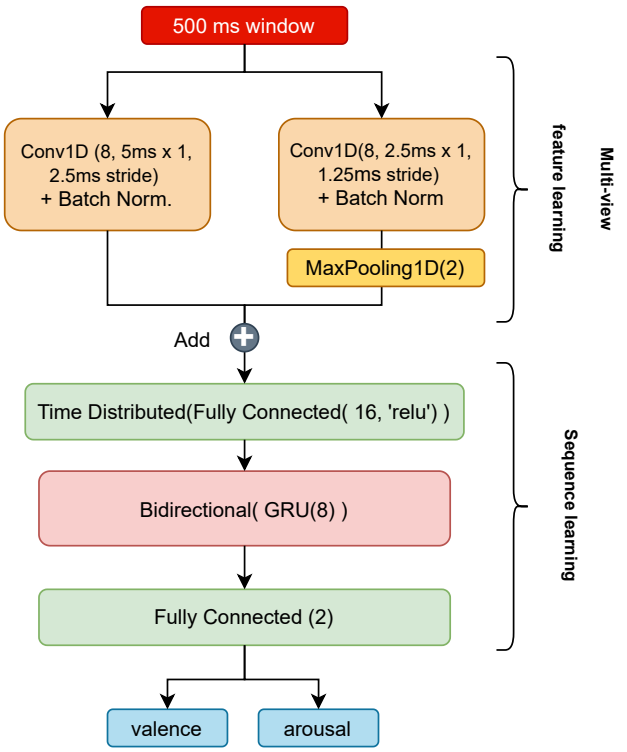

Figure 1: Our proposed network.

The proposed architecture is illustrated in figure 1. As we already discussed, the objective is to find the appropriate mapping

$$\phi : R^M \rightarrow R^2$$

where M is the buffer dimension obtained segmenting the audio stream.
Regression is performed in two stages [8]: a multi-view feature learning stage followed by a sequence learning stage. We preserved the main architecture of [8] for the feature learning stage while we propose a new sequence learning stage based on [10] to achieve a computationally lighter stage via the implementation of a GRU instead of an LSTM. These two implementations merged in our architecture are both based on raw audio signal as input, as our proposed one.

We trained the model with Back Propagation Through Time (BPTT) and Adam optimizer with default parameters [11] [13]. The mean squared error (MSE) and root mean square error (RMSE) were used for the training phase and evaluation phase respectively.

One of our goals [6] was to be able to make a prediction in under 500 milliseconds and as shown in the results we were able to keep the times per prediction very low without sacrificing precision with our architecture.

2.1. Multi-view feature learning

A buffer of 500 ms is fed into a multi-view feature learning stage [8] composed by a fine-view CNN and a coarse-view CNN both using a ReLU activation function. Fine-view has smaller kernel sizes and is meant to observe phase variations within a frame, while coarse-view has bigger kernel size and is meant to capture patterns or periodic behaviour [14].
The fine-view has (110 x 1) kernel size with 55 strides while for the coarse-view has (220 x 1) kernel size with 110 strides. Varying the proportion within the two views' kernel sizes with respect to previous approaches from 4:1 [8] (coarse-view kernel size (4N x 1) with fine-view kernel size (N x 1)) to 2:1 gave us the best results on our dataset, a worth noting detail suggesting further researches on the influence of kernel sizes and their proportion on the precision of a multi-view approach.
Both the fine-view and the coarse-view are then followed by a batch normalization layer to reduce the internal covariate shift [15] and the fine-view branch is then connected to a max pooling layer to match the shapes of the outputs of the two CNNs. The two branches are then combined in a single output with an add layer, which terminates the multi-view stage.

2.2. Sequence learning

As previously mentioned we decided to opt for a sequence learning implementation based on a GRU. This type of recurrent neural networks is comparable to an LSTM architecture, but they are lighter (since they miss an output gate) and provided better results in the fields of speech recognition and polyphonic music modeling [16].

The output of the multi-view processing step is fed in a time distributed fully connected layer that is consequently attached to the Bidirectional GRU to learn temporal information. The advantage of a Bidirectional layer lies in its ability to propagate information in both future and past directions. Given the consequent nature of music, this appeared to be a good choice as analysed in [17].

The output is finally attached to a fully connected layer to complete the mapping from raw audio to valence and arousal.

3. EVALUATION

3.1. Dataset

We used the dataset provided for the MediaEval task, which takes place every year with different challenges in the field of multimedia information retrieval [18]. Through years, they proposed various challenges on MER, and contributed to create a benchmark for various algorithms and features sets baselines on this topic. We

[1] https://github.com/ammlyy/festivalle21

used the dataset from year 2018, that consisted of the development and evaluation set of 2014 (respectively 744 and 1000 songs) and the evaluation set of 2015 (58 songs). It includes rock, pop, soul, blues, electronic, classical, hip-hop, international, experimental, folk, jazz, country genres. It contains full songs or excerpts annotated both continuously (at 1 Hz) and statically. Each annotation ranges in the interval [-1, 1], where negative values stand for low arousal and valence.
Songs were collected from different sources, such as freemusicarchive.org (FMA), jamendo.com and the medleyDB dataset. This could be seen as a drawback, since royalty-free music could be not representative enough, or as in our case, could be not comparable to a live scenario. Also, Festivalle (the festival we are collaborating with for the design of this system) is a jazz festival, and the dataset is not genre-oriented.

All the songs and the excerpts contained in the database were cut to a fixed dimension. The first 15 seconds were ignored, since the annotations could be less stable at the beginning of a song. At the end of this pre-processing phase, we ended up with a set of 30 seconds segments sampled at 44100Hz. We then segmented the songs into 500ms long windows, so that the input to the network consisted in 22050 sequential samples used to predict a pair of values.

3.2. Metrics

The metrics that we used to evaluate our system are Root Mean Squared Error (RMSE) and R^2.

RMSE is widely used in many fields as a statistical metric to define the model performance. Defined as the root value of Mean Squared Error, it describes the standard deviation of the residuals of the true value y and the predicted value y_i.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

The other metrics that we adopted was the coefficient of determination R^2. It tells how close the data are to the fitted regression line. Formally, it is a statistical measure that indicates the proportion of the variance in the dependent variable that is predictable from the independent variable(s).
It is often expressed as percentage or with a value between 0 and 1. Highest score can be 1 and it indicates that the predictor perfectly accounts for variation in the target, while a value of 0 tells us that the predictor is not able to consider it (roughly, it outputs the average value). Negative values of R^2 can occur, telling us that the model behaves worst than an horizontal hyperplane. It is important to note that R-squared cannot determine biases in the data distribution. In fact, despite having an high value of R^2, a model could be not so precise as we expect. Therefore this metric should be analysed together with residual plots and other statistical metrics.

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \overline{y})^2}$$

where y_i represents the observed response variable, $\hat{y}_i$ the predicted value and $\overline{y}$ the mean.

3.3. Setup

The model was created using Keras API running on top of a Tensorflow backend in Python. The experiment was conducted using 10-folds cross validation on the MediaEval dataset that we previously described. The errors are defined as an aggregated metric, without distinguishing along valence or arousal dimensions.

The training was done using a batch size of 8 songs. We used an Adam optimizer for the gradient descent [13], and the Mean Squared Error as loss function.

$$MSE = \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{n}$$

We adopted an early stopping strategy, and we scheduled a decrease in the learning rate, that started from an initial value of 0.001. An L2 regularization term with a value of 0.0001 was applied to the convolutional layers in order to avoid overfitting.

After the training, the model was exported as an hdf5 file and then loaded using the frugally-deep library onto our VST plugin as mentioned in [6].

4. RESULTS AND DISCUSSION

We compare our model to other networks although their authors used slightly different databases. We claim the legitimacy of this approach since our database is an extension of the ones used by other papers, where only a small number of songs were used to evaluate the performance. We instead use a 10-fold technique in order to exploit all of our data. In table 1 we present the performance of the network on the K-fold split, on which we calculated the RMSE and the R^2 metrics.

Fold	RMSE	R2
1	0.198	0.435
2	0.193	0.395
3	0.195	0.351
4	0.191	0.156
5	0.198	0.287
6	0.214	0.232
7	0.245	0.131
8	0.237	0.192
9	0.238	0.196
10	0.223	0.237

Table 1: Results of K-Fold evaluation

In table 2 we report the average value of RMSE compared to different models. The average value of our predictions is comparable to the other networks, although we obtain a standard deviation one order of magnitude higher with respect to the previous works. The relative error is around 10%, while the other approaches have it around 1 %. Testing the network using a K-Fold split validation resulted in an higher fluctuation for the RMSE, anyway also with this increase the performance is comparable with the previous works.
Unfortunately, we were not able to compare our deviation with [8] since this information was not available. Instead they published results for the R^2 metric, that was of 28.2% while ours is of 26.1 %. We obtained a decrease of only 2% that we consider acceptable, given that our network is lighter.

Our goal was to maintain a small amount of parameters by combining two different approaches to this regression problem. Also [8] had a fairly low amount of weights ($\approx$ 3.3k) in their proposed network, but with the multi-view approach we were able to reach slightly better results while not increasing too much the number of parameters ($\approx$4.1k).

Method	RMSE
CRNN[11]	.235 ± 0.003
DNN[10]	.227 ± 0.001
MCRNN[8]	.215
Ours	.21 ± .02

Table 2: Comparison other models of average RMSE

N° runs	Average execution time [ms]
60	4.1 ± 0.4
600	4.4 ± 0.4

Table 3: Execution times in our C++ implementation

As shown in table 3, we were able to predict the values in less than 5ms on average; considering that our limit was to predict in 0.5 seconds (length of each processed buffer) this is an optimal result. The test was done by feeding the VST with a continuous stream of audio. Each run corresponds to a single buffer prediction. The obtained prediction time is two orders of magnitude smaller than our upper limit, thus confirming the goodness of our choices in terms of system weights and allowing us to integrate the model inside a reactive plugin able to operate in quasi real-time.

5. CONCLUSION

This paper introduces a multi-view neural convolutional network stacked with recurrent bidirectional recurrent network able to rapidly perform predictions. The model was trained using raw audio, in contrast to the most common approaches that use human engineered audio features. We were able to maintain a low number of weights and at the same time to have results comparable to state-of-the-art networks. This way, the model will be integrated into a more complex system to control stage lights in a reactive fashion, since the prediction times obtained are suitable for our live scenario.
Also, deep neural network demonstrated again their potential in Music Emotion Recognition tasks. In particular the multi-view approach should be investigated deeper since it allows to get signal representations from different point of views, adding new possibilities to convolutional layers applied to raw audio.

6. REFERENCES

[1] P. Lamere, "Social tagging and music information retrieval," *Journal of New Music Research*, p. 2008.

[2] S.-W. Hsiao, S.-K. Chen, and C.-H. Lee, "Methodology for stage lighting control based on music emotions," *Information Sciences*, vol. 412, 05 2017.

[3] P. Ekman, "An argument for basic emotions," *Cognition and Emotion*, vol. 6, pp. 169–200, May 2019.

[4] . S. Zentner M., Grandjean D., "Emotions evoked by the sound of music: Characterization, classification, and measurement." *Emotion*, vol. 8, pp. 494–521, 2008.

[5] R. E. Thayer, *The Biopsychology of Mood and Arousal*. New York, NY: University Press, 1989.

[6] P. J. S. G. Amerena M., Cantoni C., "An efficient deep learning approach to stage lighting control: Part 1 - system architecture," 2021.

[7] R. Panda, R. M. Malheiro, and R. P. Paiva, "Audio features for music emotion recognition: a survey," *IEEE Transactions on Affective Computing*, pp. 1–1, 2020.

[8] N. HE and S. Ferguson, "Multi-view neural networks for raw audio-based music emotion recognition," in *2020 IEEE International Symposium on Multimedia (ISM)*, 2020, pp. 168–172.

[9] F. Eyben, F. Weninger, F. Gross, and B. Schuller, "Recent developments in opensmile, the munich open-source multimedia feature extractor," in *Proceedings of the 21st ACM International Conference on Multimedia*, ser. MM '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 835–838. [Online]. Available: https://doi.org/10.1145/2502081.2502224

[10] R. Orjesek, R. Jarina, M. Chmulik, and M. Kuba, "Dnn based music emotion recognition from raw audio signal," in *2019 29th International Conference Radioelektronika (RADIOELEKTRONIKA)*, 2019, pp. 1–4.

[11] M. Malik, S. Adavanne, K. Drossos, T. Virtanen, D. Ticha, and R. Jarina, "Stacked convolutional and recurrent neural networks for music emotion recognition," 2017.

[12] B. Wu, E. Zhong, A. Horner, and Q. Yang, "Music emotion recognition by multi-label multi-layer multi-instance multi-view learning," *MM 2014 - Proceedings of the 2014 ACM Conference on Multimedia*, pp. 117–126, 11 2014.

[13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[14] J. Lee, J. Park, K. L. Kim, and J. Nam, "Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms," 2017.

[15] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 448–456. [Online]. Available: http://proceedings.mlr.press/v37/ioffe15.html

[16] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014.

[17] X. Li, J. Tian, M. Xu, Y. Ning, and L. Cai, "Dblstm-based multi-scale fusion for dynamic emotion prediction in music," in *2016 IEEE International Conference on Multimedia and Expo (ICME)*, 2016, pp. 1–6.

[18] A. Alajanki, Y.-H. Yang, and M. Soleymani, "Benchmarking music emotion recognition systems," *PLOS ONE*, 2016, under review.