SHORT-TIME EMOTION TRACKER USING A CONVOLUTIONAL AUTOENCODER

Yoon mo Yang
ECE Department at University of Rochester
yyang106@ur.rochester.edu

ABSTRACT

Music is not only meant for entertainment, but has the power to deliver different emotions to people. From the time machine learning and deep learning came under the spotlight, researchers have tried to use these techniques to retrieve a variety of information from music—genre, instrumentation and even emotion. In this paper, we propose a convolutional autoencoder [1] that extracts arousal and valence values that represent the dynamic of emotion of a given song by learning features from a mel spectrogram. The results of the proposed architecture are compared with a baseline model which consists of two convolutional layers and three fully connected layers.

1. INTRODUCTION

1.1 Background

Music has been the closest form of art in human's daily life. We listen to it through every form of media. There is huge amount of music data in the world in the 21st century. As its size keeps increasing, we need more efficient and organized methods to classify it.

Recently, the number of smart speaker users got quite large and the most frequent usage of the smart speakers is music streaming service. The growing popularity of the smart speakers and music streaming service affects music metadata since the users of the speakers began to use the words that represent emotions for music recommendation. Therefore, music emotion recognition is becoming a more important task.

Music emotion recognition is not an easy task since emotion depends on subjective human's perception. Moreover, even we do not completely understand what emotions really and exactly are. So, psychologist came up with two different views of emotion in music to understand it better. One group argues that music simply expresses an emotion and does not allow for any personal experience of emotion in the listener. The other argues that music induces real emotions in the listener. Compared to the first argument, the second view has less concrete evidence [2] since induced emotion is more subjective than expressed emotion in music. It is hard to find a valid criterion to study the second group's view. This paper focuses on the first view that says music expresses emotion rather than elicit it.

1.2 Representations of emotion

There are two main categories to represent models for emotion. The first one is based on discrete perspective while the second is based on dimensional perspective. The models with the discrete perspective utilize some groups of emotional words based on their relevance to given music. The most famous model would be Hevner's model [3]. The main drawback of this category's models is that meaning of each word depends on subjective perception. This might cause confusion when different words are used for the same emotion, e.g. "Sad"-"Melancholy". Figure 1 shows Hevner's 8 emotion groups.

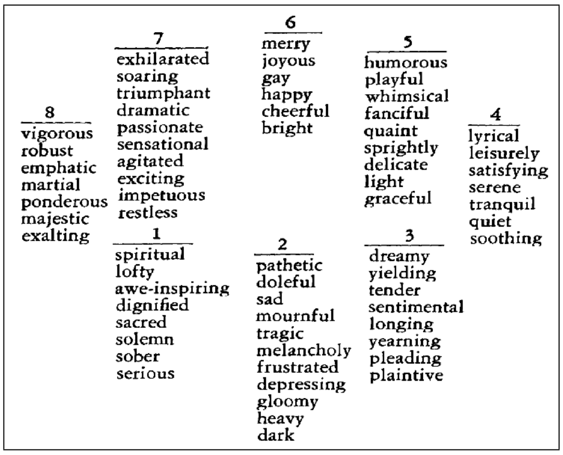

Figure 1. Hevner's 8 emotion groups

In dimensional models, emotion is scaled and measured by a continuous N-dimensional metrics. The most well-known model would be Russell and Thayers V-A model [4]. It has 2 dimensions and one corresponds to arousal (intensity), ranging high-to-low and the other corresponds to valence (an appraisal of polarity), ranging positive-to-negative. The main superiority of dimensional models over discrete models is that they can easily represent emotion that varies through time continuously. This is a good nature of these models since the emotion in a song is usually not the same through its duration. In this paper, our proposed approach use the 2-dimensional V-A model. Figure 2 below shows one of the examples of 2-dimensional V-A space. As you can see, you can locate certain emotion on this space with the corresponding arousal and valence values. For example, "happy" is located far right where valence is really high and arousal is positive but small.

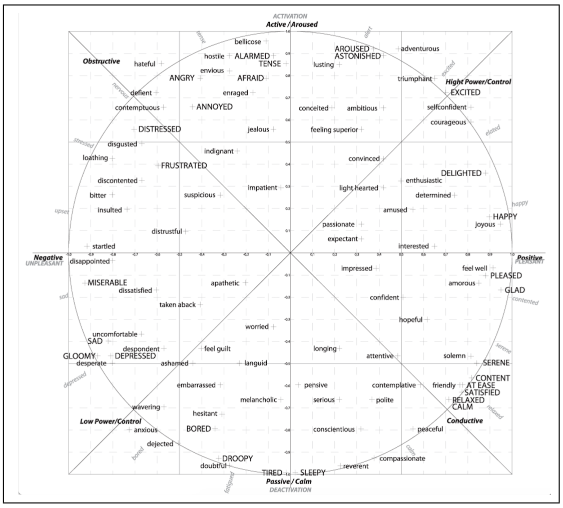

Figure 2. 2-dimensional V-A space

1.3 Related Work

Since this paper uses the dimensional model, we only list previous works that used the same emotion model (dimensional-based). There are two main tasks with the dimensional models, one focuses on static and the other focuses on dynamic of emotion in music. For the second task, the methods proposed were based on support vector regression (SVR) [5], feed-forward neural networks [6], and recurrent neural networks [7]. The dataset for these works consisted of extracted audio features (not raw audio features) and emotional annotations. The features and the annotations were for frames of audio of length 500 ms. There were other researchers who used an ensemble of six long short-term memory (LSTM) RNNs with different input sequence lengths. The final output was predicted from the ensemble using an extreme learning machine. This method achieved a root-mean-square error (RMSE) of 0.230 for arousal and 0.303 for valence [8]. In this paper, we also focus on the dynamic of emotion in music that changes over time.

2. PROPOSED METHOD

2.1 Outline

With this background from Section 1, we propose a convolutional autoencoder architecture for tracking the dynamic of emotion from music. With a proper dataset, data preprocessing and training process, we would be able to estimate pairs of arousal and valence values from music dynamically. Autoencoders are one of the neural network architectures that are often used when the inputs should be copied to the outputs. One of the advantages of the architectures is they compresses the input into a latent-space representation where the features of the neural network lie. Autoencoders generally consist of two parts; encoder that compresses the input in to a latent space and decoder that reconstructs the input from the latent space. Figure 3 shows the architecture of the convolutional autoencoder proposed in this paper.

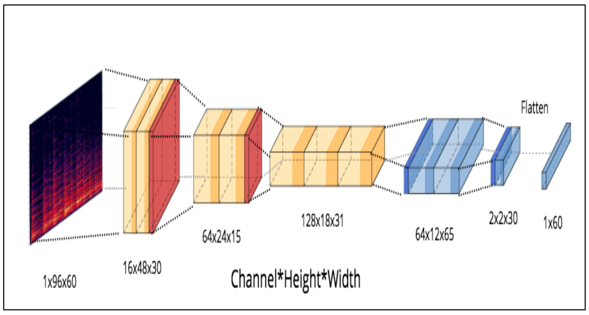

Figure 3. Architecture of the convolutional autoencoder proposed

If the purpose of using autoencoders is just copying the input to the output, using them is pointless. But the latent space helps us to obtain useful (salient) features when it has smaller dimensions than the input. Therefore, this characteristic of this architecture is good for data denoising and dimensionality reduction which leads to a low computational cost. Deep learning researchers have implemented different types of autoencoders. Each type aims to achieve different kinds of properties. In this paper, we utilize a convolutional autoencoder to extract arousal and valence values that represent the dynamic of emotion from a given song. We chose this specific type of autoencoders because unlike the convolutional autoencoder, traditional autoencoders which only consist of fully connected layers do not take account the fact that a signal can be seen as a sum of other signals. By changing the filter size of a convolutional layer, we can make our neural network learn temporal information of the input signal. To the best of our knowledge, for music emotion recognition tasks using deep learning approaches, no researchers have tried convolutional autoencoders so far.

The structure of the proposed network in Figure 3 contains 2 convolutional layers as its encoder. After those two layers, a transposed convolutional layer outputs the bottleneck (latent space) of this model. The bottleneck has the dimensions of $128 \times 18 \times 31$. After this bottleneck, two more transposed convolutional layers are added to upsample the inputs in time dimension. Each convolutional (or transposed convolutional) layer except the last layer is followed by a ReLu activation function to introduce some nonlinearities to the model. Right before the output we added a max pooling layer to downsample the inputs in time and mel frequency dimensions. A tanh activation function follows the max pooling layer to introduce nonlinearities and span the features between $[-1, 1]$. Then the final output gets flatten to have the dimensions of $1 \times 1 \times 60$. For the optimizer of the model, we used Adam optimizer. The network was implemented using Pytorch.

2.2 Dataset

Emotion in Music Database from *Soleymani et al* [9] is used for our training and recognition task. This dataset contains 1000 songs collected from Free Music Archive (FMA). Each song was annotated song ids between 1 and 1000. They removed some redundancies from the dataset which reduced its size down to 744 songs now. Its development set has 619 songs and the evaluation set has 125 songs. Each song has duration of 45 seconds all re-encoded to have the same sampling frequency, i.e, 44100Hz. The starting points of the 45 seconds excerpts were randomly chosen. The continuous annotations of arousal and values were annotated by 300 Amazon Mechanical Turk workers. They used the annotation interface made by themselves for the process. It lets the annotators use a mouse to continuously annotate the values on its GUI. They made the workers use the first 15 seconds of each song to identify emotions that music expresses so the annotations can be more precise. The sampling frequency of the annotations are 2Hz so each song has a pair 60 annotations for arousal and valence values for its 30 seconds duration. Note that these annotated arousal and valence values span between -1 and 1. The annotations and song information are stored as csv files in the dataset.

2.3 Data pre-processing

Appropriate pre-processing of the given dataset is important not only to obtain faster training process but also to perform better. Emotion in Music Database contains 45 second clips of 744 songs, the annotations of the corresponding arousal and valence values and the identification of each song as a training set or as a evaluation set. And these text-based data is saved as a csv file. According to the identifications, we first divided the songs into two folders such as ’Train’ and ’Test’. Later, we excluded several songs from the training set and used them as a validation set. Each folder contains two csv files that contain the arousal and valence values of each song.

We reduced the sampling rate of the songs to 22.5kHz. Since the sampling frequency of the annotations are 2Hz (500ms), we need 60 time frames (30 seconds) to make the inputs correspond to the annotations. We first generated mel-spectra from those 619 songs of the ’Train’ folder by using the librosa python library. We extracted the mel-band features from 500ms with 60ms of window length and 30ms of hop size (= 50% overlap). Each mel-spectrogram has a dimension of $1 \times 96 \times 60$. We took average over 17 frames to get this dimension. We discuss why we use mel-spectrograms for our model in Section 3.1.

After generating the mel-spectrograms, we generated a binary file that contains pre-processed data. By using this binary file that contains pre-procssed data, the training speed of our network got enormously increased.

3. EVALUATION

3.1 Audio Features

Mel-spectrogram provides a 2D representation by compressing the Short-Time Fourier Transform in frequency axis. It is optimized for human auditory perception. Therefore, it is more efficient in size than the STFT while preserving perceptually important information [10]. These two natures of mel-spectrogram makes it suitable for our emotion recognition task for the following reason; since it is a 2D representation like an image, it is suitable input for the convolutional autoencoder and its representation of human auditory perception is suitable for emotion recognition tasks. Figure 4 shows one of the mel-spectrograms we generated from the dataset. Note that we generated 96 mel frequency cepstral coefficients for each song.

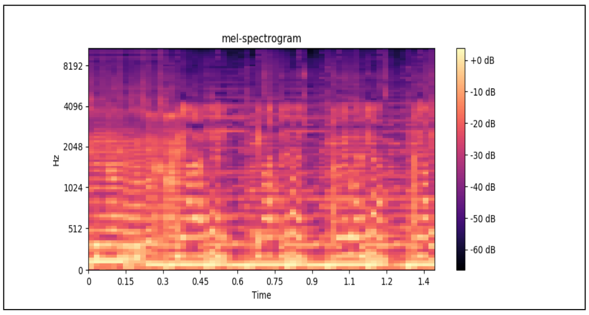

Figure 4. Mel-spectrogram

3.2 Metric

For a model performance and its loss function, we use the Mean Sqaure Error. It computes the average of the squared differences between predictions and targets over the number of features. Given N predictions $\hat{y}_n$ and the corresponding targets y_n, the MSE between them is written as:

$$MSE = \frac{\sum_{n=1}^{N}(\hat{y}_n - y_n)^2}{N} \tag{1}$$

3.3 Evaluation procedure

The very first hyperparamter estimation was done by varying the number of layers of the encoder and decoder. Before we finalized the current model, we first tried 3 and 4 for convolutional and transposed convolutional layers respectively. And later the combination of 2 from each kind was also tested. Identical dropout rates were tested for each layer from the set of $\{0.25, 0.5, 0.75\}$. And we used 0.75 at the end. The learning rate of Adam optimizer was 0.0001 which was chosen from the set of $\{0.001, 0.0005, 0.0001, 0.00005\}$. The mini-batch size was also varied and tested in the set of $\{8, 16, 32, 64\}$. The mini-batch size of 16 was chosen from the set based on the variances in the training set error and in the validation set error. In this paper, the sequence length was not varied and fixed at 60 as same as the number of annotations for each song. Table 1 shows the specification of the final version of the proposed model.

Layer	Kernel Size, Stride, Padding
CNN1	[7,7], [2,2], [3,3]
CNN2	[5,5], [2,2], [2,2]
CNN3 (Transposed)	[5,5], [1,2], [5,1]
CNN4 (Transposed)	[7,7], [1,2], [6,1]
CNN5 (Transposed)	[7,1], [1,1], [1,6]

Table 1. Specification of the proposed model

Other than the hyperparameters of the network, the number of songs of the validation set was also varied. The number of songs of the validation set was chosen from the set of $\{145, 70, 9\}$. 9 songs were eventually used as our validation set since other parameters in the set caused the model to get overfitted so easily due to the small amount of training data.

In order to provide the performance of our proposed model, we implemented a baseline model to compare the performances. Figure 5 shows the architecture of the baseline model.

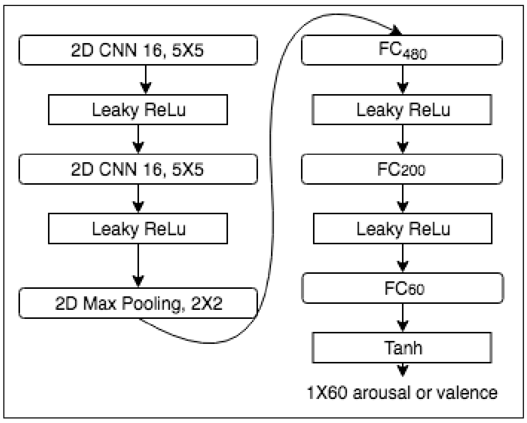

Figure 5. Architecture of the baseline model

Note that for the two convolutional layers of the baseline model in Figure 5, we set the stride size as (3,3) and padding size as (2,2).

4. RESULTS AND DISCUSSION

Figure 6 and Figure 7 below show MSE loss of arousal and valence for each epoch number. As you can see from these two figures, the loss plots of the validation set are not stable enough to converge as the loss plots of the training set do. However, with our current model, this was the best result that we achieved without overfitting.

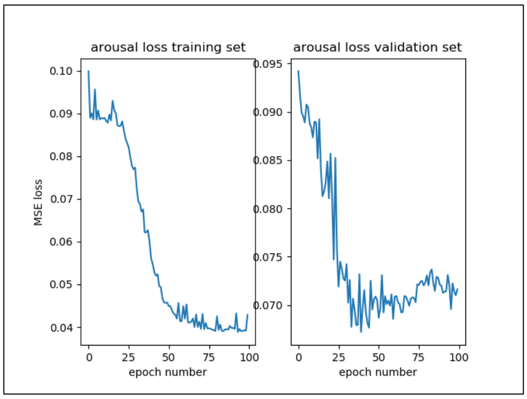

Figure 6. MSE loss of arousal of the training set and validation set for 100 epochs

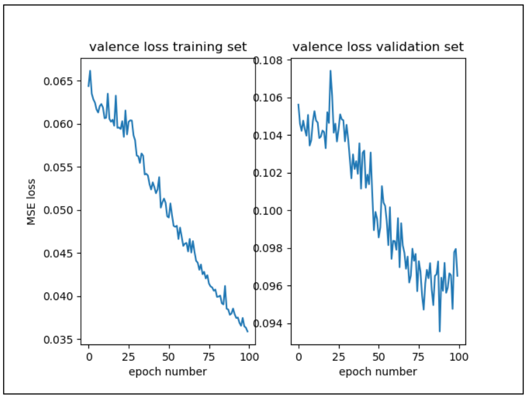

Figure 7. MSE loss of valence of the training set and validation set for 100 epochs

Table 2 below shows the average MSE loss values from the baseline and proposed (convolutional autoencoder) models on the test set. Table 2 shows that our model outperforms the baseline model.

Model	Arousal	Valence
Baseline	0.1088	0.0908
CNN Autoencoder	0.0965	0.0606

Table 2. MSE errors of the baseline and proposed models

Table 3 shows the average RMSE loss values from state-of-the-art [11] (SOTA) that used the advanced version of the dataset used in this paper [12]. Note that SOTA did not use raw audio features. Instead, they used the baseline audio features provided by the dataset, such as mel frequency cepstral coefficients (MFCCs), spectral features such as flux, centroid, kurtosis, and rolloff, and voice related features such as jitter and shimmer.

As you can see from Table 3, our model only outperforms for valence and has a much higher RMSE value on arousal.

Model	Arousal	Valence
SOTA	0.225	0.285
CNN Autoencoder	0.310	0.246

Table 3. MSE errors of the baseline and proposed models

After we evaluated our model on the test set, we also tested the model with ”Bohemian Rhapsody” by Queen. We extracted 30 seconds from the song, the part that starts with a guitar solo and changes its theme with the piano and opera- style singing. Figure 8 shows the distribution of the predictions from our network on Valence-Arousal space. Note that this part is from 2:55 to 3:25 of the original piece.

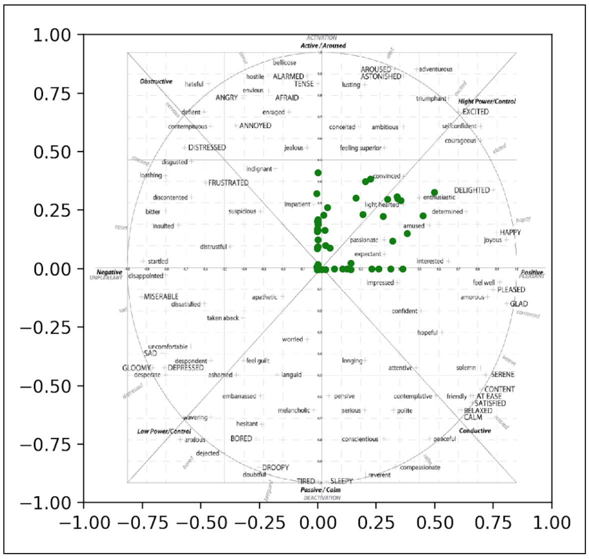

Figure 8. Arousal and valence predictions on ”Bohemian Rhapsody” by Queen

For the guitar solo part, most predictions got gathered around the y axis where the arousal values are high but the valence values are neutral. For the piano and opera singing part, it locates predictions where both arousal and valence values are between 0.3 and 0.6; for example, the predicted emotions are amused, enthusiastic, and light-hearted according to the V-A space.

5. CONCLUSION

From Section 4, we could see that the proposed model outperforms our baseline model on the test set on both arousal and valance. However it still does not outperform SOTA on arousal. We also found out that the proposed network is highly sensitive to hyperparameters and easily gets overfitted. To avoid overfitting, our first future work is to try data augmentation (e.g. adding Gaussian noise) on the training set to see if it gives the model robustness and do the same evaluation steps as we stated in Section 3.3 to tune the hyperparameters. After that, we can implement a new model by combining RNNs with CNNs and compare with the convolutional autoencoder. Since RNNs can take account longer time steps than CNNs, they might give us a better result.

6. REFERENCES

[1] Dan Ciresan Jonathan Masci, Ueli Meier and Jurgen Schmidhuber. *Stacked convolutional auto-encoders for hierarchical feature extraction*. Springer-Verlag., Berlin, Heidelberg, 2011.

[2] M. R Scherer, K. R.; Zentner. Emotional effects of music: production rules. *Music and Emotion: Theory and Research*, pages 361–387, 2001.

[3] K. Hevner. Experimental studies of the elements of expression in music. *American Journal of Psychology*, 48(2):247–267, 1936.

[4] R. E. Thayer. *The Biopsychology of Mood and Arousal*. Oxford University Press, New York, 1989.

[5] M. Malik M. Chmulik, I. Guoth and R. Jarina. Uniza system for the emotion in music task at mediaeval 2015. *in MediaEval*, 2015.

[6] D. Das B. G. Patra, P. Maitra and S. Bandyopadhyay. Mediaeval 2015: Music emotion recognition based on feed-forward neural network. *in MediaEval*, 2015.

[7] T. Pellegrini and V. Barri‘ere. Time-continuous estimation of emotion in music with recurrent neural networks. *MediaEval 2015 Multimedia Benchmark Workshop*, pages 1–3, 2015.

[8] H. Xianyu J. Tian F. Meng M. Xu, X. Li and W. Chen. Multi-scale approaches to the mediaeval 2015 emotion in music task. *in MediaEval*, 2015.

[9] E. M. Schmidt C.-Y. Sha M. Soleymani, M. N. Caro and Y.-H. Yang. 1000 songs for emotional analysis of music. *In Proceedings of the 2nd ACM International Workshop on Crowdsourcing for Multimedia*, pages 1–6, 2013.

[10] Keunwoo Choi, György Fazekas, Kyunghyun Cho, and Mark Sandler. A tutorial on deep learning for music information retrieval. *arXiv:1709.04396*, 2017.

[11] M. Xu Y. Ning X. Li, J. Tian and L. Cai. Dblstmbased multi-scale fusion for dynamic emotion prediction in music in multimedia and expo (icme). *IEEE*, pages 1–6, 2016.

[12] Y.-H. Yang A. Aljanaki and M. Soleymani. Emotion in music task at mediaeval 2015. *in MediaEval*, 2015.