Generation and Classification of Music with Sentiment

Contemporary AI
Computational Creativity
KU Leuven

Nikhil Nagaraj, Rui Barreiros da Silva

Abstract

In this paper, we propose a novel method of generation and classification of music samples based on sentiment. We take novel latent space methods for generation of music and employ classification techniques to partition the latent space based on sentiment. We demonstrate that the sentiment music classification and generation method described in this paper is viable and further exploration is warranted. Future applications of the techniques proposed in this paper include the generation of music based on the sentiment of text (e.g., audiobooks, podcasts) and movies.

The code for the experiments can be referred to on the following URL: `https://github.com/nikhilnagaraj/Capita-AI/`

Introduction

One of the biggest growing applications of deep learning is the generation of modalities (e.g., text, music, and image). Before the growth of deep learning, generative grammars (Holtzman 1981) dominated the generation of music. Predictable set of melody sequences and simplistic music pieces were obtained by these methods. With the improvement of computational capabilities, recent studies have applied deep learning to the generation of music. One of the main reasons for this adaption is the *generality* (Briot, Hadjeres, and Pachet 2017) property of deep learning. Unlike predefined rule-based models, where music can be configured to a subjective pattern of tones and instruments, deep learning emerged as a solution to generate and formulate complex musical interpretations, which can be learned from arbitrary examples. With the improvements in neural networks, more specifically in feed-forward and recurrent neural networks, it became evident that these performed extremely well in classification tasks, due to their adaptive nature.

Feed-forward neural networks, however, are not appropriate for music generation, given the lack of memory. Recurrent neural networks, however, could store information that became suitable to music generation tasks. With the introduction of LSTM networks, long-term memory required by the composition of music became feasible. Studies and researches have since employed numerous recurrent methods that make use of LSTMs to generate musical compositions.

Music sequence is a complex structural modality which consists of numerous factors, such as waveforms characters, leading to a challenging form of modelling. Recent studies have adapted music in a *symbolic sequence* such as MIDI (a format to store musical instructions such as velocity and pitches), which abstracts numerous factors to a language sequence model. By representing composition of music as a language modeling problem, many recurrent novel methods such as mLSTM and hierarchical decoders can be applied to encode a large set of features encapsulated in a music sequence.

Music can capture a wide range of emotions. The generation of music based on sentiment, under the existence of emotion, can be employed in a variety of modalities. In this work, we try to classify and generate music based on its inherited sentiment. One particular application is the generation of music from text on runtime or the generation of musical scores of a movie.

This paper is organized as follows. Background deals with the characteristics of music, sentiment and contemporary work in music generation and classification. In Model, we explain the inner workings of the network used. Data mentions the datasets used to train and classify the model. Under Experiments, we analyze results obtained from different Latent Space sizes and classifiers. The paper then concludes with a discussion of the Further Improvements of our work.

Background

Music Theory Providing a complete background to music theory with its diverse representations and interpretations across various cultures and backgrounds is a herculean task well beyond the scope of this work. Here, the focus shall lie on the aspects of music theory which will be used in the rest of this work.

A ‘beat’ in a piece of music defines a basic unit of time. The length of a beat in general is arbitrary and determined independently for a piece by its ‘tempo’. The tempo for a piece of music is usually indicated in *bpm* (beats per minute) which indicates the number of beats in a 60 second interval, attributing a fixed duration to each beat. E.g. A tempo of 120 bpm equates to 2 beats per second or equivalently 0.5 seconds per beat.

A ‘bar’ of music is a small section of music akin to a sentence in a paragraph of text. Although the ‘time-signature’

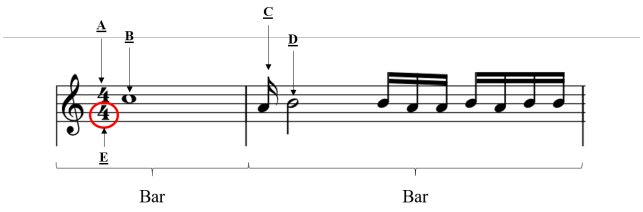

Figure 1: A piece of music containing two bars. **A**: Time signature with the number on top denoting the number of beats in a bar. **B**: Whole note, **C**: Sixteenth Note, **D**: Half note and **E**: The duration of a whole note in the following bars

of a bar independently determines its duration (viz. number of beats), each bar of music in a given piece generally has a uniform number of beats.

'Notes' are the building blocks of music. Each note denotes a music sound. The duration of a note is dependent on its type and the time-signature of the bar in which it is found. A 'whole-note' (a.k.a. semi-breve) spans the length of a bar whereas a 'half-note' spans half the length of a bar. The length of a 'sixteenth-note' is one-sixteenth the length of a whole note (Figure 1).

MIDI Acronym for 'Musical Instrument Digital Interface', it is a technical standard that describes a communication standard, digital interface and electrical connectors for a variety of electronic music equipment and computers. Defined by the MIDI 1.0 specification which was agreed upon in 1982 (Swift 1997), it grew in popularity due to its simplicity. A MIDI file contains instructions for notes, instruments, volumes and effects. These set of instructions when connected to a sound-engine produce the predefined piece of music.

Sentiment in Music Perception of sentiment in music is largely subjective to the listener's background and experience. Any classification of the sentiment embodied by a piece of music based on its characteristics is subject to many exceptions. A gross generalization of musical characteristics and their relation to sentiment is mentioned here.

- **Tempo** A slow piece of music is generally perceived to be negative, whereas a peppy, fast number might evoke a more positive sentiment.
- **Rhythm** Smooth, regular rhythm is generally used to create a sense of calm as opposed to clashing rhythms which can cause a sensation of uneasiness in the listener
- **Volume/Loudness** A soft track is generally favored during somber moments, whereas a louder track signifies more intense emotion.
- **Melody** Similar to rhythm, complementing and clashing melodies create feelings of calm and tumult respectively.

Music Sentiment Classification Classification of music sentiment without a human listener involves interpreting various characteristics of the given piece and their temporal variation. Colace and Casaburi propose a supervised approach to sentiment classification using features extracted from the given piece. Using intensity & loudness, cepstrum, LPC, pitch and voice quality fed to a regression algorithm to generate valence and arousal values for the sample (Colace and Casaburi 2016). Hu et al. compare the performance of lyrics and audio as indicators of emotional content of music (Hu and Downie 2010).

Music generation Perhaps Khalil Gibran foresaw the modern era of automated music generation when he said *'Music is the language of the spirit'*. Modern approaches for music generation treat music as a language, a sequence of symbols, each with its own significance partly determined by those around it; like words in a sentence.

Musenet [1] generates 4-minute long compositions using GPT-2, a large scale transformer model. Music Transformer[2] aims to generate music with long term structure using a transformer. Each of these models represent a given music sample as a sequence of text tokens. Representations across different networks might vary but they generally incorporate information about the pitch, tempo and velocity of every note.

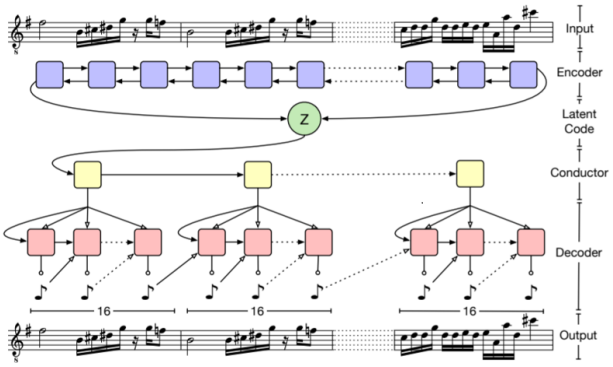

Figure 2: **MusicVAE: Network Architecture** The presence of a conductor followed by a decoder for each bar constitutes a hierarchical decoder, which forces the latent code to encode more significant long term characteristics of a sample (Roberts et al. 2019)

Model

We propose a novel method of generation and classification of music samples based on sentiment. We build upon the work of Ferreira et al. (Ferreira and Whitehead 2019) and the ideas used in MusicVAE (Roberts et al. 2019). Ferreira et al. (Ferreira and Whitehead 2019) trained a mLSTM network to generate music and used the values of its hidden layers as inputs for a logistic classifier to determine the sentiment of the piece. Using the magnitude of the weights in the trained classifier, they isolate the most important features and use a Genetic Algorithm to adaptively modify said values and generate music matching the required sentiment. They work with two coarse sentiments: namely positive and negative.

[1] MuseNet: `https://openai.com/blog/musenet/`

[2] Music Transformer: `https://research.google/pubs/pub47717/`

A novel approach to generate music samples was proposed in MusicVAE[3] which used a hierarchical variational autoencoder to encode the latent features of music in a continuous latent space, thus allowing sampling and interpolation between samples (Roberts et al. 2019). This approach tries to strike a balance between accurately encoding samples and allowing for meaningful sampling in the latent space. Using a hierarchical decoder, they enforce better encoding of sample characteristics in comparison to a flat decoder, where the autoregressive characteristic of an RNN (decoder) causes the latent code to encode only the initial few notes in the sample (Figure 2).

In this work, we combine the approaches to generation and classification mentioned above. We use the hierarchical VAE (Roberts et al. 2019) to encode music samples into a dense, semantically meaningful latent space. The latent space is then partitioned based on sentiment using a classifier (Figure 3). The partitioned latent space can then be sampled to generate pieces of appropriate sentiment based on the sampling subspace.

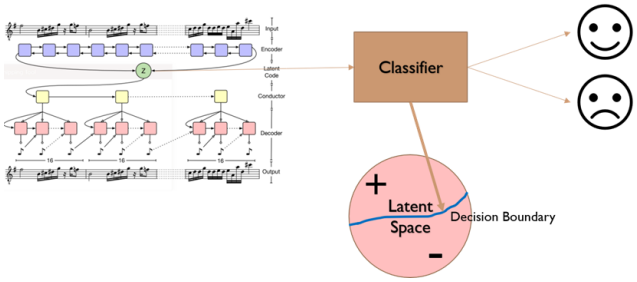

Figure 3: Proposed model for partitioning the latent space

Data

We train the hierarchical VAE with monophonic melodies (i.e. individual melodies) extracted from tracks with one or more melodies. At points where multiple individual notes were audible in the track, the note with the highest pitch was chosen. This is done to allow for the use of the existing training infrastructure with minimal modifications and as a result focus more on the actual evaluation and comparison of results.

Training Dataset We use the Lakh Midi Dataset [4] to train our models. It is a collection of 176,581 unique MIDI files (Raffel).

Classification Evaluation Dataset To evaluate the performance of our network. we use the VGMidi dataset [5]. VGMIDI is a dataset of 95 MIDI labelled piano pieces (966 phrases of 4 bars) and 728 non-labelled ones, all piano arrangements of video game soundtracks. Each piece was annotated by 30 human subjects according to a valence-arousal (dimensional) model of emotion. The sentiment of each piece was then extracted by summarizing the 30 annotations and mapping the valence axis to a coarse sentiment, positive or negative (Ferreira and Whitehead 2019).

Data Representation We follow the data representation methods used in MusicVAE (Roberts et al. 2019). We use Monophonic melodies, which are encoded as 16^{th} note events. This results in a 130 dimensional output space (categorical distribution over tokens) with 128 "note-on" tokens for the 128 MIDI pitches, plus single tokens for "note-off" and "rest". To allow for the use of MusicVAE training infrastructure, each piece is 16 bars long, thus in combination with the encoding scheme each piece has 256 events.

Experiments

To evaluate our approach various experiments involving classification of music by sentiment and generation of music based on sentiment were performed.

Classification

We evaluate the performance of the network against a variety of Latent Space sizes (z-dimension size) and classifiers. Further, the performance of this classification method is evaluated against some popular incidental music.

Effect of z-dimension size We compare the classification performance of the network using a Logistic Regression classifier trained on 256, 512 and 1024 latent space sizes. While we train the VAE from scratch for 256, 512, and 1024 z-spaces, we also use the pretrained model for 512 dimension z-space to account for any biases that might have crept in due to lack of training time/infrastructure.

Z-size	Accuracy
256	59.38% $\pm$ 0.005
512	58.60% $\pm$ 0.03
512 (pretrained)	65.18% $\pm$ 0.05
1024	58.60% $\pm$ 0.02

Discussion A stark contrast is noticed in the accuracy of the pretrained model against the models which we've trained from scratch. The pretrained model performs much better than our models which might be explained by shorter training times and lower amounts of data. The size of the latent space does not seem to impact the accuracy of the sentiment classifier an implication of which might be that the sizes used are much larger than required to encode sentiment.

Different Classifiers We compare the performance of various classifiers on the 512 dimension z-space pretrained model. We chose this model as it performed the best in our previous comparison. The hyper parameters for each classifier were optimized using grid search with a 10-fold cross validation set.

[3]MusicVAE: https://magenta.tensorflow.org/music-vae

[4]Lakh Midi Dataset: https://colinraffel.com/projects/lmd/

[5]VGMidi: https://github.com/lucasnfe/vgmidi

Classifier	Accuracy
Logistic Regression	65.18%
KNN	55.55%
SVM	61.53%

Discussion Logistic Regression and SVM outperform KNN, due to their ability to filter unnecessary features in a high dimensional space. But the difference between Logistic Regression and SVM is not significant enough to come to any sort of a conclusion on the merits/demerits of either in this task.

Comparison with SoA We compare our best results with the SoA results obtained by Ferreira et al (Ferreira and Whitehead 2019).

Model	Accuracy
Ferreira et al. (Gen. mLSTM + Log. Reg.)	89.83%
512 z-size VAE + Log. Reg. (Our model)	65.18%
Ferreira et al. baseline (supervised mLSTM)	60.35%

Discussion Our model fails to outperform the current SoA. A reason for the same is that our model is trained on vastly different data than on which it is evaluated. This is not the case for the generative mLSTM + Logistic regression model (Ferreira and Whitehead 2019). Another reason might be the lack of a training objective which incentivizes effective and classifiable storage of sentiment information.

Popular Music Classification Using a Logistic Regression classifier and the pretrained 512 dimension z-space model, we try classifying 12 pieces of popular incidental music whose sentiment is not under much doubt. We use a voting based method to classify the track, where the predicted sentiment is the weighted majority of the sentiment of each segment. We obtain an accuracy of **52%**.

Song	Ground Truth	Predicted
Jingle Bell	Positive	Positive
007 Theme	Positive	Positive
Oogway's Death Theme	Negative	Negative
Mia & Sebastian's Theme	Negative	Positive
A whole new world	Positive	Positive
Sadness & Sorrow	Negative	Positive

Discussion The network does barely better than a random guess while classifying unseen tracks. We hypothesize that this is attributable to the fact that the model has to judge sentiment based on only monophonic melodies. For most soundtracks, the sentiment of the track is embodied in the backing tracks which the model does not have access to. Further experiments by training the model on the complete track should aid in improving generalised classification accuracy.

Generation

To evaluate the performance of the model on a sentiment based music generation task, we generated 10 positive and 10 negative samples and asked 10 human evaluators to classify the generated pieces.

	True Positive	True Negative
Predicted Positive	70%	45%
Predicted Negative	30%	55%

Discussion The generation task demonstrates a considerable bias towards the generated music being of positive sentiment. Like the popular music classification task, we hypothesize that this is due to the monophonic constraint placed on the network. Human perception depends on the complete track.

Further Improvements

Currently the network is hamstrung by the amount of data and the lack of appropriate training time and infrastructure. Training the current network on more data for a longer period of time and further optimizing training methods should yield better results.

Using a sentiment based training objective The VAE is currently trained to effectively encode samples without a specific focus on effectively encoding sentiment based information in an interpretable/classifiable manner. One method to rectify this could be using the classification loss as an additional loss function along with the reconstruction loss while training the Variational AutoEncoder.

Using polyphonic melodies Currently, the network is capable of classifying only monophonic melodies. Unfortunately a lot of the sentiment in music is expressed by the backing tracks, the tracks which the current network ignores. Modifying the network such that it is trained and evaluated on polyphonic melodies should allow the network to gather more information about the piece of music and the sentiment it portrays, thus improving generation and classification capabilities.

Effectively mapping the latent space for various sentiments In this work, pieces are classified as positive or negative. A variety of finer sentiment can be explored. Different directions in the latent space can be mapped out to indicate a change in the type/intensity of sentiment expressed. Voynov et al. (Voynov and Babenko 2020) offer a viable approach to discover interpretable directions in the latent space.

Real world applications Using a model which is able to receive fine grained sentiment and generate music accordingly, one could create an automated music composer for the spoken word. Networks which classify the sentiment of a piece of text could be used to provide the appropriate input to the generative network which produces music as required. Going forward, this could be applied to visual and other media.

Conclusion

In this work we try to combine two diverse approaches to music generation and sentiment classification to produce a novel approach to generate and classify music based on sentiment. This current work demonstrates the viability of such

an approach. Improvements to the training setup and data might lead to better accuracy and quality of generated music.

References

[Briot, Hadjeres, and Pachet 2017] Briot, J.-P.; Hadjeres, G.; and Pachet, F.-D. 2017. Deep learning techniques for music generation–a survey. *arXiv preprint arXiv:1709.01620*.

[Colace and Casaburi 2016] Colace, F., and Casaburi, L. 2016. An approach for sentiment classification of music. In *ICEIS (2)*, 421–426.

[Ferreira and Whitehead 2019] Ferreira, L. N., and Whitehead, J. 2019. Learning to generate music with sentiment.

[Holtzman 1981] Holtzman, S. R. 1981. Using generative grammars for music composition. *Computer Music Journal* 5(1):51–64.

[Hu and Downie 2010] Hu, X., and Downie, J. S. 2010. When lyrics outperform audio for music mood classification: A feature analysis. In *ISMIR*, 619–624.

[Raffel] Raffel, C. Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching. 222.

[Roberts et al. 2019] Roberts, A.; Engel, J.; Raffel, C.; Hawthorne, C.; and Eck, D. 2019. A hierarchical latent vector model for learning long-term structure in music. *arXiv:1803.05428 [cs, eess, stat]*. arXiv: 1803.05428.

[Swift 1997] Swift, A. 1997. An introduction to midi.

[Voynov and Babenko 2020] Voynov, A., and Babenko, A. 2020. Unsupervised discovery of interpretable directions in the gan latent space. *arXiv preprint arXiv:2002.03754*.

Other Details

Time We spent a lot more time than the recommended 50-60hrs (excluding training time). Apart from the actual implementation, the major bottleneck was creating an idea feasible within the constraints of the course.
Contribution Both of us contributed equally to the project.
Training Infrastructure We use Google Colab[6] to train and test our networks.

[6] `https://colab.research.google.com`