CS230

Literary Muzak

Chuan He
chuanhe@stanford.edu

Liang Ping Koh
lpkoh@stanford.edu

Qiyin Wu
qiyin@stanford.edu

Abstract

Translating text to music is difficult as identifying the emotional quality of text as well as generating complex musical output of a certain emotional quality remain difficult and unsolved problems. This paper surveys and leverages state of the art transformer models for emotional identification in text, and experiments with the use of bidirectional LSTMs in a conditional GAN network, coupled with transfer learning, to generate music from emotional identifiers in a text. Realistic sounding music was generated with subtle difference between emotional classes, but further improvement could be made by looking for more comprehensive datasets and making use of better methods of embedding emotional labels into the music.

1 Introduction

Github: https://github.com/spicypig/text_to_music

We are trying to transform a piece of text into a piece of music. We train two models, a text to emotion classifier and emotion to music generator.

For the text to emotion classifier, we train a transformer model on the SemEval2018 dataset of 6857 tweets, each with a binary label for each of the following emotions: {Anger, Anticipation, Disgust, Fear, Joy, Sad, Surprise, Trust}.

We then develop a mapping function that takes an emotion from the class above and maps it to the more restricted {Sad, Happy, Peaceful, Scary}. This allows us to use our classifier to classify a text, and then translate that label into a label that our next model can recognize.

For our emotion to music generator, we build a conditional GAN to train a generator that takes in noise and an emotion label from the restricted class above and generates a music of that emotion. We utilize bidirectional LSTM models for both discriminator and generator. We first train the conditional GAN on 307 Pokemon pieces, 88 pop songs and 92 final fantasy pieces. We then take these weights, and then continue training with them on a set of 200 labelled MIDI music pieces, each training input with 100 frames (each frame is a note/chord).

We are motivated by this project because we wondered what it would be like to have musical accompaniment to reading. It is also technically interesting because of two things:

- Text to emotion classifiers still greatly lag human level accuracy, and we wanted to explore different architectures that represent the state of art in this field.
- Emotion to music generation is a difficult task that involves mapping a low dimensional input to a high dimensional, complex, and patterned output. We wanted to design a GAN for this task and improve it, and experiment with recently developed ideas for this task.

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

2 Related work

2.1 Text to Emotion classifier

2.1.1 Practical Text Classification With Large Pre-Trained Language Models [1]

The paper compares two main models, multiplicative LSTM models, and transformer language models, to classify massive datasets into one of eight emotion classes. Multiplicative LSTMs are novel RNNs for sequence modelling that combine (LSTM) and multiplicative RNN architectures. It has different recurrent transition functions for each possible input. It outperforms standard LSTM and its deep variants for a range of character level modelling tasks, and that this improvement increases with the complexity of the task. The transformer architecture replaces RNN cells with self-attention and point-wise fully connected layers, which are highly parallelizable and thus cheaper to compute. Together with positional encoding, transformers are able to capture long-range dependencies with vague relative token positions.

Via our exploration, this is the state of the art model, outperforming all other models we have seen. It utilizes architectures recently developed at OpenAI. The project differs from our work in that the emotional labels are not the ones present in our dataset, but we adapt that to our purposes by designing a mapping function to map their emotional labels to ours.

2.2 Emotion to Music Generator

2.2.1 LSTM Based Music Generation System [2]

The paper tests an algorithm which can be used to generate musical notes using Recurrent Neural Networks (RNN), principally Long Short-Term Memory (LSTM) networks. LSTMs have an advantage because they can recall past details and structure of musical notes in the generation of the next notes.

The work differs from ours because they do not condition their music on a specific emotion, but their use of the LSTM is sensible and we adopt it.

2.2.2 DeepJ: Style-Specific Music Generation [3]

DeepJ is an end-to-end generative model that is capable of composing music conditioned on a specific mixture of composer styles. They condition each layer on a style by embedding it into the layer. Furthermore, the embedding representation is learned, rather than just using a one hot representation provided by the input. A weakness they mention is that there is a lack of long term structure and central theme in the music. They mentioned GANs as a way to better learn the long term structure.

We decided to embed emotions as representations in our input, and capture the long term structure of music by training a GAN on large corpuses of music so that it may learn the long term structure of music from natural music pieces. Furthermore, we create the central theme in the music via the use of emotion as an anchor for the music piece.

2.2.3 Learning To Generate Music with Sentiment [4]

This paper presented a generative mLSTM that can be controlled to generate symbolic music with a given sentiment. Results showed that the model outperformed an equivalent LSTM trained in a fully supervised way when compared using a classifier that classified music into sentiments. This represents a possible future direction for our project in which we use mLSTMs instead of LSTMs.

2.2.4 Music Generation Using Bidirectional Recurrent Network [5]

This paper proposes a music generation model based on bidirectional recurrent neural network, which can effectively explore the complex relationship between notes and obtain the conditional probability from time and pitch dimensions. Experiments with classical piano datasets have demonstrated higher performance in music generation tasks compared to the existing unidirectional LSTM method.

We incorporate the use of bidirectional LSTMs from this paper in our GAN. The Bidirectional LSTM layers contextualize the present note by using both past and future information. This layer trains two LSTMs in parallel with one reading the inputs in reverse order.

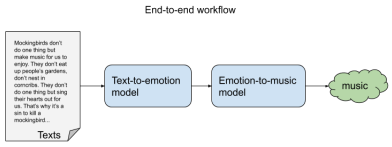
3 Dataset and Features

For text to emotion, we split our SemEval2018 tweets of with binary labels in eight emotion classes into 70-10-20 test/dev/val sets. No preprocessing was needed beyond this for SemEval dataset.

For emotion to music, we have 307 Pokemon soundtracks, 88 pop songs, 92 final fantasy soundtracks and 200 emotion labelled music pieces. We write a script using Music21 to convert the MIDI files into notes/chords. For all songs, we use a sliding window of size 100 to break each soundtrack into 100 note segments, to expand our dataset. For the 200 labelled songs, we replace the last five notes with an emotion label. This is how we "embed" our emotion into the dataset. Originally, we replaced 50 notes with the emotional label, but this amplified the emotional signal to the point where it drowned out the music signal, and so we reduced the representation of the emotional label. We also normalized each note/emotion in the input training data to be between -1 and 1. Overall, we had 234613 music snippets, 20000 corresponding to the 200 labelled music examples.

4 Methods

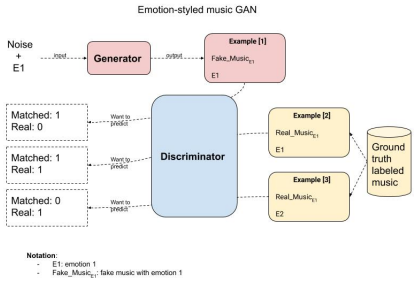
Here is the high level view of our approach:



We have a text to emotion model, and then an emotion to music model to generate music from text.

For text to emotion, we replicated the transformer language model described in related work 2.1.1 for text to emotion classification, then mapped $\{Anger, Anticipation, Disgust, Fear, Surprise \rightarrow Scary\}$, $\{Sad \rightarrow Sad\}$, $\{Joy \rightarrow Happy\}$, and $\{Trust \rightarrow Text\}$.

For emotion to music, we trained a GAN that can take in noise input with an emotional label, and transform that into music of that emotional quality. We summarize this in the following diagram:



Our generator thus has to learn the general structure of music, and also the specific qualities of music of a certain emotional label. Specifically, the generator is fed with random noise and a chosen emotion to make an output that will be converted into a .mid file. The discriminator and the generator has the first two layers as LSTM layers. This allows the discriminator to learn from our music and emotion inputs as sequential data during training. Without these layers, we found that the discriminator was unable to distinguish real music from fake music as long as the generator was able to figure out the discrete domain of the input data. With the LSTM, the generator now has to do more than simply figure out the domain of the real data; it also needs to figure out that music follows certain patterns. In the discriminator, the output layer is a Sigmoid activation function, as we want out outputs to be a single 0 or 1, representing fake data or real data, respectively.

We define our generator and discriminator as follows:

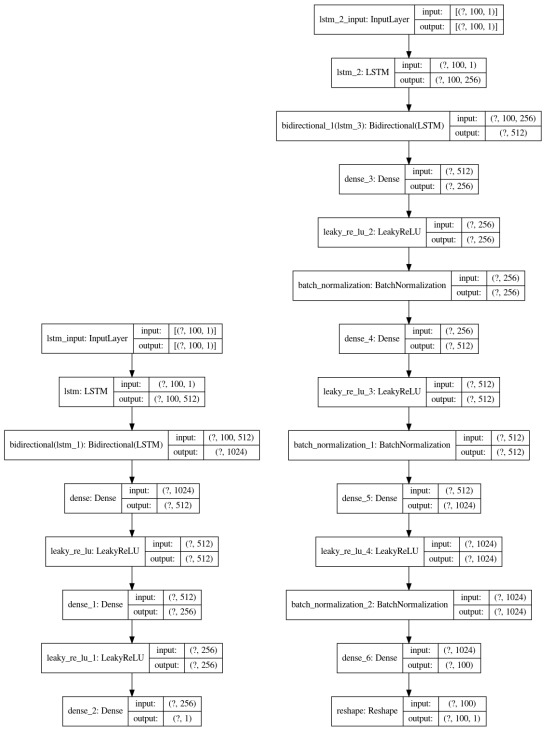
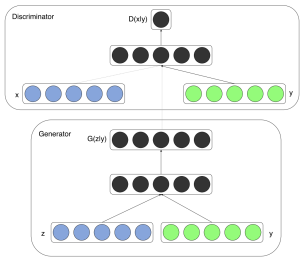


Figure 1: Discriminator (Left), Generator (Right)

We first train our GAN on a large corpus of data (214613 examples) without any emotional label so it can learn to generate real sounding music. We ran a thousand epochs on the training examples. We used binary cross entropy loss for our GAN.



We next train our GAN using emotionally labelled data to teach the generator how to generate real music with an emotional quality. A conditional GAN embeds the condition (label) into the model by creating a representation of them and concatenating it into the input. We encode the class labels into the generator and discriminator models by embedding the label into the music data. As described above, we do so by changing the last five notes of the data into a repeat sequence of the emotional label. We tried other sequences and forms of representation, such as 50 sequence repeats or one label per musical note, but these representations overemphasized emotion relative to note, causing the model to learn an emotion and output an identical note all the time, rather than restructure a diverse set of notes based on an emotion. We then train the GAN with the 20000 labelled examples for a 1000 epochs to allow the model to learn how to generate music of a certain emotional quality.

5 Experiments/Results/Discussion

For the first model, we replicated the results of related work 2.1. We chose the transformer model as it performed the best on the SemEval dataset:

Table 8: F1-score of finetuned language models on SemEval plutchik classification, with IBM Watson as a baseline.

		Anger	Anticipation	Disgust	Fear	Joy	Sadness	Surprise	Trust	**Average**
Semeval	Transformer	**.771**	**.403**	**.764**	**.765**	**.818**	**.691**	**.400**	**.271**	**.610**
	mLSTM	.548	.275	.576	.319	.651	.491	.122	.168	.394
	ELMo	.614	.294	.662	.388	.734	.531	.154	.181	.445
	Watson	.498	-	.331	.149	.684	.359	-	-	-

We managed to replicate the results of the original paper, achieving the same classification accuracy.

For the error analysis of the emotion to music model, we do not have a benchmark to obtain a numerical estimate of the error, but we will provide links to music snippets we generated to show how the music has changed through our iterations.

The first approach, we used only the 20000 labelled examples in our GAN, but this corresponded to a note vocabulary of only 50 notes. Hence, when we trained our model on this, we obtained very poor music pieces as there was very little diversity of sounds and very few training examples, leading to us not properly learning a varied and good approximation to the distribution of natural music. ([6] is link to music snippet)

Seeing this, we saw the need to expand the dataset and allow the generator to learn the natural structure of the music. We thus obtained the other 214613 music pieces trained the music. This expanded our note vocabulary to 531 notes. Furthermore, our generator was much better able to identify the proper structure of music, as our generated music sounds a lot smoother. (Happy [7], Sad [8], Scary [9], Peaceful [10])

Finally, we saved the weights and continued training on our original emotionally labelled 200 training examples to allow our music to learn how to generate music of a certain quality. This allowed us to preserve the natural structure of music we previously learned, but also learn how to transfer emotional qualities to music. The results are shown in the music pieces we have attached (reference). As can be heard, there tends to be higher notes and less jarring music for happy and peaceful music, while sad and scary music tends to have more jarring noises and lower sounds.

For the above, we have used mini-batch size of 32, epochs of 1000, training sequence length of 100, emotion embedding length of 5, output sequence length of 100, also we use Adam optimizer with a learning rate 0.0002, beta one 0.5, batch normalization with momentum 0.8 in the generator, leaky Relu with alpha 0.2, LSTM with and Bidirectional unit numbers. Please refer to [11] for the detailed model architecture.

We trained the C-GAN on AWS EC2 instance with GPU for 1000 epochs, it took about 4 hours to finish. The discriminator and generator is converging at a loss of 0.75.

6 Conclusion/Future Work

For emotion to music, the use of C-GAN with bidirectional LSTMs in both discriminator and generator, as well as use of transfer learning and the tuning of embedding of emotional labels, allowed for strong performance even with a relatively shallow network.

Points of improvement remain. Our current music is differentiated but the difference is subtle, and differentiating points are rather simple. For example, it is primarily the pitch of the note that distinguishes the happy from sad music, while scary music has repeated jarring noises, which is a simplistic way of making something scary. Thus, we could experiment with deeper networks, a much more varied dataset, both labelled and unlabelled, for both transfer and actual learning. Furthermore, previous papers have successfully allowed learning to happen for the representation of the embedding of labels into music piece, suggesting that there are better ways a machine can find to embed emotional data. A flaw is also that our current model overall is two piece rather than end to end from text to music. We made a simplistic assumption that a text is characterized by emotional qualities, but an ambitious end to end model might allow us to learn some deeper/more complex conversion of textual data into complex dimensional outputs that can then be transformed into music. An intermediate step before this would be to explore the extraction of more features, such as length, urgency of text, etc. in addition to simplistic emotional labels that give a richer set of features for more specific music generation. Note that much of this is held back by the availbility of current labeled datasets. Perhaps unsupervised learning could also be explored here.

7 Contributions

Chuan He: mainly worked on music generation, C-GAN coding, and AWS training, also setup the GitHub repository. Liang Ping Koh: mainly worked on the text to emotion coding, and final report and poster. Qiyin Wu: worked on model architecture design, also the C-GAN coding.

References

[1] Neel Kant, Raul Puri, Nikolai Yakovenko, and Bryan Catanzaro. 2018. Practical text classification with large pre-trained language models. arXiv preprint arXiv:1812.01207. (https://arxiv.org/abs/1812.01207)

[2] Sanidhya Mangal, Rahul Modak, Poorva Joshi. 2019. LSTM Based Music Generation System. arXiv:1908.01080 [cs.SD] (https://arxiv.org/abs/1908.01080)

[3] Huanru Henry Mao, Taylor Shin, Garrison W. Cottrell. 2018. DeepJ: Style-Specific Music Generation. arXiv:1801.00887 [cs.SD] (https://arxiv.org/abs/1801.00887)

[4] Lucas N. Ferreira, Jim Whitehead, Learning to generate music with sentiment. (http://www.lucasnferreira.com/papers/2019/ismir-learning.pdf)

[5] Tianyu Jiang ; Qinyin Xiao ; Xueyuan Yin, Music Generation Using Bidirectional Recurrent Network. (https://ieeexplore.ieee.org/document/8839399)

[6] https://onlinesequencer.net/1302194

[7] https://onlinesequencer.net/1302166

[8] https://onlinesequencer.net/1302165

[9] https://onlinesequencer.net/1302163

[10] https://onlinesequencer.net/1302167

[11] https://github.com/spicypig/text$_to_music discriminator - architecture$

[12]