CSC2541 Project Paper: Mood-based Image to Music Synthesis

Mary Elaine Malit
Department of Computer Science
University of Toronto
elainemalit@cs.toronto.edu

Jun Shu Song
Department of Computer Science
University of Toronto
jsong32@cs.utoronto.edu

Abstract

This paper explores the generation of music that expresses a similar mood as a given image. Combining techniques from areas of computer vision and sequence modelling, we show how a CNN with an RNN can generate music conditioned on an image. After training seven different models, we conducted 360 surveys to evaluate our results, and find them promising. In addition, we also explore the ability of variational recurrent autoencoders in capturing global features of music that can potentially help with our objective. Due to time and resource constraints, this exploration has not yet produced a conclusive answer, although we present several directions to take in the future to further enquire into this topic.

1 Introduction

In recent years, advances in generative models have made it easier to synthesize music. However, research in this area usually involves generating music as an end in itself, stopping at the production of reasonable-sounding music without a more specific application. In this paper, we tackle the more concrete application of trying to generate music based on the mood of a given image. We apply lessons from previous research to produce pleasing music in this interesting, multi-modal goal.

This is our research contribution: we show how a joint model of a pre-trained CNN and a long-short-term-memory RNN can be used to generate music with a similar mood as an image. In addition, inspired by recent success stories of the variational autoencoders, we begin the first stages of inquiry into the possibility of using a variational recurrent autoencoder to achieve our objective.

There are many possible applications of a model that can synthesize music from the features of an image. Expanding our project goal even a little bit, one can easily extrapolate to music generation for videos, advertisements, soundtracks, and eventually, perhaps even virtual reality!

2 Related Work

To our knowledge, the first attempt at generating music that is conditioned on images was Nakamura et al.’s research in “Automatic Background Music Generation based on Actors’ Mood and Motions” [1]. Because this occurred in 1994, their process was quite primitive, devoid of the advanced generative techniques from recent years. Their goal, similar to ours, is to generate music based on an image. However, it differs slightly in that their focus is to accompany individual actors’ mood and behaviour with music and sound effects, whereas we are more concerned with synthesizing music that evokes the same mood as the overall atmosphere of an image.

Dunker et al.’s insights in image/music classification [2] led us to choose three moods that are equidistant from each other in Thayer’s mood model: happy, sad and anxious as shown in Figure 1.

Instead of working with a discriminative model like Dunker did, we would like to explore and better understand the features of our data by training generative models.
Peng et al.'s work in Emotion6 [3] showed that it was difficult to classify images into moods even with a state-of-the-art CNN. It demonstrated the difficulty of working with an ambiguous and subjective topic like moods. One important contribution of this paper, however, is the provision of a dataset of pre-classified images, which we used as part of our dataset.

In the past, one of the main reasons music was difficult to generate was because of its temporal nature. To address this, Eck and Schmidhuber developed an LSTM model that successfully produced state of the art blues improvisation [4]. However, this model was built using a very restricted set of chords as input data. In our case, we work with a wide variety of music.

Another model that was able to effectively generate music is Boulanger-Lewandowski et al.’s work in “Modeling Temporal Dependencies in High-Dimensional Sequences” [5]. They were able to generate music by coupling the RNN, which modeled temporal sequence, with an RBM, which modeled the note dependencies.

Inspired by both Eck and Boulanger-Lewandowski’s work, Daniel Johnson developed the biaxial model [6], which became the foundation for ours. Johnson’s biaxial model leverages the recurrent relationship of an RNN in two ways. First, at each iteration, the model sends the input as sequential data through a time axis. The output of this section is then passed to the pitch axis. Along this axis, the model receives the notes played simultaneously as input data. This way, the model learns both the temporal relationship between notes and the joint relationship of chords.

Though unrelated to music, Show and Tell by Vinalys et al. demonstrated how an RNN can be conditioned with the features of an image retrieved through a CNN [7]. The goal of their research is to automatically generate captions given an image. Their impressive results motivated us to use the same structure for our model.

### The CNN-LSTM Model

As shown in Figure 2, our model is fashioned after Show and Tell’s model that generates captions based on an image [7]. Similar to their approach, we use a pre-trained CNN to extract the features of an image, and then condition an RNN with these features to generate music. More specifically, the features extracted from the CNN pass through a fully connected layer to be reduced from 512 x 7 x 7 to a vector with the same dimension as the hidden states of the biaxial LSTM. The image feature vector is then set as the weights of the first layer of the LSTM.

Since there has not been many attempts at generating music conditioned on an image, our CNN-LSTM model does not exactly expand upon a previous model, as much as it amalgamates models that were used for other purposes. Although based on Show and Tell’s caption generator, CNN-LSTM deals specifically with music. The structure of our data is naturally different from that of written text, with many possible co-occurring notes in each timestep for chords, as opposed to single letters.

The model’s learning algorithm is as follows:
Algorithm 1: Biaxial’s gradient update calculation

```plaintext
function Update_Model (input_matrix, img_feature);
    Input : mapped midi matrix M and image feature I
            M ∈ {0, 1}^{time \times note}
            I ∈ R^{1 \times hidden_layer_size}
    for i = 1, . . . , #epochs do
        for ∀{M_j, I_j} do
            M'_j ← dropout(M_j)
            n_{-1} ← I_j
            P_{time}(N) ← time_lstm(M_j)
            M' ← [P(:, -1), M_j(1 :)]^T
            M' ← dropout(M')
            P_t(N) ← pitch_lstm(M')
            Calculate the likelihood given by:
            \prod_{i=1}^{N} p_i \times n_i + (1 - p_i) \times (1 - n_i)
        Backprop (adadelta) to adjust weights
    end
end
```

Figure 2: Our model combines a CNN and an LSTM to generate music conditioned on an image. The hidden state of the 0th timestep is initialized to the image features extracted from the CNN. N denotes the set of all notes, n_i denotes the notes that are played at timestep i. P(N) denotes the probability distribution of all the notes at timestep i.

4 Comparison and Demonstration

4.1 Dataset and Implementation

Our music data consists of video game piano midi files classified by hand into the three target moods of happy, sad and anxious. Our image data comes from Emotion6 [3] and movie-screencaps.com. While the Emotion6 dataset is already classified into moods by the authors of the paper, we categorized the movie screencaps into the three moods ourselves.

We obtained a Keras implementation of a pretrained VGG16 [8], and used this to extract image features and resize their dimensions to one accepted by the biaxial model. The server we used to train our earlier experiments had 4 Intel GPUs and 256 GB of memory. Our latter experiments were trained on a server with 2 8-core Intel CPUs with 32GB of memory.

3
4.2 Experiments

We conducted several experiments on different setups of our model. Table 1 summarizes our experiments. Each model was trained for several days.

<table>
<thead>
<tr>
<th>No. of music clips</th>
<th>Dropout</th>
<th>Image Dataset</th>
<th>Conditional Axis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp1</td>
<td>10</td>
<td>0.5</td>
<td>time</td>
</tr>
<tr>
<td>Exp2</td>
<td>1</td>
<td>0.5</td>
<td>time</td>
</tr>
<tr>
<td>Exp3</td>
<td>3</td>
<td>0.5</td>
<td>time</td>
</tr>
<tr>
<td>Exp4</td>
<td>1</td>
<td>0.35</td>
<td>time</td>
</tr>
<tr>
<td>Exp5</td>
<td>1</td>
<td>0.35</td>
<td>time and pitch</td>
</tr>
<tr>
<td>Exp6</td>
<td>3</td>
<td>0.35</td>
<td>time and pitch</td>
</tr>
<tr>
<td>Exp7</td>
<td>3</td>
<td>0.35</td>
<td>time</td>
</tr>
</tbody>
</table>

Table 1: This table shows the variables for our models. No. music clips indicates the number of music clips mapped to each image during the training. The dropout column shows the dropout rate we used. The Image Dataset column indicates the subset of our images used for that experiment, and the last column indicates which of biaxial’s axes (time or pitch) was conditioned with the image features.

Before performing a more rigorous evaluation, we qualitatively assessed the samples generated from our experiments. From our own assessments, we found that the models that are trained with images mapped to only one music clip were better in evoking a similar mood as the image. However, the music samples seemed to suffer from repetitiveness and lack of musical creativity. Because one of our goals is not only to generate music with a given mood, but also to generate music that sounds good, we found this result a little disappointing.

On the other hand, the models where the images were mapped to multiple music clips during training seemed to generate richer music. However, these models suffered from being unable to produce music with distinguishable moods.

Initially in the first few experiments, we conditioned only the time axis of the LSTM with the image features. When we also conditioned the pitch axis, we did not find the samples to be either better or worse.

The dropout values that we experimented with also did not make much of a difference in the samples generated. As for the number of images we trained on, it was necessary to cut back some of the images due to limited computing and memory resources of our server. We found that removing the movie screencaps seemed to generate better samples for some moods in some experiments but worse samples for others.

4.3 Results

To better discover the effect of each variation on the models, we generated samples from four different models using a small test set of images. After carefully assessing which models generated promising samples, we chose models produced in experiments 2, 3, 5 and 7. For each model, we generated musical samples using two test images from each mood\(^1\). Each of these 24 samples were tested on 15 participants in surveys on Amazon Mechanical Turk (AMT). The survey poses three questions:

1. What emotion do they think the given test image evokes?

\(^1\)https://soundcloud.com/user-675702369-146328751/sets/cnn_rnn_samples
2. What emotion do they think the given music sample evokes?

3. Based on a 7-point Likert Scale (1 – extremely disagree, 7 – extremely agree), how much do they agree with the statement: “The music matches the image”?

The results from AMT are summarized in the graphs shown in Figure 3. Analyzing the survey results, we see that the model from Exp2 fared the best; on average, responders somewhat agree that the mood of the image matches the mood of the music.

We also initially thought that conditioning on the pitch model did not affect the samples very much. However, the results seem to suggest that conditioning on the pitch model has an adverse effect on the samples, since both Exp2 and Exp5 had one-to-one image and music mapping, but Exp5 had a worse score.

In Figure 3b, we also see that most people find anxious images to match anxious music better. We surmise that this is because randomly generated notes that may not produce a coherent chord may give off the feel of anxiety. Hence, such chords are more likely to match anxious images, and decrease the sense of happiness or sadness in samples generated for sad or happy music.

Although these preliminary results are encouraging, they are not yet strong enough. There are many unexplained variations in our results. Our next step was to explore variational autoencoders, particularly in their application to sequential data. Some recent success stories like those mentioned in the next section motivated us to take this direction. We believe that variational recurrent autoencoders can help us learn more about our dataset, provide insight into their latent representation, and capture global features that might just help us produce better music with more distinguishable moods.

5 Related Work on Variational Recurrent Autoencoders

The most relevant work in using variational autoencoders for sequences come from Fabius et al.’s research into Variational Recurrent Auto-Encoders [9]. Their model maps time-series data to latent vectors, and is able to generate new data by sampling in this latent space. Their research explores the variational autoencoding of music, and we attempt to verify their findings with our own dataset.

In addition, Bowman et al. uses the Variational Recurrent Auto-Encoder in the attempt to generate sentences from a continuous space [10]. Their model encodes and decodes entire sentences using LSTMs to learn global features of written text corpus. This enables them to interpolate between data points to create relevant and coherent paragraphs. We would like to apply their finding to music.

Our ultimate goal is to repeat our experiments with the CNN-LSTM, but using a VRAE. However, because we are still in the early stages of investigating the use of VRAE, this next section focuses on
first exploring the abilities of the VRAE in successfully generating music. Thus, this next section does not deal with images or conditioning on their features.

6 Formal Description of VRAE

We explored the use of a VAE to capture meaningful attributes of the music in latent space, in the hopes that these will help us generate better image-conditioned music. Identical to Fabius’s Variational Recurrent Auto-Encoder, our VRAE encodes an entire sequence of music into a 2-dimensional latent variable as demonstrated in Figure 4. The latent variable is obtained from the hidden state of the last timestep, $h_{\text{end}}$, from the encoding RNN [9]:

$$
\mu_z = W_\mu^T h_{\text{end}} + b_\mu \\
\log(\sigma_z) = W_\sigma^T h_{\text{end}} + b_\sigma 
$$

We then sample a $z$ from the normal distribution given by $\mu$ and $\sigma$ and use it to obtain $h_0$ for the decoding RNN [9]:

$$
h_0 = \tanh(W_z^T z + b_z) \\
h_{t+1} = \tanh(W_{\text{dec}}^T h_t + W_{x}^T x_t + b_{\text{dec}}) \\
x_t = \text{sigmoid}(W_{\text{out}}^T h_t + b_{\text{out}})
$$

We worked with two types of variational recurrent autoencoders. One uses vanilla RNNs as encoders and decoders, which we will simply call VRAE, and the other which uses LSTMs, which we’ll call LSTM-VRAE.

For the VRAE, the output of the model can be interpreted as the probability that a note is played at each timestep. As for the LSTM-VRAE, the output requires a little more manipulation, as it is filled with negative numbers that are incompatible with the midi format. To produce correct samples, we employ a softmax function to scale the raw output to range from 0 to 1, and then use threshold values so that only the notes with a higher probability are played.
Figure 5: (a) shows the distribution of the 2-dimensional latent variables in the latent space using the VRAE that has been trained for 100 epochs. (b) shows the distribution of the latent variables using the LSTM-VRAE that has been trained for 3950 epochs.

7 Demonstration

7.1 Dataset

We used the same dataset as used previously with the CNN-LSTM model. This time, during the training process, we extract 120 timestep portions from our midi data to serve as training and test data.

7.2 Experiment

To kick off the experiments, we gave both VRAE and LSTM-VRAE one data point with 120 timesteps to see if they are able to generate music learned from just one sample. Both learned to produce music, though LSTM-VRAE had less overfitting. However, as soon as we introduced more music clips in the training set, we discovered that both models had trouble generating music.

First, the VRAE really overfits. When sampling from the probability distribution, we found that the music generated was repetitive with very distinct patterns that echoed specific training samples. Moreover, when we tested LSTM-VRAE, the samples generated from our process were filled with either very few notes or plenty of random notes.

We decided to visualize the latent space of these models. As shown in Figure 5a, the latent variables encoded by the VRAE do not seem to reflect any differences in features between happy, sad or anxious music. However, as seen in Figure 5b, it appears as if the moods are starting to cluster in the latent encodings of the LSTM-VRAE, which was trained for over 3000 epochs. The sad points seem to cluster mostly on the left, whereas anxious seems to cluster mostly on the right. For now, we can only wrap up our investigations by saying that we found nothing conclusive, and that further experiments and training are required. In the next section, we outline some possible next steps in relation to using variational recurrent autoencoders for music generation.

8 Limitations

In this section, we discuss the limitations of our research and discuss how additional work can address these challenges.

https://soundcloud.com/user-675702369-146328751/sets/vrae_lstm_samples
8.1 Mood Ambiguity

After experimenting with image conditioning, we realize that learning a joint distribution of three moods is a difficult task. The three moods cover a wide range of music in terms of tempo and notes, hence our generated samples would often be ambiguous in terms of mood. It would be interesting to compare our model’s performance with models that learn each mood separately.

Another possible reason for this mood ambiguity is the fact that our images and music are from different sources. The images were collected from Flickr and movies, whereas the midi files are soundtracks to video games. Perhaps if we use images from video clips, and music corresponding to soundtracks from those particular clips, the music and images may be more related, and the model might be able to learn the relationship between them more easily.

8.2 Image Analysis

Images are a large and important component of our model. However, we realized belatedly that we focused all our attention on generating the music. For future work, we should dedicate more attention on the images. For example, we can cluster them to identify any patterns or groups. Perhaps we can also run more experiments and analysis to clean out outliers or those that are ambiguous in mood.

8.3 Training Improvements

While training the CNN-LSTM, VRAE and LSTM-VRAE, we did not provide the models with any guidance for creating music, other than what is already inherent in their structure. We believe that integrating musical properties into the training process might guide these models to reduce variability or instability.

Another limitation we faced is the time and space for training. All of our experiments were slow; VRAE and LSTM-VRAE in particular seem to require a longer training time than was available. We might get better results if we train them for longer, and may even be able to experiment with conditioning the music on images using those models. This was something we were unable to do due to time restrictions.

In addition, our setup requires memory-intensive infrastructures. During the latter part of our experimental stages with CNN-LSTM, we were forced to limit our image dataset to those from Emotion6 only and to remove the movie screenshots. In the future, we should either seek another server with more memory, or reorganize our script so that training does not take up a lot of space.

8.4 Evaluation

The final difficulty we faced while training, and probably, the most problematic, is that there is no formal or structured way to evaluate the model. We were often unable to assess how the model is doing or how we can improve it. The loss error and samples from the model are not enough to indicate which direction we should choose for our next step. Once again, the ambiguity of mood – and the subjective quality of music in general – also make evaluation difficult. Some samples that might sound good to one person does not necessarily make it a good sample for another. This also posed problems for us in assessing whether a model is doing well, and whether the loss functions were properly reflecting the quality of the samples.

9 Conclusion

In this paper, we attempted to generate music based on the mood of a given image. We found that a CNN with an LSTM can produce music that somewhat matches the mood of its conditioning image. While promising, we also suggest more guided experiments in the future and a more rigorous evaluation process to obtain stronger results. On the other hand, we found inconclusive results when it comes to using variational recurrent autoencoders for generating music. The success of the VAE in previous research denotes its ability to capture globally meaningful features in sequential data; and so, it seems theoretically applicable to our research objective. We believe that its promise in theory can eventually be reflected in practice given additional training time and resources.
References


