Touchpoint 2 and Midterm Report

Manas Angalakuduti, Matthew Arnold, Ryan Cooper, Pranav Kandarpa, Bradford Peterson

GitHub Page: https://matthewa1999.github.io/Group11_CS4641/

Abstract (What is the problem? What was done previously? How do you approach it? What are the expected results?):

Generating music can seem uncomplicated when there are given parameters such as rhythm, dynamics, intensity, and pitch; however, generating music from an emotion can be quite difficult. There have been many projects with similar goals, such as inputting a database of song lyrics and generating an accurate tag or genre for the song. We want to be able to generate music given a mood, based on other songs with a respective ‘emotional tag’ that we would have the algorithm train. First, we need to create and expand a dataset. To do this, we will be using a support vector machine to create labels of a song based on factors such as valence (musical positivity) and arousal. Next, we will train a recurrent neural network by using a database of songs’ and their emotional tags. Our ultimate goal is to generate a song given an emotional tag.

Introduction/Background (Motivation: Why? Why is it important? What was done previously/ Related work?):

Music generation has been attempted using neural networks since 1989, but it has gained traction with the discovery of deep neural networks' ability to learn from big data sets. As machine learning algorithms have become more widespread, and music streaming has become a larger part of people's lives, companies like Spotify and Apple have tried their hands at creating the most personalized listening experience. Spotify has recently taken a step forward in their efforts, and has patented software that could allow them to analyze user voices and recommend songs based on emotional state, accent, and other user data points.

Similar music generation projects using neural networks have used long short-term memory recurrent neural networks to generate new music from previous training data. Likewise, we believe that using a Long Short-term Memory network will allow us to create pieces that resonate with people as appealing to a certain mood or emotional state.

We will use a support vector machine to expand our database because they perform well with linear classification. The data is also labelled, which further supports the use of a support vector machine. The original algorithm for support vector machines was created in 1963 by Vladimir N. Vapnik and Alexey Ya. Chervonenkis.

Problem Definition (Inputs? Outputs? Modelling assumption (not a deep learning model, but answer to questions like, Is this an MDP? Is this a classification problem? What assumptions are you making when making this model choice?):

First the classifier must be trained. Therefore, the first input into the problem is a dataset of MIDI files and corresponding labels of valence and arousal. The MIDI files within the dataset are then properly cleaned and labeled. The newly trained classifier is used to expand our current MIDI dataset. The expanded MIDI dataset is then imported into the recurrent neural network. These inputs will be predefined, and each label will accompany a data set that contains a multitude of melodies with the given label. After inputting this into the neural network, the goal of this project is to output a melody that successfully replicates the mood described. We plan to use a long short-term recurrent neural network to generate a novel MIDI sequence given a data set of MIDI sequences depicting each mood.

Because the perception of emotions in music may vary from person to person, certain assumptions must be made. We will make the assumption that valence and arousal will be valid indicators that can classify whether a song is happy, sad, or angry. For example, we will be assuming that a song with low valence and low arousal will be a sad song, which may not always be true.

Data Collection

Data collection was completed in two parts. The SVR was trained with a MIDI dataset with corresponding valence and arousal labels from https://github.com/lucasnfe/vgmidi. This github repository contains the MIDI files that our regressor was initially trained on, which was piano renditions of various video game soundtracks. Each MIDI file was parsed and a list containing the number of occurrences of notes for each file was created. This list was then standardized to ensure that every MIDI track was in C major. In addition, the repository also included two JSON files that contained a valence and arousal array for each MIDI track, which was used for the labels. Each MIDI track inside the JSON file contained a list of floating point numbers indicating valence and arousal. These numbers ranged from -1 to 1. These valence and arousal numbers were averaged for each MIDI track and are used as the response variables for the classifier.

For the second part of data collection, we can currently use any MIDI file as a training sample, so we are just using the vgmidi dataset but not using the valence/arousal labels. Our idea is that we can use the regressor to generate more labeled data that we can then use for our RNN from a variety of MIDI sources with a wide range of valence/arousal content.

Methods

Our approach to this problem involves two separate processes. We first need a valence/arousal classifier to enlarge our dataset of labelled midi data. This will also serve as part of our evaluation to measure the valence/arousal prediction of our model output. For this step we are using sklearn's SVR for regression with default parameters and linear kernel. Our training and testing data must come from the vgmidi dataset because we need midi files with valence/arousal labels for this supervised learning approach. We are currently only using one feature for this classification to get a baseline model for this task. The feature involves the pitch class distributions for each MIDI file, this is performed by counting the number of times each note type or pitch class occurs in the file, then shifting each file to the key of C so that the distributions represent comparable data, and normalizing this vector. This feature is reasonable because for major songs, the major 3rd pitch class will have more energy, while for minor songs the minor 3rd will have more energy, etc. For touchpoint three we will be attempting to improve the performance of our classifier through tweaking hyperparameters and finding more meaningful features to give our LSTM model as much accurately labelled input data as possible.

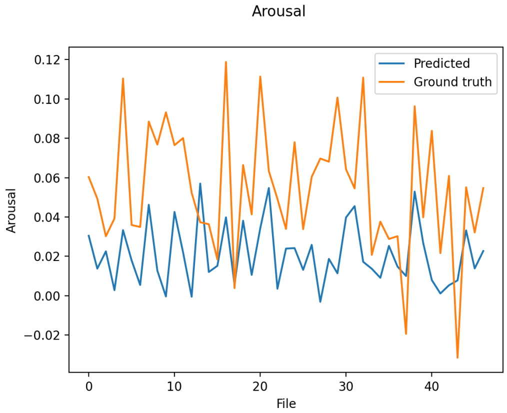
The second process is a LSTM RNN with labelled midi data for training. We found a repository for a project of music generation with LSTM RNN, but this project has no valence/arousal aspect. The model learns the characteristics of the training midi files and generates one novel midi file as its output. We believe this to be a good baseline to work with, and our goal is to incorporate the emotion/intensity into the model architecture to fit our project aims. Our loss function for the model is categorical cross-entropy.

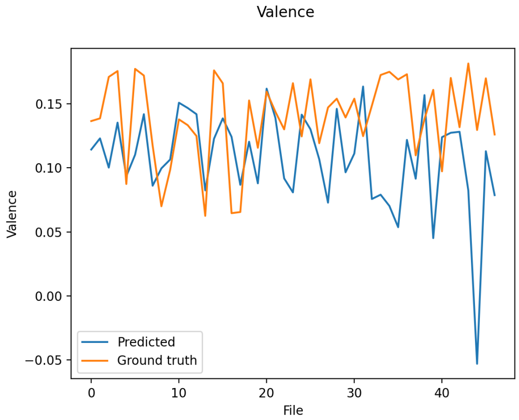
Metrics

For our valence/arousal model, this is a regression problem and we are using the same three metrics on both the arousal predictions, and valence predictions. The metrics are R^2, Mean Squared Error (MSE), and Mean Absolute Error (MAE). These are common metrics for regression tasks and at this preliminary stage they fit our needs well.

For the LSTM RNN metrics evaluation, we are slightly limited by the nature of the problem because as it stands with our current design, we can only use the single prediction from the valence/arousal model on our LSTM output compared to the input. This does not give us many angles of analysis by which to measure our success, but if we can optimize and maximize the performance of our SVR regressor then we should have reliable results.

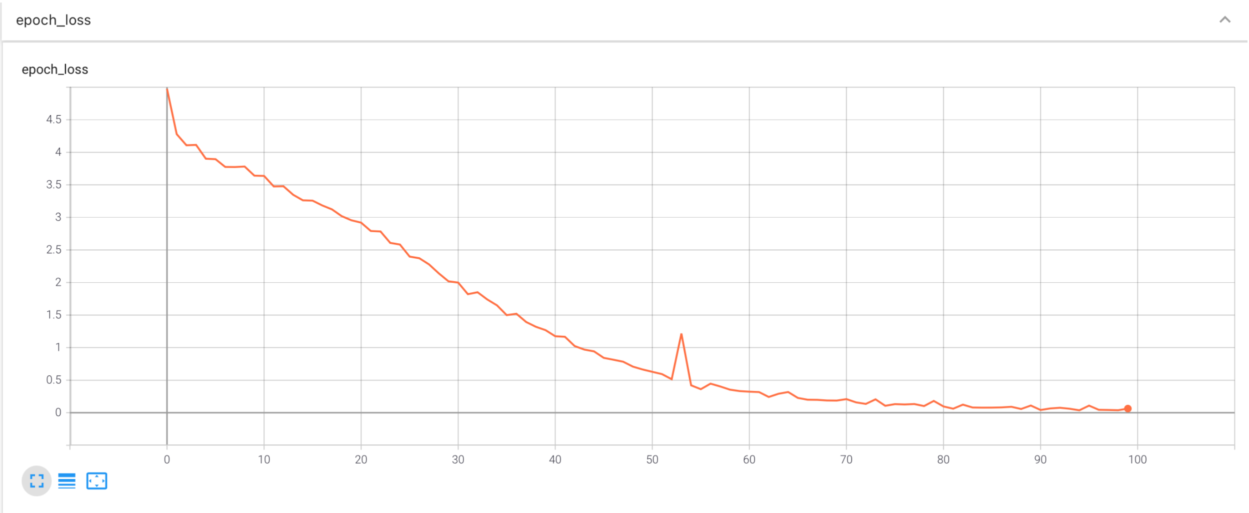
Results





	Valence	**Arousal**
R2	-2.0899	-0.9812
MSE	0.0033	0.002
MAE	0.0439	0.0374

Above is a graph of the predicted and ground truth valence and arousal for each file, using the predicted results from the newly developed SVR classifier. This was done with a 70/30 split of the vgmidi data set (203 files total). For arousal, the R^2 value was -0.9812, the mean squared error was 0.002 and the mean absolute error was 0.0374. Qualitatively, the SVR tended to underestimate the arousal of the songs of the file. In the valence category, R^2 was extremely low at -2.0899, mean squared error was 0.0033, which is quite low, and the mean absolute error was 0.0439. There are a few outliers in the valence category that need to be researched further; the 44th file was predicted to be significantly less than the ground truth.



Above is the loss function using 3 midi files for 100 epochs. This served as a baseline test to observe the performance of the generation on a smaller set of files. The loss function converges very well, but as the training dataset size increases the loss function will not be so smooth. The audio of the output is below.

Discussion

The goal of this touchpoint was to create an SVR to determine the arousal and valence of a particular song. We plan to use this SVR to expand our dataset to gather enough data to use a LSTM recurrent neural network to generate music based on an emotion, which will be quantified by the numerical arousal and valence labels. In setting this goal, we were successful in creating an SVR to determine the arousal and valence of any given MIDI file. However, there are improvements that could be made. Our R^2 value was extremely low in both the valence and arousal categories. In addition, the arousal predictor consistently underestimated the arousal in a song. This could be remedied by weighting the arousal predictor differently within the SVR. The valence predictor performed more accurately, but there are some outliers that will need to be addressed in future iterations of this problem.

Despite these issues, we now have a solid foundation that we can build upon to improve the SVR and to create the music generator. For the SVR, we have some future plans that will both expand the usefulness of the SVR and will increase the accuracy. Firstly, we would like to add more features to the SVR. These include, but are not limited to: average notes per measure, beats per minute, time signature, and the melodic intervals between notes. We hope to add some of these features to the SVR qualifier. Secondly, we hope to include cross validation for the SVR using the method of leave one out testing. This would allow for more useful results since the dataset is relatively small. We will be improving our SVR to enrich our database for our LSTM, but a key issue remains.

A critical task for our next project iteration is to research how to incorporate the data from the SVR (predicted emotional labels) into the LSTM model architecture. We have been researching various academic papers, and we will incorporate these findings into our final project. The implementation of the recurrent neural network is still being considered, and if generating music from 2 emotional tags is unsuccessful, we have other options as well. Another potential method of generating music will be to generate music from one song instead of an emotional tag. This type of problem will be a regression problem, as we are generating new MIDI data based on another input file.

References

Briot, Jean-Pierre, et al. *Deep Learning Techniques for Music Generation*. Springer, 2020.

L. Casini, G. Marfia and M. Roccetti, "Some Reflections on the Potential and Limitations of Deep Learning for Automated Music Generation," *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Bologna, Italy, 2018, pp. 27-31, doi: 10.1109/PIMRC.2018.8581038.

Conklin, Darrell. *Music Generation from Statistical Models*, 2003, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.3.2086.

Cruz, Ricardo, et al. *I-Sounds Emotion-Based Music Generation for Virtual Environments*, 2007, citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.101.2517.

Nayebi, Aran, and Matt Vitelli. *GRUV: Algorithmic Music Generation Using Recurrent Neural Networks*, 2015, cs224d.stanford.edu/reports/NayebiAran.pdf.

Yang, Li-Chia, et al. *MidiNet: A Convolutional Generative Adversarial Network for Symbolic-Domain Music Generation*, Cornell University, 2017, arxiv.org/abs/1703.10847.