CS230

Generate music with customized music style (Music Generation)

Zhengyu Chen
Department of Aeronautics and Astronautics
Stanford University
zchen131@stanford.edu

Cali Chen
U.S. Department of Agriculture
calic@stanford.edu

Hongxu Ma
Google X
hxma@stanford.edu

Abstract

Creating music can be challenge for people who don't have enough music related knowledge. The goal of this project is to create an end-to-end model to generate songs based on a user's facial expression and preferred musical style. For example, the model will generate sad Jazz music if the user feels sad and likes Jazz music. A LSTM model is used for music generation, a CNN model is used for emotion recognition based on facial expression, and a nearest neighbor algorithm is used for music recommendation based on the output of the CNN combined with a song database that includes valence-arousal scores for songs. The models we present in this project can recommend music based on human facial expressions and then use the recommended music as input to generate new music that inspired by the recommended music. The final product is a customized song according to the user's preferred music style that matches the user's emotion.

1 Introduction

People listen to music in their daily life but it is hard for audience to create music using the features they can understand without enough music composition experience. Thus, it is valuable to help people generate some music with deep learning. The features we can easily get from music pieces can be: emotion of the music, music genres, etc.

Our goal is to develop an end to end system starting from mood detection based on facial expression and ending with music generation characterized by the detected mood. The starting input is an image of a face. This is input into a CNN which is trained to recognize facial expressions and outputs both valence and arousal scores, which is part of a psychological framework for characterizing emotion. Valence measures the degree of positive and negative emotions, and can range from most unpleasant (-1) to most pleasant (+1). Arousal measures emotional intensity and ranges from calming (-1) to highly agitated (+1).

The valence/arousal (V-A) score is output from the CNN and a nearest neighbor algorithm is run to find a neighborhood which contains songs V-A scores closest to this output V-A score. These songs are recommendations for the user based on the user's emotion. One of these songs is chosen from the set of recommendations and a sequence from this song is used to seed a LSTM sequence to sequence model to output music that mirrors the user's emotion.

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

2 Related work

2.1 Emotion recognition and music recommendation

Traditional methods of facial recognition detection have relied on handcrafted features and techniques such as non-negative matrix factorization. Since 2013 sufficient training data from real world scenarios have been collected and deep learning methods have been developed that achieved detection accuracy that exceeded previous results by a large margin.[7]

For example, many CNN models such as the VGG-16 and ResNet50 have shown good performance. A recent study showed that ensemble learning of VGG-16 and ResNet50 outperformed either of the individual models.[8, 9]

2.2 Music generation

RNNs have been used to generate music in a way similar to text generation. However, text generation is simpler because at each time sequence there is only need to generate one word at a time, whereas in music there there can be multiple notes played at the same time and notes have duration (the amount of time the note is sustained) and dynamics (how the note is played). More recently, WaveNet implementations[1, 2] have used 1D CNNs with dilations to increase the receptive field. Like RNNS, WaveNet sare also considered autoregressive learning. GANs[3] have also recently been used for music generation.

WaveNet implementations need GPUs to train and the model is probabilistic and autoregressive[5]. However, this model cannot make the samples exhibit interesting structure at timescales of seconds and beyond without scale up the model size.

The other music generation approach is using sequence LSTM model[6] which is the model we used in the project. This is an RNN model and can be performed without using GPUs. The traditional LSTM model is able to generate similar music given one piece of input music. Based on the traditional LSTM model we have, the model we present generated music with music style combination by operate the weights in the model.

3 Dataset and Features

The AffectNet dataset is used for facial expression detection. AffectNet is a databases of facial expression images collected from the internet. Around 440k images were labelled with facial emotion types and valence and arousal scores. There are seven types of emotion categories, including "Happy", "Sad", "Surprise", "Fear", "Disgust", "Anger", "Contempt". There are also "None", "Uncertain" and "Non-face". We use this dataset to train the network to take an image of a face and output V-A scores.[7]

When training for emotion recognition model, we use 50k images randomly selected from AffectNet as training dataset and 10k images as test dataset. Considering that color doesn't affect facial emotion very much, we condense RGB images into grey-scale images. The input size of the image is 49 by 49.

Deezer is a online music streaming service. There is a Deezer mood detection dataset which maps each song in Deezer to a V-A score. After we predict a V-A score from an image of a face, we use this dataset to construct the song recommendation set whose songs have V-A scores close to the predicted V-A score.

We attempted to train our baseline and final music generation model with 349 pieces of classical piano music and 349 of jazz piano music. We use the Music21 package to read and pre-process the MIDI files which included transposing the music to the the key of C major or A minor. We did not reserve any files to use as a validation set as we are generating music and not predicting ground truth. Thus we did did not plan to perform model tuning based on a validation set.

There is a total of 68,018 distinct notes and chords and 108 distinct note durations from our 349 files of jazz music. From our sample of 349 files of classical music, we obtain 212,553 distinct notes and chords and 158 distinct note durations. Distinct note pitches and durations are the "vocabulary" that will be used to generate new music. Note pitches and durations need to be predicted together. We

tried training music generating models with these files in Colab with a Tensor Processing Unit, but were not able to even get past the pre-processing process without crashing. Thus, when training on jazz and classical piano music, we take a random sample of only 10 files.

When training on pop songs for the purpose of generating music that mirrors the mood characterized by the input face image, we train on 50 pop songs. As songs are shorter and much simpler than jazz and classical music, there are fewer unique notes, chords, and durations, thus there was no problem completing the training process.

4 Methods

For the facial emotion recognition model, we implement two CNN based models. One is a 10 layer CNN model with one 11 by 11 conv layer, one 5 by 5 conv layer and three 3 by 3 conv layers. The other one is a transfer learning based on AlexNet. And we found that the transfer learning model performs better in terms of predicting valence and arousal values. MSE is used as the loss function.

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(\hat{y}_t - y_t)^2$$

For the music recommendation part, we calculate the distance of the vector of our facial emotion recognition model to the vectors of all the songs in valence and arousal space.

By now, using an LSTM to generate sequences has become typical, however that is the approach we decided to use even for our final model. This is similar to generating a sequence of characters to build a dinosaur name. However when generating music, even our simplistic approach is more complicated. The input and output of our model requires two sequences. One is a sequence of note pitches and the other is a sequence of note durations.

To preprocess our data, our training set of MIDI files are read in by the Music 21 package, transposed to a common key, and split into sequences of 32 note pitches and durations to be used as the input training set. The training output labels are the notes and durations following each of sequence of length 32.

The baseline LSTM network use categorical crossentropy as loss function and Adam as optimizer. Our final LSTM network also trained with the categorical crossentroy loss function, but with RMS Prop as the optimizer.

Compared to our baseline LSTM network, our final network also includes a Bidirectional LSTM layer in the beginning, and the use of attention for predicting the next note/duration. We expected that the training loss would decrease faster due to these additions and our results support this.

To generate music, the song selected by the facial expression detection algorithm is sampled for a random sequence of length 32. The user can manually impose structure such as Section A, Section B, Section A (variation 2), Section C, Section A (variation 3).

5 Experiments/Results/Discussion

For the facial emotion recognition model, we start with a relatively simple CNN model with 10 layers. However the model could not even handle the emotion classification problem. The best of the model could reach is 51.38% in a 8 classes classification task, shown in Fig. 1

Then we start to implement transfer learning based on the AlexNet model to predict the valence and arousal values. We use two individual models to predict valence and arousal value respectively. We compared the RMSE and correlation between the predicted valence and arousal and the true values of 10 layers CNN model and the AlexNet based model, shown in fig.2

Our baseline LSTM, at 90 epochs shows a training accuracy of 97% and training loss of 0.088. At the same 90 epochs our final model gives a training accuracy of 98% and training loss of 0.0516. This is not surprising as the final version includes a Bidirectional LSTM layer in the beginning and uses attention when prediction the next note. Baseline LSTM model is used to perform music style transformation since this structure is easier to operate between different model weights we have. In

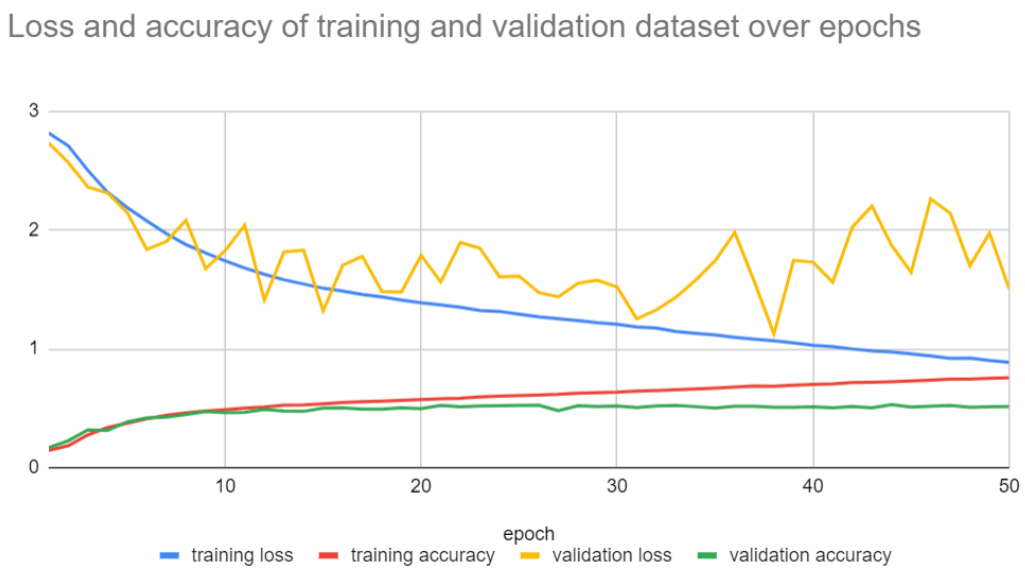

Figure 1: Loss and accuracy of training and validation dataset over epochs

	10 layers CNN Model		AlexNet CNN Model	
	Valence	Arousal	Valence	Arousal
RMSE	0.572	0.543	0.413	0.422
Correlation	0.417	0.436	0.588	0.572

Figure 2: RMSE and correlation metrics of valence and arousal prediction

the experiment, we use the results generated from music generation section(see Fig. 4) as one of the music feature and use jazz music as the other music feature.

The jazz music feature is including improvisation and individual interpretation. In Fig. 5 we can visualize one type of jazz feature is the strong nodes in the scripts with highlight. In Fig. 6 we can visualize that some strong nodes is adapted to the music pieces which indicate the model works properly. When generate the music, we can tune a list of hyper-parameter as listed below:

1. the epoch number for weight data(90 epoch is the optimal in our model).
2. number of initial character we want to use in the music transformation
3. length of music sequence we want to generate.
4. batch size is 16 since 16 samples are enough to update the weight.

The result shows in Fig. 6 using 90 epoch, 35 initial character and sequence size 400.

Bidirectional LSTM model is used to perform music generation with the referencing music shows in Fig. 3 and the generated output shows in Fig. 4.

6 Conclusion/Future Work

For the facial emotion recognition, we found that the simple 10 layer CNN model is too simple to handle this complex task. It gets under-fitted. Thus we use a transfer learning model based on AlexNet which is pre-trained to capture the features of images. And this model gives us the highest performance. And we found that, the RGB doesn't affect the valence and arousal value prediction very much, considering the facial emotion has no difference on red band, green band or blue band. Thus the images are pre-processed into grey-scale.

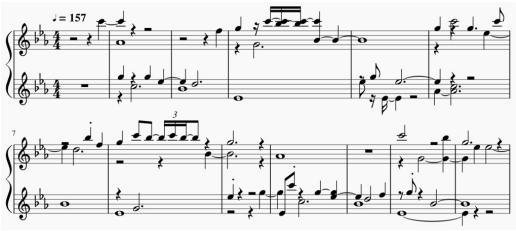

Figure 3: Music referenced in the generation model

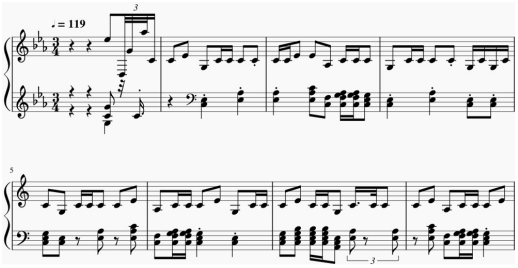

Figure 4: Generated music from referencing music

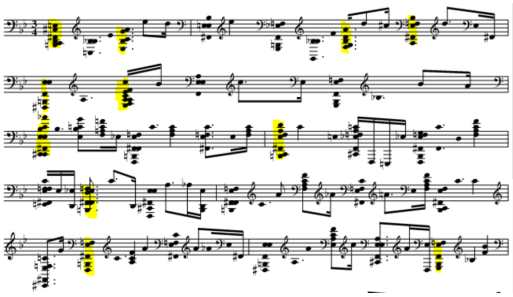

Figure 5: The jazz music with strong notes highlighted

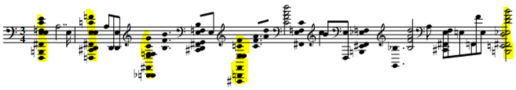

Figure 6: Output music with Jazz feature

Generating music with a desired mood is very challenging. One problem was that the mp3 to MIDI conversion contained extraneous elements other than note pitches and durations. Another major problem is that the mood of a song depends on the whole performance (singer, instrumentation, key, etc). In other words, music is very complex and to generate a particular mood would require a much more complex architecture and much larger training size.

Future work could include the following the ideas below.

1. Learn song structure instead of manually imposing it during the music generation process. There are papers that describe how this can be done using reinforcement learning.
2. Generate music with V-A scores as a direct input instead of sampling from a song with a particular V-A score.
3. Develop a robust evaluation method using music a emotion classifier.

7 Contributions

H.M. contributes to the facial emotion recognition model and music recommendation model. C.C. contributes to the music generation that inspired by recommended music. Z.C. contributes to the music generation that adding user's designated music style. H.M., C.C. and Z.C. all equally contribute to the idea discussion, presentation and draft writing.

References

[1] Sander Dieleman,Aäron van den Oord,Karen Simonyan, "The challenge of realistic music generation:modelling raw audio at scale" 2018,CoRR,abs/1806.10474.

[2] Y. Zhao, X. Wang, L. Juvela and J. Yamagishi, "Transferring Neural Speech Waveform Synthesizers to Musical Instrument Sounds Generation," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 2020, pp. 6269-6273.

[3] G. Brunner, Y. Wang, R. Wattenhofer and S. Zhao, "Symbolic Music Genre Transfer with CycleGAN," 2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI), Volos, 2018, pp. 786-793.

[4] Music dataset:https://data.world/datasets/music

[5] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499, 2016.

[6] Gaurav Sharma, "Music Generation Using Deep Learning" Online available: https://medium.com/datadriveninvestor/music-generation-using-deep-learning-85010fb982e2

[7] Ali Mollahosseini, Behzad Hasani, and Mohammad H. Mahoor, "AffectNet: A New Database for Facial Expression, Valence, and Arousal Computation in the Wild", IEEE Transactions on Affective Computing, 2017.

[8] D. Kollias, et. al.: "Deep Affect Prediction in-the-wild: Aff-Wild Database and Challenge, Deep Architectures, and Beyond". International Journal of Computer Vision (2019).

[9] D. Kollias, et. al. "Recognition of affect in the wild using deep neural networks", CVPRW, 2017.

[10] Abadi, Mart, Barham P, Chen J, Chen Z, Davis A, Dean J, et al. Tensorflow: A system for large-scale machine learning. In: 12th Symposium on Operating Systems Design and Implementation (16). 2016. p. 265–83.

[11] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesnay. Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research, 12, 2825-2830 (2011)

[12] John D. Hunter. Matplotlib: A 2D Graphics Environment, Computing in Science Engineering, 9, 90-95 (2007), DOI:10.1109/MCSE.2007.55 (publisher link)

[13] Stéfan van der Walt, S. Chris Colbert and Gaël Varoquaux. The NumPy Array: A Structure for Efficient Numerical Computation, Computing in Science Engineering, 13, 22-30 (2011)

[14] Travis E. Oliphant. A guide to NumPy, USA: Trelgol Publishing, (2006).