Music Style Transfer via Variational Autoencoder

Yubi Chen[1], *Ruizhe Zhou*[1]

[1]Department of {Physics, Computer Science},
The University of Chicago, Chicago, IL, 60637
{chenyubi, rzhou12}@uchicago.edu

Abstract

In this project, we present a music style transfer approach using pitch, instrument, and velocity features trained by Variational Autoencoder(VAE). Our project is inspired by Brunner's paper published in 2018[1]. In our project, we showed that VAE perform on MIDI style can perform style transfer on symbolic music by automatically altering pitches, dynamics and instruments of a music piece from, e.g., a Pop to a Jazz style. We also can interpolate from our VAE model and produce medleys songs between several pieces of music. The interpolations smoothly change pitch, velocity, and instrumentation to create a harmonic bridge between pieces.

Index Terms: Deep Learning, VAE, Genre, MIDI Style, Neural Networks

1. Contributions

We share the similar contributions. Almost all the works were done together. We both modified the code, discussed the related references, understood the architecture, and ran the program. The different contributions are that, Ruizhe Zhou made the general plans for this project, and we are responsible for different parts in writing proposals, giving presentations, and writing final report.(Ruizhe Zhou on introduction and data-prepossessing; Yubi Chen on Style Classification and evaluation.)

2. Introduction

Music is fundamentally a sequence of notes. A composer constructs long sequences of notes which are then performed through an instrument to produce music. Two important aspects of music are the composition and the performance [2]. The composition focuses on the notes which define the musical score, and the performance focuses on how these musical notes are played.

The ambiguity of interpretation from symbolic sheet music results in a variety of different realizations of the same sheet. This means that the mapping between the sheet symbol and the performed music is not a one-to-one mapping. An example of this are cover songs, Ellis and Poliner stated that Indeed, in pop music, the main purpose of recording a cover version is often to investigate a radically different interpretation of a song. [3].

Music style can be easily distinguished by human instincts, but we need to represent it on paper. So, before we start the experiment, we need to give our own definition of music, that is we determine a music style based on its dynamics, which is the loudness of music; its instruments, and pitch.

Deep generative model can be applied to change properties of existing data in principled way, even transfer properties between data sample. This idea has been highly using in computer vision domain, and can be approved by a lot of astonishing results. In this project, we take a step towards the goal of transferring properties in music. The model that we experienced with is an architecture consists of parallel Variational Autoencoders (VAE) with a shared latent space and an additional style classifier.

3. Related Work

Gatys et al. [6] introduce the concept of neural style transfer and show that pre-trained CNNs can be used to merge the style and content of two images. Van den Oord et al. [7] introduce a VAE model with discrete latent space that is able to perform speaker voice transfer on raw audio data. Malik et al. [8] train a model to add note velocities (loudness) to sheet music, resulting in more realistic sounding playback.

4. Model Architecture

Our model is based on the VAE[4] and applies on a symbolic music representation that extracted from MIDI[5] files. We extend the standard piano roll representation of pitches with dynamics and instrument rolls, modeling the most important information that we considered contained in MIDI files. We uses parallel recurrent encoder-decoder pairs that share a latent space in our modified version VAE. A style classifier is applied to the latent space to force the encoder learns a compact encoding of ”latent style label” that we can then use to perform style transfer. The architecture is shown in Figure 1.

4.1. Symbolic Music Representation

We use the MIDI format to represent our music, which is a symbolic representation. MIDI files have multiple tracks, and can be easily extracted by a pretty_midi package. Tracks can either be on with a certain pitch, held, or be silent. An instrument is assigned to each track. To feed the note pitches into the model we represent them as a tensor $P \in \{0,1\}n_P \cdot n_B \cdot n_T$, where n_P is the number of pitch values, n_B is the number of beats and n_T is the number of tracks. The note velocities are mapped as a matrix $V \in [0,1]n_P \cdot n_B \cdot n_T$. Velocity values between 0.5 and 1 means a note is on for the first time, whereas a value below 0.5 means that either no note is being played, or that the note from the last time step is being held. We design the instrument matrix to be$I = \{0,1\}n_T \cdot n_I$, where n_I is the number of possible instruments in a certain track. The instrument assignment is a property for the whole song, and thus, it remains constant over the duration of one song. Finally, each song in our dataset belongs to a certain style, designated by the style label which is given by the source where we collected the data $S \in \{Classic, Jazz, Pop\}$.

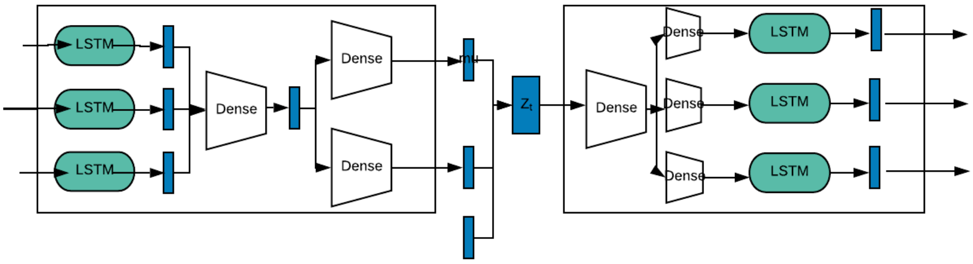

Figure 1: VAE architecture, with LSTM layer

4.2. MIDI-VAE

MIDI-VAE is the architecture of the source code, and is based on standard VAE with a hyperparameter β to weigh the KL divergence $D_{KL}(q_\theta(z|x)||p(z|x)) = \sum_z q(z|x)\log\left(\frac{q(z|x)}{p(z|x)}\right)$ in the loss function. a VAE consists of an encoder $q_\theta(z|x)$, a latent variable z and a decoder $p_\phi(x|z)$, where the latent variable z is imposed with a prior distribution $p(z) = \mathcal{N}(0, I)$. The effect of VAE rather than Autoencoder makes the interpolation of latent space valid, as the latent space distribution has better mathematical properties.
VAE Loss function contains reconstruction loss and KL divergence

$$L_{\text{VAE}} = \mathbb{E}_{q_\theta(z|x)}[\log p_\phi(x|z)] - \beta D_{KL}[(q_\theta(z|x)||p(z)], \quad (1)$$

Figure 1 refers to the MIDI-VAE architecture. As each multi-track sample music is represented as pitch, velocity and instrument rolls, a joint distribution is formed by passing these rolls through respective three encoders(implements as RNN). The output of these encoders is concatenated and passed through fully connected layers to get $q_\theta(z|x) = \mathcal{N}(\mu_z(x), \sigma_z(x))$.
A latent variable z is sampled from $\mathcal{N}(\mu_z, \sigma_z * \epsilon)$ by the reparameterization trick. The latent vector is then fed into three parallel fully connected layers. Finally, the three decoders reconstruct the pitch, velocity and instrument rolls.

The goal of the project is to construct the harmonic multi-track music, so we want to learn a joint distribution instead of three marginal distribution, and thus we chose to use three parallel encoder-decoder pairs instead of three individual autoencoders

4.3. Style Classifier

VAE has the benefit of a disentangled latent space, thus allowing modification on the latent vectors to transfer music style. As shown from the figure , attach a softmax style classifier to the first k (defined as the number of different styles in input data: $|S|$) dimensions of latent variable **z**. The loss of the classifier is added to the total loss, so it can learn a suitable k dimensions to classify the genres. Take two styles as an example: the 2 dimensions for style classifier will behave like $(1, 0)$ for a Jazz song and $(0, 1)$ for a Pop song. In order to change a song's style, we just need to swap the 2 values of the latent vector, and then reconstruct the song by decoder.

4.4. Losses

The loss of this classifier is cross entropy $H(S, \hat{S})$, and $\hat{S}$ denotes the predicted style given by style classifier. Style, instrument and pitch losses used cross entropy because they are discrete. Velocity used MSE(mean squared error) because it's continuous. The total loss function is

$$\begin{aligned} L_{tot} =& \lambda_P H(P, \hat{P}) + \lambda_I H(I, \hat{I}) \quad (2) \\ &+ \lambda_V MSE(V, \hat{V}) + \lambda_S H(S, \hat{S}) - \beta D_{KL}(q||p) \quad (3) \end{aligned}$$

5. Implementation

5.1. Dataset and Prepocessing

Our dataset is collected from different MIDI sources, and it contains genre Classic, Jazz, and Pop etc., and the details about our dataset can be seen in the table below.

Data Set	# of songs	# of bars
Jazz	554	72190
Pop	659	65697
Classic	477	60523

We use a train/test split of 90/10. We select number of instrument track to be 4 from each song by picking the tracks with the highest number of played notes, and for each track we chose the highest and soundest voice, meaning picking the highest notes per time step. We chose the 16_{th} note as smallest unit we consider, and we consider the 4/4 time signature as our standard time signature. In a 4/4 time signature, there are 4 beats in a bar, and for each beat it can contain 1 4_{th} note, 2 8_{th} note, and 4 16_{th} note. 91% of Jazz and Pop songs in our dataset are in 4/4 , whereas for Classic the fraction is 34%. For songs with time signatures other than 4/4 we still designate 16 16_{th} notes as one bar. During training we shuffle the songs for each epoch, but keep the bars of a song in the right order.

5.2. Parameters and settings

The loss function weights in equation (2) $\lambda_P, \lambda_V, \lambda_I$ and λ_S were set to 1.0, 1.0, 0.1 and 0.1 respectively. λ_P was set to 1.0 to favor high quality note pitch reconstructions over the rest.V was also set to 1, because the MSE magnitude is much smaller than the cross entropy loss values. $\beta = 0.1$ should be small because high β would increase the losses significantly.

Use Gated Recurrent Units (GRUs) as RNN rather than LSTMs can increase the performances dramatically. Most of the layers will behave better if using $\tanh$ than sigmoid.

5.3. Interpolation

MIDI-VAE is able to produce interpolations. Merge the songs together by linearly interpolating the latent vectors. The result of this medley vector shows smooth variation between songs. As an example, Figure 2 shows the reconstructed pitch for a medley of two songs. Red and blue bars are the pitches of two original sons, while black bars are the shared pitches. The original pitches are identifiable when listening to the samples, and we can see that the bridge of different pieces of the music is musically consistent.

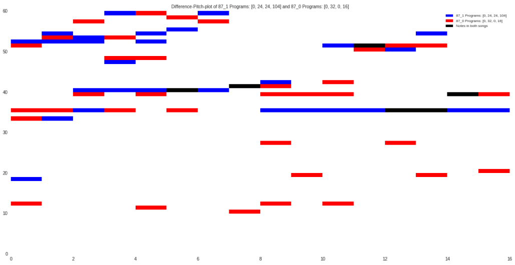

Figure 2: Pitch of a medley song from two different songs(red and blue). Black is where the two songs share the same pitch.

6. Experiment Result

6.1. Samples

Whitney Houston - My Love Is Your Love (Pop to Jazz)
Medley song from 4 original songs

6.2. Training Loss

The training loss is shown in figure 3. We only show the fist 100 iterations because a lack of computation time. We can see that the training loss are decreasing, while the training accuracy and KL loss are increasing. The total loss is bad when KL divergence is too large.

7. Conclusions

The work of Brunner's paper [1] is mostly reproduced. An effective model to transfer music style, MIDI-VAE is presented by using pitch, velocity and instruments. The interpolated songs show pitch similarity to the original songs and can connect medley pieces in musically consistent way.
If we had more time, we would apply the weight annealing trick in the KL divergence layer, which can make the results rely more on the latent space rather than the hidden states of RNN. At the same time, we would also do more evaluation work on the latent space and the effects on different styles.

8. Acknowledgements

Thanks Bowen for the help of understanding the architecture of the paper.

9. References

losses.png losses.bb
1552406356 9136615

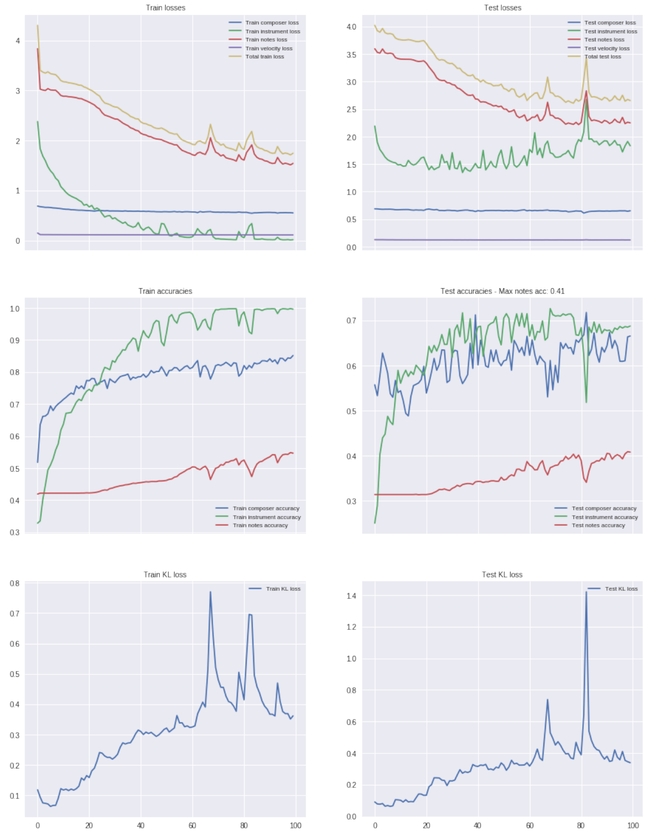

Figure 3: Losses and accuracy of first 100 iterations.

[1] Gino Brunner *MIDI-VAE: Modeling Dynamics And Instrumentation of Music With Application to Style Transfer*. ISMIR 2018, Paris, France.

[2] Ramon Lopez de Mantaras and Josep Lluis Arcos *Ai and music from composition to expressive performance*. AI Mag., 23(3):4357, September 2002. ISSN 0738-4602.

[3] Daniel P.W. Ellis, Graham E. Poliner, *Identifying cover songs with chroma features and dynamic programming beat tracking,* 2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP 07.

[4] Diederik P. Kingma, Max Welling. *Auto-encoding variational bayes*, CoRR, abs/1312.6114, 2013.

[5] Midi association, the official midi specifications. https://www.midi.org/specifications. Accessed: 01-06-2018.

[6] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 2414–2423. IEEE, 2016.

[7] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6309–6318, 2017.

[8] Iman Malik and Carl Henrik Ek. Neural translation of musical style. *CoRR*, abs/1708.03535, 2017.