Automatic Chinese Traditional Melody Generation by Transfer Learning Techniques

Jincheng Liang@jl12003, Shaomin Xu@sx2311, Taishan Zhao@tz2516

Abstract

In this paper, we present the transfer learning approach for automatic traditional Chinese music melody generation. Traditional Chinese music composition is one of the most magnificent cultural achievements throughout the history of ancient China. Yet, the lack of scientific recording method and professional musical education among the majorities resulted in music pieces that are either incomplete, or inaccurate. With a limited amount of sample, it would be challenging to reproduce these compositions. Transfer learning is an effective approach to tackle the problem of data insufficiency, by transferring the trained common features from one domain to another. To evaluate the effectiveness of transfer learning method for traditional Chinese music melody generators, we constructed and trained a recurrent variational autoencoder as the generative model, and selected a genre-unspecified MIDI dataset as our source dataset and a traditional-Chinese-music-only dataset, ChMusic (Gong et al. 2021), as our target dataset. We implemented the multi-task method, which introduces an additional genre label as a conditional variable so as to train the model together on the source and target datasets simultaneously, and trained a genre classifier to further improve the melody generator. The evaluated results proved the consistency and the performance of the generator to be excellent even when compared to the generator introduced in the referred literature. The project codebase and trained models can be accessed at https://github.com/JinchengLiang/DLproject

Introduction

Deep learning has emerged as a powerful tool for music technology industry, allowing not only music classifications and sound wave processing, but also creation of completely new pieces by learning and reproducing existing music compositions. This is most frequently done by firstly training a deep learning model with a large, structured, machine-readable dataset of music compositions, mostly in formats such as MIDI files, then implement deep neural networks such as generative adversarial network (GAN) (I. J. Goodfellow and Bengio 2014) or variational autoencoder (VAE)(Kingma and Welling 2013) to learn and compose new pieces based on the provided dataset. A great example for such datasets is the Lakh MIDI dataset (LMD) introduced by Colin Raffel (Raffel 2016) is a public collection of 176,581 MIDI files. The original goal of this collection was to facilitate large-scale music information retrieval. This collection is not only enormous, but also diverse. At the same time, it contains only MIDI files, which greatly reduces the workload of data processing.

As previous studies shown, the main advantage of deep neural network method for music generation compared to traditional algorithm-based approaches, comes from the unmatched ability of deep learning models to find and learn from unnoticeable patterns from big data. There are existing deep learning models which return promising results for generating music in certain genres such as Rock, Pop, and Classical, thanks to the sufficient data and availability of MIDI resources of these genres.

However, the scenario is different when it comes to World music, especially traditional, historical music compositions from different cultural background. Due to the lack of dataset availability and data consistency, training a efficient and promising deep learning model can be challenging. A great example would be traditional Chinese music. Because of how ancient they are, there wasn't a scientific method to record them, and transcribing by hand could sometimes be inconsistent and inaccurate. As a result, music sheet from ancient China is so scarce, not to mention MIDI files. Consequently, a successful traditional Chinese musical melody generator has not been introduced.

The motivation and main objective of this project is to construct and train a generative model that successfully composes traditional Chinese musical melodies. To tackle the data scarcity issue, we implemented transfer learning technique. The main idea is to train the model with a source task in which the dataset is sufficient, then adapt that model for a target task in which the dataset is insufficient. There are different methods of transfer learning, such as fine-tuning and multi-tasking. Based on the experimental results provided in (Hung et al. 2019), we decided to proceed with a multi-tasking approach and construct a VAE model which trains the model on the source and target datasets simultaneously with a separately musical genre classifier for maximum performance.

While a traditional Chinese music composition may consist multiple elements, such as instruments or chords, we are only focusing on melody generation in this project, and

"unconditioned" melody generation only, which means no conditions or information such as tempo or mood would be given prior to melody generation. The performance of our model was evaluated based on the work of Yang and Lerch (Yang and Lerch 2020), where nine different features would be collected for quantitative evaluation.

Related Work

A generative model, JazzGen, that can generate specifically Jazz melody with few data by training a VAE model with transfer learning approach, was introduced in (Hung et al. 2019). The research focused on investigating which transfer learning technique is more effective for musical melody generation and how much does a transfer learning method benefits from increasing the size of the source domain dataset. For this research, the author collected a genre-unspecified dataset that contained 11,329 melody phrases from a music theory forum called TheoryTab as the source domain dataset, and a clean, Jazz-melody-only collection containing 575 four-bar melody phrases composed by a well-trained Jazz composer, as well as 240 unique Jazz songs from the Jazz Realbook, as the target domain dataset. Two different methods were implemented and the performances were compared and evaluated. The first method was to pre-train the VAE model with the source dataset then fine-tune the model on the target dataset. The second method introduced an additional genre label as a conditional variable, so as to train the model together on the source and target datasets, with a separately trained genre classifier for further performance improvement. To study the influence of increasing the scale of source domain dataset, six different levels of source-to-target data ratios were considered and experimented.

The feature matrix for this research was based on the study of Yang and Lerch (Yang and Lerch 2020), which describes two different aspects of music, including pitch and rhythm. After extracting features, the Overlapping Area (OA) concept was implemented to measure the performance of the model. More details about the calculation of OA will be introduced in the Evaluation section of this paper, including obtaining the Probability Distribution Function (PDF) and Kullback-Leibler Divergence (KLD).

The experimental results of this research stated the second method (multi-tasking) improves the performance of the base model, which trained only on the Jazz dataset, slightly better than the first method (fine-tuning) did. On the other hand, the change in source-to-target data ratio did not have a significant impact on the overall performance of the model.

Our project serves as an extension of this research. We adopted the VAE architecture of JazzGen and modified certain existing programs within the code-base for our traditional Chinese music generator. As suggested by this research, we proceeded with the multi-tasking transfer learning method for a promising performance of our model. In our case, the target domain dataset, ChMusic, is even smaller than the Jazz dataset, and the source-to-target data ratio is significantly larger. We expect such differences between our situations may require us to train our model for more epochs for a higher accuracy. Nonetheless, this research is a perfect reference of our work and a great explanation of our objectives.

Datasets

For this project, we used a genre-unspecified collection called Lakh MIDI dataset as our source domain dataset, and a Chinese traditional music (CTM) collection, ChMusic, as our target domain dataset. The scale and information of these two dataset are summarized in Table 1.

	Lakh MIDI	ChMusic
Genre	Unspecified	CTM
Track	Melody, Chord, Drum	Melody
Time Signature	4/4	4/4
Number of Phrases	176,581	288
Number of Bars	1,412,648	2304

Table 1: Summary of the Two Datasets

The Lakh MIDI dataset is a collection of 176,581 unique MIDI files, 45,129 of which have been matched and aligned to entries in the Million Song Dataset. The MIDI files it consists have multiple tracks, each representing a unique instrument. Only one of the tracks represents the melody, and the rest are either playing chords, or drums. In most of the cases, the track labelled as "vocal" are recognized as the melody, but there are also outliers within these files. Therefore, we programmed a data processor to filter out the real melody track, by first taking out the track which is labelled as drum, and the track with the fewest notes and pitches, which is mostly the bass line. To differentiate the melody track from the chord tracks, we took the advantage of the feature of chords, which have a highly recognizable repetitive pattern and less pitch count than the melody track. Once we filtered out the excess tracks, we divided the melody into phrases, where a phrase was defined as a eight-bar segment.

Our target dataset, the ChMusic collection, consists samples of traditional Chinese melodies played by eleven different instruments, each instrument has five music samples. Therefore, The total number of musical compositions in this dataset is fifty-five, stored as music (.wav) files. To make the data samples machine friendly, we first used a MIDI processor, Basic Pitch, to transform these .wav files into MIDI files, then follow the same data processing procedures as the source dataset.

To illustrate a melody phrase computationally, we adopted the step-based method and represented each eight-bar melody phrase as a pianoroll (.npy file), a matrix-like data structure generated by the Numpy module. The horizontal axis denotes time steps, and the vertical axis denotes frequency. Both features can be extracted from the MIDI file. For each bar, we set the range of the matrix to 48, in other words, considering MIDI notes from C3 to B6, and the width (time resolution) to 16 (i.e., 16 time steps per bar, or equivalently 4 time steps per beat in 4/4 time signature). Therefore, the size of the target output tensor for melody generation model is 8 (bars) × 16 (time steps) × 48 (MIDI notes) × 1 (track).

Methodology

Model Architecture

In this particular approach, a recurrent VAE model is utilized. The encoder component takes four-bar melody sequences as input and employs bidirectional gated recurrent units (BGRU) to capture the interdependencies among the bars. The outputs of all GRU time steps are combined by concatenation and further processed through several dense (fully-connected) layers to obtain the embedding vector. Essentially, given an observed input melody x, the encoder E_θ with parameter set θ encodes x into a latent vector $z = E_\theta(x)$. Besides, we can also choose the popular transformer(Vaswani et al. 2017) as the encoder and the decoder.

In the decoder component, a latent vector z is sampled from a normal distribution with mean μ and standard deviation δ. The sampled vector is then fed through several fully-connected layers parameterized by ϕ to generate the initial states of the melody. These initial states are subsequently inputted into a unidirectional GRU, followed by a sigmoid activation layer, to produce a four-bar pianoroll as the final output.

Multitask Learning

In Figure 1, we combine a one-hot genre label y with the latent vector z. The model is trained simultaneously on LMD and ChMusic, treating them as distinct tasks. For the Chinese traditional music (CTM) dataset ChMusic, the label y is set as $[0, 1]$, while for LMD, the label y is set as $[1, 0]$.

To assess whether our model effectively generates melodies of different types, we employ a pre-trained genre classifier $C()$ to determine if the output melody contains Jazz elements. The classifier has a similar structure to the VAE encoder, but with a modified output layer size of 1. When a generated melody is evaluated by the genre classifier, it produces the probability of being CTM. We utilize a sigmoid activation on the final layer's output neuron and optimize the classifier using cross-entropy loss. The training objective is to assign a probability of 1 for CTM and 0 for non-CTM melodies. Consequently, once the VAE model generates an output melody, it undergoes evaluation through this genre classifier.

The objective function for optimizing the recurrent VAE model trained using multitask learning based transfer learning includes the genre prediction loss, denoted as $L_{genre}(\hat{y}, y)$, which is added to the overall objective function. Hence, the objective function for the model is as follows:

$$L(\theta, \phi, x, y) = L_{recon}(x, y) + L_{lat}(x) + L_{genre}(\hat{y}, y)$$

where $L_{recon} = E_{q_\phi(z|x)}[log\ p_\phi(x|z)]$is the reconstruction term, and $L_{lat} = -D_{KL}(q_\phi(z|x)\ ||\ p(z))$ regularizes the encoder to align the approximate posterior $q_\phi(z|x)$ with the prior distribution $p(z) \cdot p_\phi(x|z)$ is the data likelihood.

Evaluation

To explore the advantages of transfer learning methods with larger source domain training data, we analyze six levels of source-to-target data ratios (R):

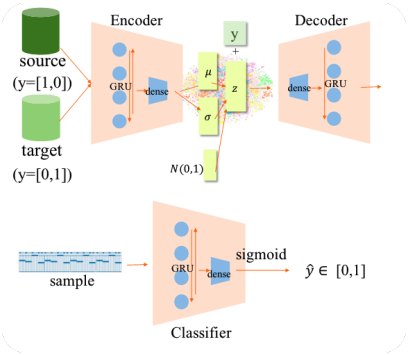

Figure 1: model

$$R = \frac{number\ of\ non-CTM\ training\ phrases}{number\ of\ CTM\ training\ phrases}$$

Feature Metrics

To assess the quality of a generated melody, various features are derived based on the research conducted by Yang and Lerch (Yang and Lerch 2020). These features capture two key aspects of music: pitch and rhythm.

The pitch-related features consist of the following five descriptors, which indicate preferences in pitch arrangement:

- **Pitch count (PC)**: The number of different pitches within a sample. The output is a scalar for each sample.
- **Pitch class histogram (PCH)**: The pitch class histogram is an octave-independent representation of the pitch content with a dimensionality of 12 for a chromatic scale. In our case, it represents the octave-independent chromatic quantization of the frequency continuum.
- **Pitch class transition matrix (PCTM)**: The transition of pitch classes contains useful information for tasks such as key detection, chord recognition, or genre pattern recognition. The two-dimensional pitch class transition matrix is a histogram-like representation computed by counting the pitch transitions for each (ordered) pair of notes. The resulting feature dimensionality is 12×12.
- **Pitch range (PR)**: The pitch range is calculated by subtraction of the highest and lowest used pitch in semitones. The output is a scalar for each sample.
- **Average pitch interval (PI)**: Average value of the interval between two consecutive pitches in semitones. The output is a scalar for each sample.

On the other hand, the rhythm-related features focus on the arrangement of notes within the melody and include the following four descriptors:

- **Note count (NC)**: The number of used notes. As opposed to the pitch count, the note count does not contain pitch information but is a rhythm-related feature. The output is a scalar for each sample.
- **Average inter-onset-interval (IOI)**: To calculate the inter-onset-interval in the symbolic music domain, we

find the time between two consecutive notes. The output is a scalar in seconds for each sample.

- **Note length histogram (NLH)**: To extract the note length histogram, we first define a set of allowable beat length classes [full, half, quarter, 8th, 16th, dot half, dot quarter, dot 8th, dot 16th, half note triplet, quarter note triplet, 8th note triplet]. The rest option, when activated, will double the vector size to represent the same lengths for rests. The classification of each event is performed by dividing the basic unit into the length of (barlength) / 96, and each note length is quantized to the closest length category. The output vector has a length of either 12 or 24, respectively.
- **Note length transition matrix (NLTM)**: Similar to the pitch class transition matrix, the note length transition matrix provides useful information for rhythm description. The output feature dimension is 12×12 or 24×24, respectively.

Overlapping Area

In their work, Yang and Lerch (Yang and Lerch 2020) introduced the Overlapping Area (OA) as an evaluation measure. The rationale behind this approach is as follows: when comparing different sets of outputs, it is often more advantageous to employ relative measurements rather than relying solely on the mean values of individual features. By utilizing relative measurements, a more comprehensive understanding of the dataset's diversity can be obtained.

OA is defined as the common area under two probability density functions, and it is a measure of similarity between two populations. The value of OA ranges from 0 to 1, where a value 0 indicates that there is no overlap and a value 1 shows that the two populations are identical. It has wide applications as a similarity measure.

Let $f_1(x)$ and $f_2(x)$ be the probability density functions of two continuous populations. The OA can be defined as

$$OA = \int min\{f_1(x), f_2(x)\}\, dx$$

The Kullback-Leibler Divergence (KLD) is a widely employed method for comparing two distributions. However, when dealing with discrete probability distributions, the KLD is calculated element-wise, which can lead to limited sensitivity in cases where the distributions have identical shapes (as indicated by similar Kurtosis and Skewness), but differ primarily in their mean values or x-axis shifts. In such situations, the KLD may fail to detect significant differences. On the other hand, OA metric can effectively highlight these discrepancies and provide a more informative assessment of dissimilarities between such distributions.

Results

Overlapping Area

Table 2 and Table 3 show the OAs of the model using GRU and transformer as encoder and decoder evaluated on different features under six levels of source-to-target data ratio(R). The bold number indicates the hightest OA for each feature under different R.

For the model using the GRU, when the R is higher, most of the features performs better except for the PR and PCH/bar. The model using the transformer has a similar performance as the GRU one in the features related to notes. However, it performs worse in the pitch features. One possible reason may be the melody depends more on adjacent pitches, so the GRU does better.

	R=2	R=3	R=4	R=5	R=6	R=7	R=8
NC	0.64	0.62	0.61	0.59	0.59	**0.68**	0.56
NC/bar	0.66	0.68	0.67	0.63	0.65	**0.69**	0.58
NLH	0.76	0.75	0.72	0.76	0.73	0.72	**0.77**
NLTM	0.62	0.60	0.61	0.62	0.61	**0.63**	0.56
IOI	0.73	0.72	0.74	0.74	0.78	0.79	**0.79**
PC	0.60	0.53	0.41	0.48	0.44	**0.65**	0.43
PC/bar	0.61	0.53	0.51	0.52	0.50	**0.62**	0.42
PR	**0.78**	0.71	0.70	0.73	0.66	0.76	0.63
PCH	0.83	0.80	0.76	0.78	0.75	0.74	**0.83**
PCH/bar	**0.74**	0.69	0.71	0.70	0.70	0.69	0.63
PCTM	0.79	0.80	0.78	0.82	0.77	0.81	**0.82**
PI	0.75	0.74	0.75	0.77	0.78	0.72	**0.80**

Table 2: OA between generated melodies and target dataset when using GRU as encoder and decoder

	R=2	R=3	R=4	R=5	R=6	R=7	R=8
NC	0.62	0.67	0.61	0.70	0.64	**0.72**	0.70
NC/bar	0.69	0.63	0.58	0.61	0.71	**0.75**	0.63
NLH	0.70	**0.71**	0.63	0.68	0.65	0.70	0.66
NLTM	0.61	0.62	0.60	0.64	0.63	0.66	**0.70**
IOI	0.62	0.57	0.54	0.60	0.64	**0.67**	0.65
PC	0.28	0.32	0.34	0.33	0.37	**0.41**	0.32
PC/bar	0.31	0.34	0.30	0.39	**0.43**	0.38	0.40
PR	0.35	0.37	0.32	0.30	0.38	0.31	**0.40**
PCH	0.50	0.52	0.48	0.53	0.54	0.51	**0.59**
PCH/bar	0.54	0.60	0.53	**0.67**	0.59	0.57	0.61
PCTM	0.49	0.47	0.41	0.39	0.47	**0.49**	0.43
PI	0.77	0.75	0.69	0.68	0.73	**0.80**	0.78

Table 3: OA between generated melodies and target dataset when using transformer as encoder and decoder

Pitch-related Analysis

In order to take a closer look to how these models manage the pitches, Figure 2 compares the pitch class histograms of the target ChMusic dataset and the melodies generate by the model using GRU with R = 8. We can see that the probility of presence of evry pitch form C to B are close, especially E, F#, Ab, Bb and B. This means that the model learns the probability of the presence of a scale in Chinese traditional music. In other words, our model does work.

Conclusion

Based on statistical evaluation, the objective of building traditional Chinese melody generator has been met. Future ex-

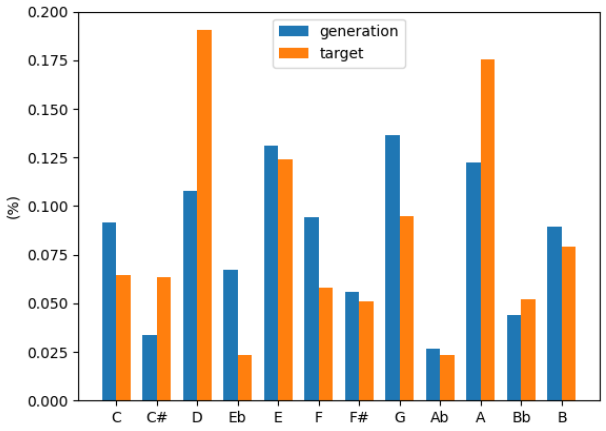

Figure 2: PCH of generated melodies and target dataset

tension on this project may include adding the characteristics of traditional Chinese instruments to the generative model using WaveNet (van den Oord et al. 2016). There are several reasons we did not include this feature within this project. First of all, the conference code was based on python2, some methods in which were discard. Therefore, we decided to update version to python3 to make the code be available to more researchers in the future and this task takes lots of time. Secondly, we had to spend plenty of time dealing with datasets. To be more specific, we were required to filter the damaged MIDI files and cut out the most important and information contained fragment of each good MIDI files in Lakh MIDI dataset. Overall, the task was successful, and the objectives has been completed.

References

Gong, X.; Zhu, Y.; Zhu, H.; and Wei, H. 2021. ChMusic: A Traditional Chinese Music Dataset for Evaluation of Instrument Recognition. arXiv:2108.08470.

Hung, H.-T.; Wang, C.-Y.; Yang, Y.-H.; and Wang, H.-M. 2019. Improving Automatic Jazz Melody Generation by Transfer Learning Techniques. arXiv:1908.09484.

I. J. Goodfellow, M. M. B. X. D. W.-F. S. O. A. C., J. Pouget-Abadie; and Bengio, Y. 2014. Generative adversarial nets. In *Proc. Advances in Neural Information Processing Systems*, 2672–2680. .

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. arXiv:1312.6114.

Raffel, C. 2016. *Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching*. Ph.D. diss., School of Arts and Sciences, Columbia Univ.

van den Oord, A.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A.; and Kavukcuoglu, K. 2016. WaveNet: A Generative Model for Raw Audio. arXiv:1609.03499.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention Is All You Need. arXiv:1706.03762.

Yang, L.-C.; and Lerch, A. 2020. On the evaluation of generative models in music. *Neural Comput Applic*.