LUPINE PUBLISHERS

Current Trends in Computer Sciences & Applications

DOI: 10.32474/CTCSA.2024.03.000161

ISSN: 2643-6744

Research Article

Machine Learning and Deep Learning Techniques in Music Style Generation: Trends and Challenges

Fatemeh Jamshidi[1]*, Maryam Bigonah[1], Daniela Marghitu[1] and Richard Chapman[1]

[1]*Computer Science and Software Engineering Auburn University, USA*

***Corresponding author:** Fatemeh Jamshidi, Computer Science and Software Engineering Auburn University, USA

Received: May 20, 2024 **Published:** May 24, 2024

Abstract

The recent advances in image style transfer have spurred significant interest in applying similar deep learning techniques to music style transfer. In this paper, we present a hierarchical decoder framework, where the encoder compresses variations in high-dimensional datasets into a lower-dimensional code, and the decoder expands these variations to generate an output. We introduce Music-VAE, which addresses the challenge of modeling long-term musical structures. Our approach employs a novel sequential auto-encoder with a hierarchical decoder, where classical music intended for interpolation with jazz is processed through the encoder to inform the generation of the output. Additionally, we introduce Melody RNN, which is capable of extracting musical style features to generate music in specific styles. To evaluate the effectiveness of our method, we propose an online system based on the Turing test, allowing users to compare the generated music with the original input to assess the quality and accuracy of style transfer.

Introduction and Motivation

Deep generative models are powerful neural network architectures that replicate data by estimating the probability distribution of existing data points. This capability enables the generation of "fake- but-realistic" data from learned distributions. Recent advancements have demonstrated success in generating realistic images with millions of pixels and audio with hundreds of thousands of timestamps. A variety of models have been developed, including Auto-Encoders, with recent breakthroughs in deep learning addressing music-related challenges using Recurrent Neural Networks (RNNs) and Variational Auto-Encoders (VAEs). In this paper, we primarily focus on RNN models and deep latent variable models. VAEs, in particular, are a deep learning technique for learning latent representations by modeling data with a directed latent-variable structure.

We introduce novel methods leveraging these models to advance the state-of-the-art in music style transfer, highlighting the effectiveness of hierarchical decoders and sequential auto-encoders in generating musically coherent outputs. Our approach underscores the potential of deep generative models in transforming the landscape of music generation and style transfer. Music Style Generation (Spring 2024),

$$p(x,z) = p(x \mid z)\,p(z) \tag{1}$$

The advantage of these models is that they indirectly model both $p(x \mid z)$ and $p(z)$, where z is the latent vector. Like regular autoencoders, VAEs compress relevant information about the input into a lower-dimensional latent code. They consist of an encoder $q_{\lambda}(z \mid x)$, which approximates the posterior $p(z \mid x)$, and a decoder $p_{\vartheta}(x \mid z)$, which parameterizes the likelihood $p(x \mid z)$.

Most of the deep learning community has remained focused on fixed-dimensionality domains, such as images. Until recently, little attention had been given to adapting these ideas to music. Although VAEs have shown success in latent representation of short sequences of natural data, they have yet to be successfully applied to long-term sequences. To address this problem, we introduce a novel sequential auto-encoder with a hierarchical recurrent decoder. In this paper, we focus on generating classical or jazz-style music. Popular classical music exhibits strong long-term structures, such as repetition and variation between measures and pieces. First, songs are divided into sections, which are further broken down into measures and beats. Additionally, music often involves multiple players generating music with strong inter-player dependencies. These unique characteristics make the hierarchical vector model ideal for our application.

The remainder of the paper is structured as follows: In Section 2, we review related work in this area. Section 3 describes our data model and methodology. Section 4 presents our results, and Section 5 concludes the paper.

Related Work

The domain of music style transfer leverages deep learning techniques to alter the stylistic attributes of musical pieces while preserving their core content. This technology has been inspired by the successes in image style transfer and has since evolved to address the unique challenges posed by the multi-modal and hierarchical nature of music representation. This literature review explores the historical context, recent advancements, challenges, and potential applications of music style transfer.

Historical context and early developments

Bharucha and Todd pioneered the use of Recurrent Neural Networks (RNNs) for generating music in the style of Bach as early as 1989, setting a foundational precedent for neural network applications in music creation Bharucha and Todd (1989) [1]. Since then, neural networks have steadily gained traction in this field, evolving to address various facets of music generation and style transfer.

Key models and methodologies

MusicVAE:

a) **Description**: MusicVAE is a recurrent Variational Autoencoder (VAE) with a hierarchical decoder designed to enhance sampling, interpolation, and reconstruction of musical sequences.

b) **Significance**: It addresses the challenge of modeling long-term musical structures, which are crucial for maintaining the coherence of generated music.

c) **Performance**: Qualitative and quantitative experiments have demonstrated MusicVAE's superior performance in generating stylistically coherent music compared to other models Engel et al. (2018) [2].

ToneNet

a) **Description:** ToneNet integrates three different models, with a sequence-to-sequence model as the baseline, compared against VAE-GAN and Seq-GAN architectures.

b) **Findings**: Seq-GAN achieved remarkable results due to the effective feedback mechanisms from the discriminator to the generator's LSTMs, whereas VAE-GAN underperformed due to oversimplified assumptions about music representation Malik and Ek (2017) [3].

StyleNet

a) **Description:** This model uses Long Short-Term Memory Networks (LSTM) to generate music performances indistinguishable from human performances.

b) **Challenges:** While effective, StyleNet's reliance on note velocity as the sole stylistic feature is insufficient for capturing the full spectrum of musical style, resulting in generated music that closely mimics but slightly deviates from the original Malik and Ek (2017) [3].

Recent advances and emerging trends

Hierarchical and multi-level representations

a) **Advancements:** The introduction of hierarchical decoders and multi-level representations has improved the ability of models to handle the complex structure of music, from beats and measures to entire compositions Dai et al. (2018) [4].

b) **Impact:** These methods allow for more nuanced and accurate style transfers, capturing the intricacies of different musical styles.

Transformer models and positional encodings

a) **Innovations:** The use of Transformer models with innovative positional encodings, such as stochastic positional encoding (SPE), has shown promise in better extrapolating musical sequences beyond training lengths, enhancing the generation of coherent and stylistically accurate music Cífka (2021) [5].

Self-supervised learning and disentanglement techniques

a) **Approaches:** Techniques such as vector-quantized variational autoencoders (VQ-VAE) with self- supervised learning strategies have been developed to disentangle timbre and pitch, allowing for more precise timbre transfer applications Cífka (2021) [5].

Challenges in music style transfer

Defining music style

a) **Complexity:** Music style is a multi-dimensional concept encompassing timbre, performance, and composition, making it challenging to model and transfer accurately Dai et al. (2018) [4].

b) **Disentanglement:** Successfully separating content and style remains a significant hurdle, requiring advanced techniques to ensure meaningful style transfer without content loss.

Long-term structure modeling

a) **Importance:** Long-term dependencies in music, such as motifs and themes, are crucial for maintaining coherence but are difficult for models to capture effectively Cífka (2021) [5].

b) **Solutions:** Hierarchical and recurrent models have been proposed to address this, though further refinement is needed to achieve consistent results.

Evaluation metrics

a) **Limitations:** Developing objective metrics for evaluating style transfer quality is challenging due to the subjective nature of music perception Dai et al. (2018) [4].

b) **Progress:** Efforts are ongoing to create standardized evaluation protocols that can more accurately assess the fidelity and creativity of generated music Cífka (2021) [5].

Applications of music style transfer

Creative tools for artists

a) **Co-creation:** Music style transfer models serve as collaborative tools for artists, allowing them to explore new creative directions by transforming existing pieces or generating new ones from scratch Dai et al. (2018) [4].

b) **Innovation:** These tools can inspire novel compositions and remixes, broadening the scope of musical expression.

Automated music generation for media

a) **Utility:** Automated systems can generate background music for videos, advertisements, and games, tailored to specific stylistic requirements, thereby reducing production time and costs Cífka (2021) [5].

b) **Customization:** These systems can personalize music to match user preferences, enhancing the consumer experience.

Music education and analysis

a) **Pedagogical Use:** Music style transfer models can be used in educational settings to demonstrate the characteristics of different musical styles and aid in music analysis and composition studies Dai et al. (2018) [4].

Music style transfer is a burgeoning field within deep learning, characterized by rapid advancements and significant challenges. While substantial progress has been made in modeling and transferring musical styles, ongoing research is required to refine these models and address existing limitations. The potential applications of music style transfer are vast, spanning creative, commercial, and educational domains, underscoring its importance and impact on the future of music technology. In the next section, we will use Music-VAE and Melody RNN models to extract the features of Jazz and Classic music style.

Methodology

Dataset

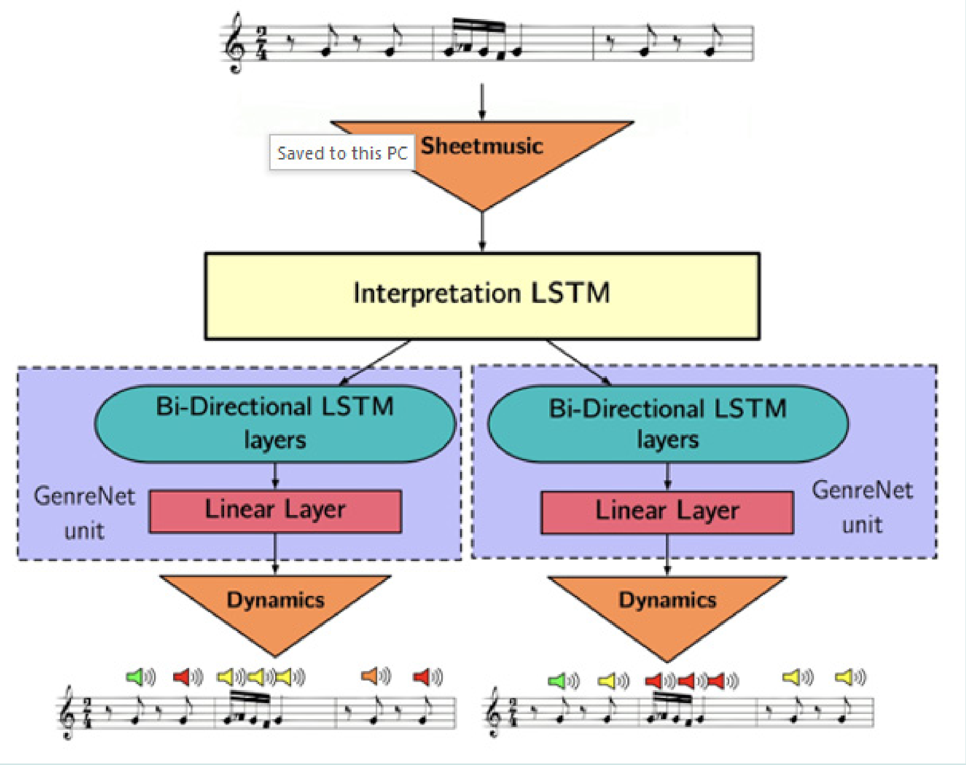

Figure 1: Schematic diagram of Style Net.

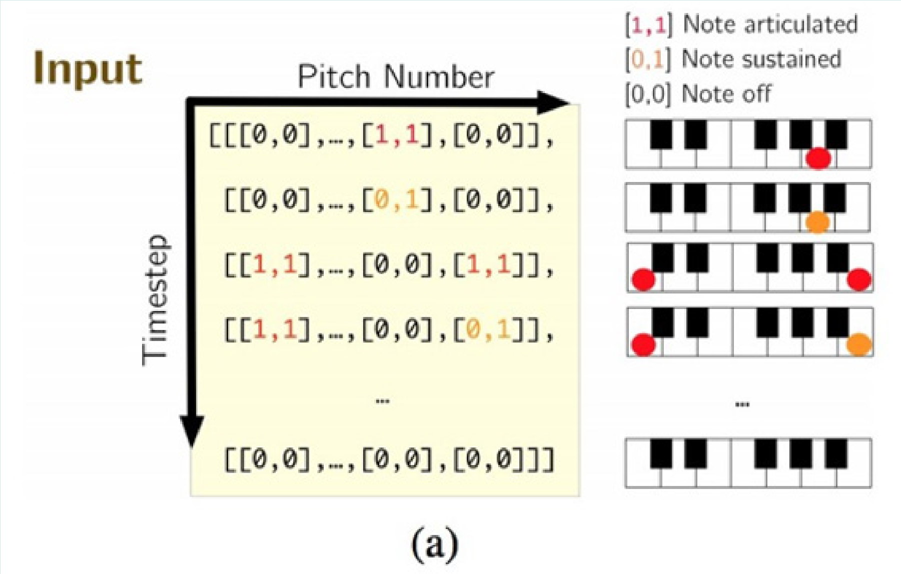

Figure 2: Data Representation.

The dataset used in this study has been taken from the paper “Neural Translation of Musical Style” Malik and Ek (2017) [3]. This dataset comprises approximately 350 piano-only Jazz and 350 classical songs in MIDI format, with the average length of the training data being about 4 minutes. The primary objective is to train models to learn and generate music in the jazz style. During training, the MIDI files are converted to Note Sequences, a data format that is faster and simpler to use during modeling than MIDI files Mag. The input MIDI file is encoded into a P matrix, where T is the number of time steps in the song and P is the number of pitches in the instrument (for example, a piano with 88 keys has 88 pitches). Each value in the matrix encodes note events using a 2-D vector for each pitch: [1 1] for note on, [0 1] for note sustained, and [0 0] for note off, as shown in (Figure1, 2) Malik and Ek (2017) [3].

To facilitate training, each file in the dataset has been quantized to align with a particular time interval (4/4time signature) and formatted to MIDI format 0. The dataset can be accessed from: https://medium.com/@suraj.jayakumar/ tonenet-a-musical-style-transfer-c0a18903c910

Comparison of existing datasets for music style transfer

Table 1 provides a comparison of existing datasets commonly used for music style transfer. Each dataset varies in the genres it covers, the format of the music files, the number of songs included, and the average length of the pieces. This comparison helps highlight the diversity and scope of datasets available for training and evaluating music style transfer models.

Table 1: provides a comparison of existing datasets commonly used for music style transfer. Each dataset varies in the genres it covers, the format of the music files, the number of songs included, and the average length of the pieces. This comparison helps highlight the diversity and scope of datasets available for training and evaluating music style transfer models.

Dataset	Genres	Format	Number of Songs	Average Length
Neural Trans-lation of Musical Style Malik and Ek (2017)	Jazz, Classical	MIDI	700 (350 each)	4 minutes
MusicNet Thickstun et al. (2017)	Various (Clas-sical Focus)	MIDI, Audio	330	Varies (5-10 minutes)
Lakh MIDI Dataset Raffel and Ellis (2016)	Various	MIDI	176,581	Varies
MAESTRO Hawthorne et al. (2018)	Classical Piano	MIDI, Audio	1,184	5-10 minutes
NES-MDB Donahue et al. (2018)	Chiptune/8-bit	MIDI, Audio	497	Varies
JSB Chorales Allan and Williams (2005)	Bach Chorales	MIDI	382	1-2 minutes

Melody-RNN

The Recurrent Neural Network (RNN) has proven to be an effective model for predicting and generating sequential data, displaying dynamic behavior in sequences. This paper employs two methods from Magenta’s Melody-RNN: the basic RNN and the lookback RNN Waite. Both utilize Long Short-Term Memory (LSTM) architecture as the RNN cell to handle longer sequences, but they differ in the format of inputs and labels, allowing melodies to be encoded in different representational formats.

In the basic RNN, the input is a vector representing the previous event, and the label is a vector representing the next event. The lookback RNN, however, incorporates additional events to provide more context from previous melodies and the current position within measures (assuming a 4/4-time signature). This allows the model to understand event patterns and identify repetitive musical properties more easily, such as mirrored melodies Waite. Specifically, the input includes the current event, events from the previous 1 and 2 bars, a binary representation indicating whether the current pattern repeats from previous bars and beats.

To provide extra information, the labels also consider two events: binary representations of whether the melody is repeating from 1 bar ago and whether it is repeating from 2 bars ago. If the current melody is repeating from previous bars, its label will be set to the same label as the previous bars. These additional labels help reduce the complexity of the training model by aiding the learning of musical patterns. The output consists of two probabilities: one for each note that is chosen to be played and one for the note's continuity when it is on. The Beam Search algorithm is then used to find the sequence of notes with the highest probability, resulting in the musical generation. The framework of Melody RNN is shown in (Figure 3).

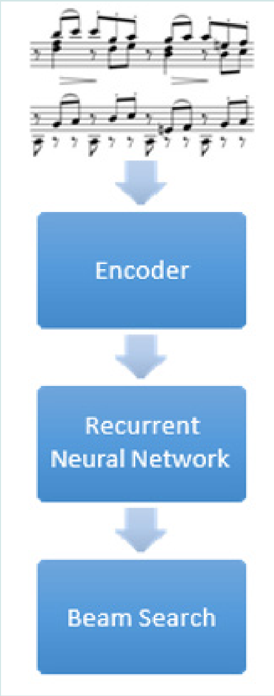

Figure 3: Melody RNN.

Music-VAE

In Music-VAE, we have used a recurrent encoder and decoder same as the model used in Sketch RNN (Ha and Eck (2017) [6]. Generally, the encoder $q(z|x)$ is a recurrent neural network, that takes $x=\{x_1,x_2,x_3,....x_t\}$ as input sequence and produces hidden states $h_1,h_2,h_3,....h_t$. The decoder produces the output sequence $y=\{y_1,y_2,y_3,....y_t\}$. The diagram of our model is shown in (Figure 4).

Bidirectional encoder

For the encoder $q(z|x)$, we have used a two-layer bidirectional LSTM network. We obtain forward ht and backward ht from bidirectional LSTM layer by processing input sequence $x=\{x_1,x_2,x_3,....x_t\}$. Forward hT and backward hT are concatenated to produce the final hidden state and from final h, μ and σ are produced:

$$\mu = W_{h\mu} h_T + b_\mu \tag{2}$$

$$\sigma = \log\left(\exp\left(W_{h\sigma} h_T + b_\sigma\right) + 1\right) \tag{3}$$

Where Whμ, Whσ, bμ, bσ are weight matrices and bias vectors respectively.

Using a bidirectional recurrent encoder helps to have a parametrization of the latent distribution

long-term context about the sequence of the input (Adam Roberts et al. (2018)) [7].

Hierarchical decoder

The decoder processes the latent vector z and produces to generate the output sequence. The simple decoder is not efficient for long sequences so based on the paper "A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music", (Adam Roberts et al. (2018)) [7] which proposed a novel hierarchical RNN for the decoder, we implemented our model. Assuming that the input sequence x can be divided into U subsequences y_u with endpoints i_u so that,

$$y_u = \left\{x_{iu,} x_{iu+1,} x_{iu+2}, \ldots, x_{iu+1} - 1\right\} \quad (4)$$

$$x = \left\{y_1, y_2, \ldots, y_u\right\} \quad (5)$$

Where i_{U+1} = T. Then latent vector z is passed through a fully connected layer to get the initial state of conductor RNN which produces $c = \left\{c_1, c_2, c_3 \ldots c_u\right\}$ for each subsequence. Now each vector c is individually passed through a fully connected layer followed by tanh activation to generate the initial state for the decoder. The decoder RNN produces a sequence of distributions over output tokens for each subsequence y_u. These output subsequences are also concatenated with previous output and passed as input to the decoder RNN.

Interpolation

For creative purposes, based on the paper "A Hierarchical Latent Vector Model for Learning Long- Term Structure in Music", (Adam Roberts et al. (2018)) [7], we carried out interpolation between 350 classic melodies dataset (A) and 350 jazz melodies (B), with SoftMax temperature 0.5 to sample the intermediate sequences. We choose "Data" interpolation as our baseline model and compare the results with interpolations from flat decoder and hierarchical decoder. In Figure 5, relative LM cost and hamming distance for the baseline model, flat decoder, and hierarchical decoder are marked with green diamonds, yellow circles, and red squares respectively (Adam Roberts et al. (2018)) [7]. In the baseline model, an element from either sequence a or b is chosen for each time step by sampling the Bernoulli random variable with parameter α.

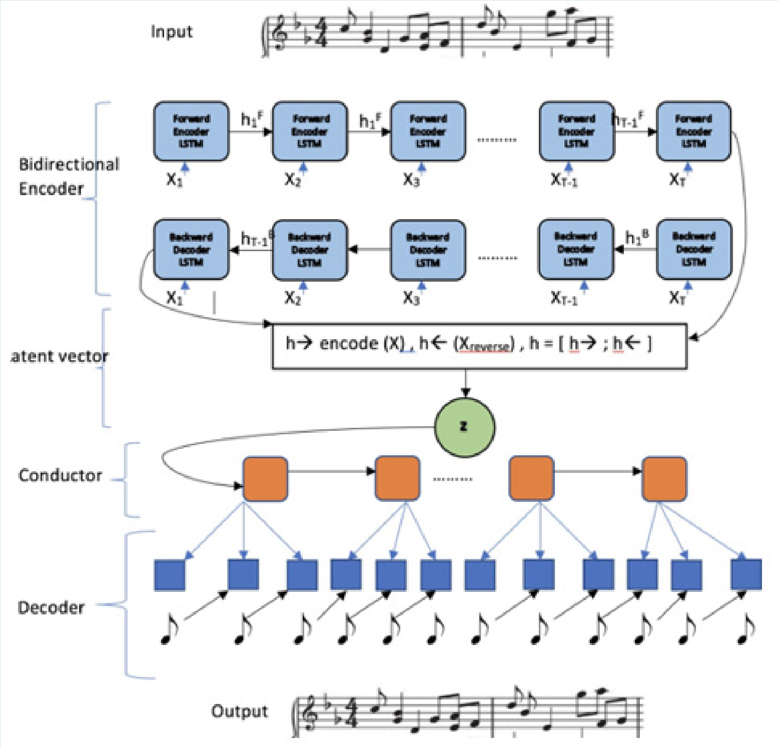

Figure 4: Schematic of hierarchical recurrent variational auto-encoder, (Music VAE).

i.e, $p\left(x_t = b_t\right) = \alpha$, $p\left(x_t = a_t\right)1 - \alpha$

Hamming distance is the proportion of timestep predictions that differ between the interpolation points and sequence A. As we can see in the top graph of (Figure 5), the hamming distance for the baseline model varies linearly, following the mean of the Bernoulli distribution. For flat and hierarchical decoders hamming distance varies smoothly and is less like sequence A. Because construction doesn't remain in one mode and suddenly jumps to another mode.

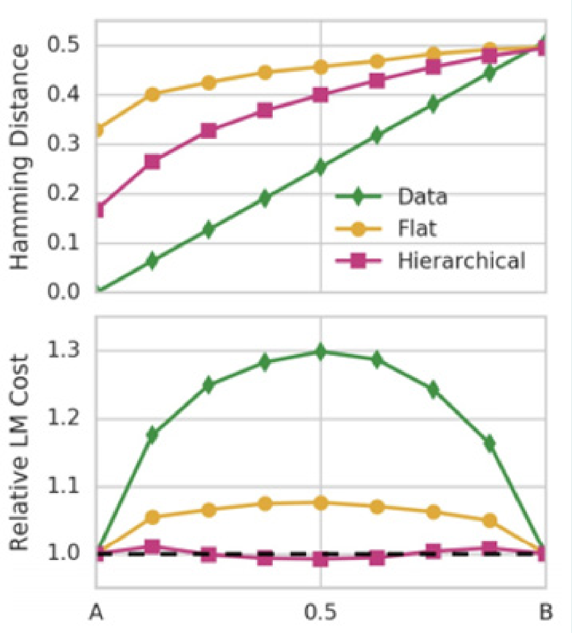

Figure 5: Interpolation (Adam Roberts et al. (2018)).

The relative LM cost for each interpolated sequence is given by, $C_\alpha / \left(\alpha C_B + \left(-1\alpha\right) C_A\right)$ where C_α is language model cost and C_A and C_B are costs for endpoint sequences A and B. The sudden bump for the baseline model shows a lower probability for interpolated sequences than the original melodies. The flat decoder does better than the baseline but the hierarchical decoder produces interpolations of almost equal probability to the original melodies. (Figure 6) shows the interpolation of classic melody A and Jazz melody B.

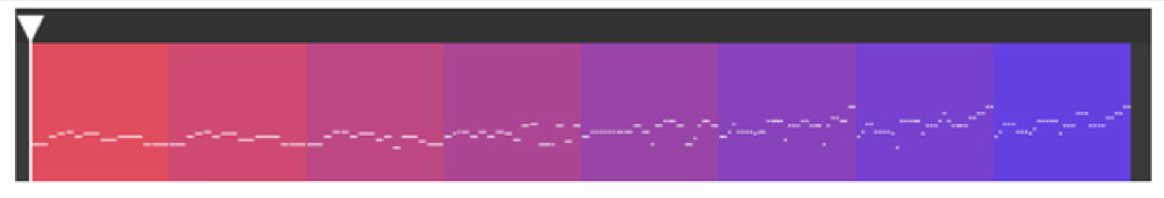

Figure 6: Interpolation.

Results

Experiments

We trained Melody-RNN models with Basic RNN and Lookback RNN respectively by feeding Jazz/Classical music dataset as input. (Figure 7), shows our result of the loss and accuracy graphs in different models and dataset for 700 steps, which took about 2-4 hours to train on a CPU for each model. Those models can be trained well after training 1000 steps, whose accuracy is all above 80%. Then we randomly give the model a single note as input to start generating the rest of them to see whether the model can generate a music having certain style we want. Also, we are interested in whether the model can generate specific style music based on other style melodies, so we also give the model a short different style midi file as a priming melody. For example, we used classical music to train the model and gave jazz music as the primary input to generate music.

Compared to existing models by feeding the same short midi file, we can see in the example in (Figure 8), that the generated music from lookback RNN has more musical, which is close to the input melody, than the one from basic RNN. However, they rarely generate style from training data but only style from input. Even though the loss term looks good in the training part, it does not mean we can always get good performance in the generating part. Some of them are recognized as machines, even noises, which will be discussed in the next section. We also trained the Music-VAE model to model 2-bar monophonic music sequences with jazz and classical datasets. However, it took longer time to train the model than RNN. Therefore, we only got the trained model with less than 50 training steps (about 2-3 days) on the CPU. We used our trained model to generate music in 2-Bar Melody Model part on Music-VAE Colab (https://colab.research.google.com/notebook#fileId=/v2/external/notebooks/magenta/music_vae/music_vae.ipynb).

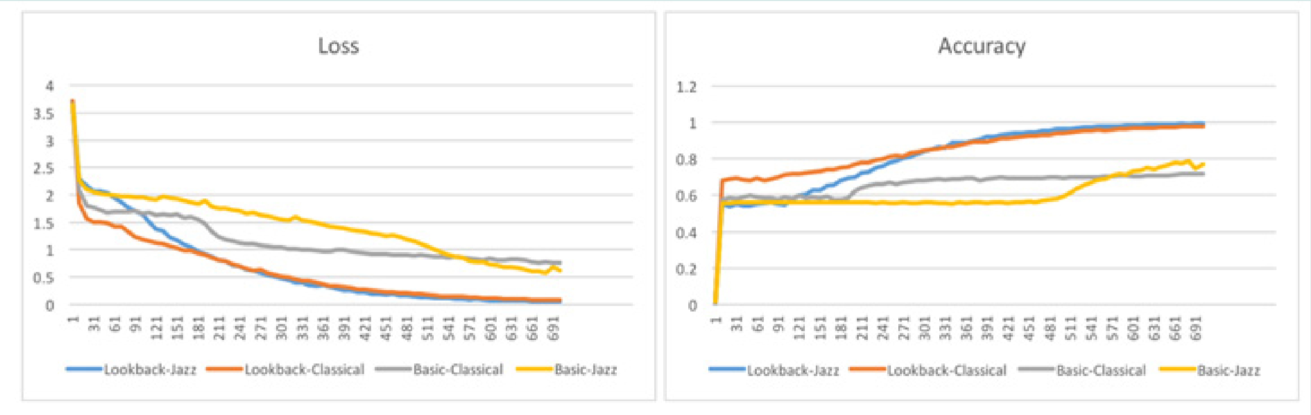

Figure 7: Loss (left) and Accuracy (right) graphs for model trained on different settings (Basic- RNN/Lookback-RNN, Jazz/Classical dataset).

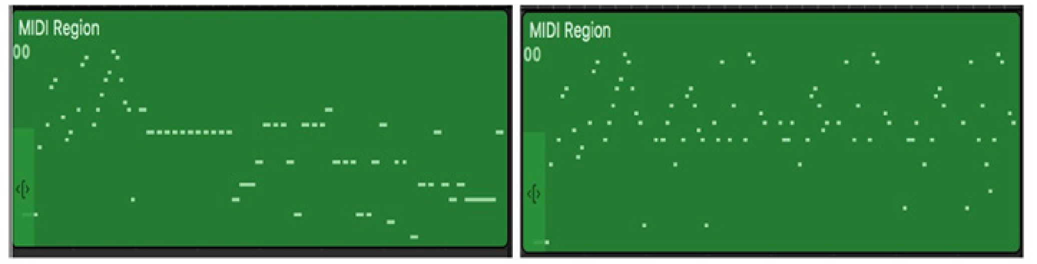

Figure 8: Generated Music from Basic RNN (left) and Lookback RNN (right).

It did gen- erate some interesting music and interpolate two musical melodies (Figure 9). Those music demos will be provided online (https://drive.google.com/drive/folders/1MBX7GUFXTRLVCBSBYlr7Lth2bEqujdAJ?usp=sharing), but music from Music-VAE 2-Bar Melody Model is too short (about 3-5 seconds) to allow people to recognize the human-performance and the style of music. Therefore, we only focus on melody RNN for evaluation in the next section.

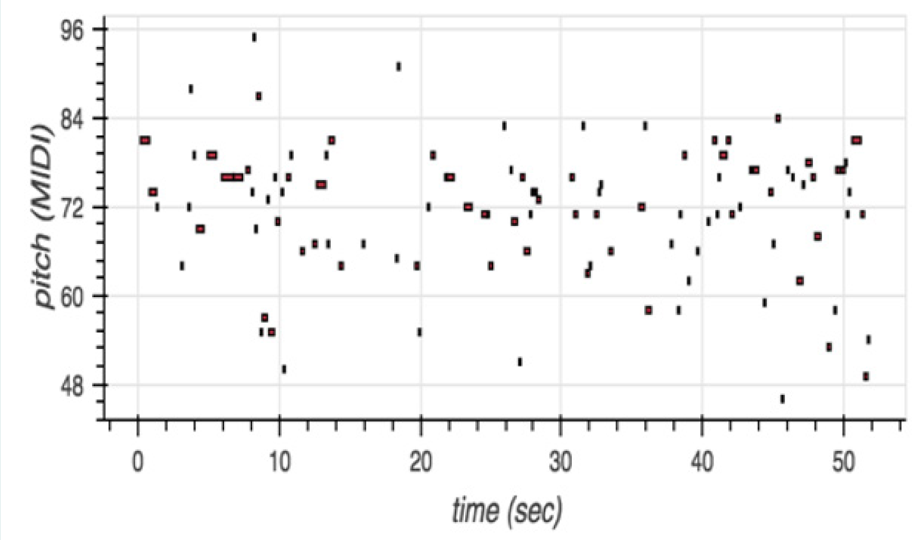

Figure 9: Interpolation of Jazz Music and Classical Music.

Citation: Fatemeh Jamshidi*, Maryam Bigonah, Daniela Marghitu and Richard Chapman. Machine Learning and Deep Learning Techniques in Music Style Generation: Trends and Challenges. Curr Tr Comp Sci & App 3(3)- 2024. CTCSA.MS.ID.000161.
DOI: 10.32474/CTCSA.2024.03.000161

Evaluation

Several projects have attempted to evaluate music generation models. Based on the results of previous studies, the most effective way to evaluate generated music is to maintain the integrity of beats and pitches, as confirmed by human assessment. Therefore, in this paper, we employed the Turing test to achieve this goal. We designed two listening tests: the "Identify Human" test and the "Identify Style" test, to compare generated melodies with human-composed ones. Participants were asked to rate the likelihood of human performance on a Likert scale and to identify the style of each piece of music, which was randomly sampled from human or machine compositions. Fifteen participants evaluated our model before submitting the project. We provided four samples each of human, generated classical, and generated jazz music.

For the "Identify Human" test, participants rated the performance on a Likert scale with five levels. For the "Identify Style" test, participants chose from five style options. To summarize the survey results in an understandable manner, (Figure 10) (left) shows the combined totals of "most likely" and "likely" responses as the "likely" label, and the combined totals of "most unlikely" and "unlikely" responses as the "unlikely" label. The "don't know" responses are labeled as "can't determine." For the "Identify Style" test, ((Figure 10) (right) shows the combined totals of correct style identifications as the "correct" label and the combined totals of incorrect style identifications and noise as the "incorrect" label.

The results of the "Identify Human" test ((Figure 10), Left) indicate that our model still struggles to compete with human performance, with 70% of respondents selecting "unlikely." This is attributed to several issues, such as the inability to generate multiple notes simultaneously, lack of rhythmic variation in the generated music, and the perceptible difference in quality between generated and human-composed music. The "Identify Style" test ((Figure 10), Right) also shows subpar performance, indicating that the model does not sufficiently capture the characteristics of musical styles to generate convincing melodies. The same issues identified in the "Identify Human" test likely contribute to these results as well.

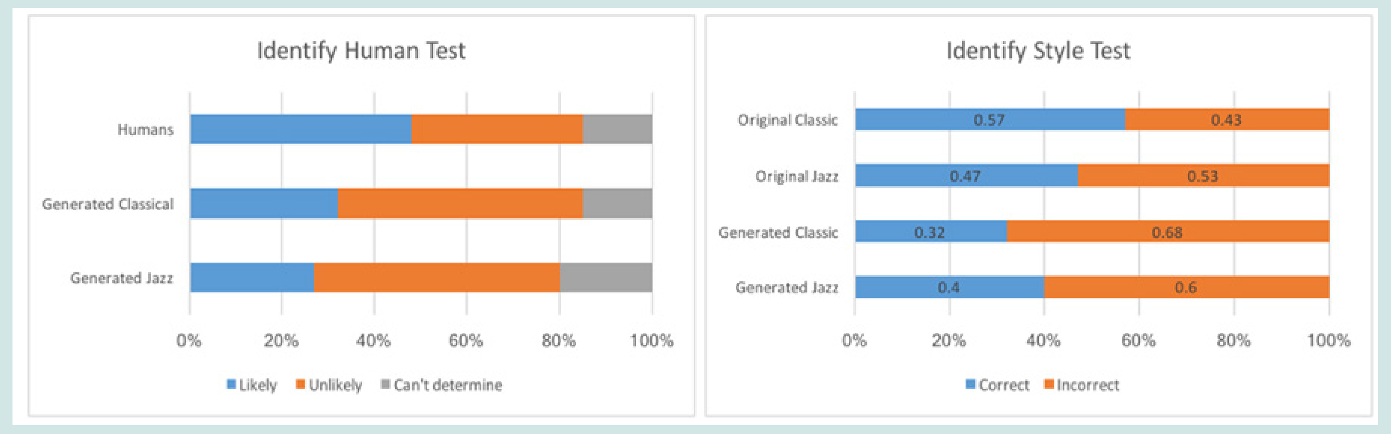

Figure 10: (Left): "Identify the Human" Survey Results, (Right): "Identify the Style" Survey Results.

Conclusion and Future Work

This paper demonstrates the application of a multi-layer LSTM model to generate music in two different styles. While the model shows promise, it requires significant improvements to generate stylistically accurate melodies. One key limitation is the model's inability to generate polyphonic music, as it currently only handles monophonic melodies. To address this issue, we propose inte-grating the model with the "Biaxial RNN" developed by Daniel Johnson Johnson (2017) [8], which can handle multiple notes simultaneously and maintain temporal and note pattern invariance, potentially enhancing the model's performance. Despite its limitations, the model provides an interesting tool for co-creative music composition. For instance, the Google Magenta Cyborg Collaboration demonstrates how AI can inspire and assist human musicians in creating new melodies Magenta (2019) [9]. This highlights the potential for AI models to support and augment human creativity in music.

Furthermore, Google has introduced several advanced melody models within the Magenta program, such as Attention RNN, Improved RNN, and Performance RNN. These models incorporate more sophisticated architectures and techniques, potentially offering superior performance in extracting stylistic characteristics and generating music. Future work will involve experimenting with these advanced models, comparing their effectiveness, and identifying the best model for generating stylistically coherent melodies.

In addition to exploring new models, future research will also focus on enhancing the current evaluation methodologies. While the Turing test and listening tests provide valuable insights, developing more objective and quantitative evaluation metrics will be crucial. This will ensure a comprehensive assessment of the models' performance and guide further improvements [10-16].

Overall, while there are challenges to overcome, the progress made so far demonstrates the potential of deep learning models in music generation. Continued research and development in this area will likely yield increasingly sophisticated tools that can create, inspire, and transform music in innovative ways.

References

1. JJ Bharucha, PM Todd (1989) Modeling the perception of tonal structure with neural nets. Computer Music 13(4): 44-53.
2. J Engel, C Resnick, A Roberts, S Dieleman, K Simonyan, et al. (2018) Neural audio synthesis of musical notes with wavenet autoencoders. CoRR.
3. I Malik, Ek Carl Henrik (2017) Neural translation of musical style.
4. S Dai, Z Zhang, GG Xia (2018) Music style transfer: A position paper.
5. O Cífka (2021) Deep learning methods for music style transfer. PhD thesis Institute Polytechnique de Paris.
6. David Ha, Douglas Eck (2017) A neural representation of sketch drawings.
7. JE Adam Roberts, C Raffel, C Hawthorne, D Eck (2018) A hierarchical latent vector model for learning long-term structure in music.
8. DD Johnson (2017) Generating polyphonic music using tied parallel networks. Computational Intelligence in Music, Sound, Art and Design: 128-143.
9. G Magenta (2019) Cyborg collaboration.
10. (2018) Magenta github: Melody rnn.
11. J Thickstun, Z Harchaoui, S Kakade (2017) Learning features of music from scratch. In International Conference on Learning Representations (ICLR).
12. C Raffel, DP Ellis (2016) Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching. In IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP): 292-296.
13. C Hawthorne, A Stasyuk, A Roberts, I Simon, S Dieleman, et al. (2018) Enabling factorized piano music modeling and generation with the maestro dataset.
14. C Donahue, Z C Lipton, J McAuley (2018) The nes music database: A multi-instrumental dataset with expressive performance attributes.
15. M Allan, CKI Williams (2005) Harmonising chorales by probabilistic inference. In Advances in neural information processing systems: 25-32.
16. E Waite (2018) Generating long-term structure in songs and stories.

DOI: 10.32474/CTCSA.2024.03.000161