Dilated CNN and LSTMs for Music Style Transfer

Haojun Li (haojun) [1], Jialun Zhang (jzhang07) [2]
[1] Department of CS, Stanford University, [2] ICME, Stanford University

***Abstract*—In this study, we present two neural network models for music style transfer. The first is a Dilated Convolutional Neural Network that was trained separately on Jazz and Classical piano music. This allows us to generate classical or Jazz-like dynamics (how the volume of the music changes over time) for both classical and Jazz pieces. The second neural network is an LSTM model used to generate Jazz or Classical Music *inspired* by a Classical or Jazz piece. Using LSTM, we were able to generate music with distinctive Jazz or Classical characteristics, even though the overall structure of the music is still not ideal.**

I. Introduction

Music style can be defined as the speed of the music, the rhythm, the chords, and many other factors. We would like to investigate whether we can transfer music from one style to the other by using deep learning methods and models. More specifically, we would like to see if music style transfer can be achieved through changing 1) How hard each note is being pressed (the velocity of each note) and 2) The chords themselves. We are only interested in Jazz and Classical piano style for this project.

We used a set of MIDI files of 2 genres (Jazz and Classical). We transformed it into 3 matrices (notes-on, velocity, and chords). For task 1, the input of the model is the notes-on and chords matrices, and the output would be the velocity matrices, For task 2, the input would be just the chords matrices, and the output would also be chords matrices.

For transferring style on how hard the note is being pressed, we will be using two dilated Convolutional Neural Networks (Wavenet[4]), one for classical music and one for jazz music. Then, we will take a jazz music's chords and notes-on matrix and put it into the classical music's CNN to generate a "classical velocity" matrix. For transferring the chords themselves, we will first train a different Dilated CNN that classifies the music as jazz or classical, and use the activation layers within the network as features (or encoding) of the music. Then, we trained 1 modified LSTM networks with only classical music chords and features, and put one jazz music feature into the model to see if the network can generate a classical music chords "inspired" by a jazz music.

Disclaimer: We used the same technique in subtask 1 of task 2 (classifying music genres through dilated CNN) in our CS229 project. But we did not use the same dataset, nor have the same data representation and output, and completely different code.

II. Related Works

Music style transfer generally involves classifying the music first, and then incorporating other genre's music feature into the existing music by either switching or adding instruments while keeping the melody the same. Other methods changes the rhythm or speed. This type of style transfer can easily be done without the help of neural networks.

Recent works have investigated whether there are music style differences that can be recognized and transferred using neural networks.[3] In the previous referenced study, the author uses a bidirectional LSTM to generate the new velocity of the music. Although we will be using similar data representations, we will use wavenet architecture to generate the style matrix instead of the training method defined in that paper. Other generative methods have been investigated by Wavenet[4] where they generate speech using dilated convolution networks. Some other studies uses GANs to transfer the style and have achieved reasonable results[2].

There are also other style transfer method being used on images[5]. The study also involved convolutional neural network with content and style matrices.

III. Dataset and Representations

The dataset used is from the Neural Translation of Musical Style paper[3] with similar data representation format.[1] We have 349 pieces of classical music in MIDI form and 349 pieces of jazz music in MIDI form.

Each MIDI file is pre-processed into three 2D Numpy arrays, corresponding to the three graphs in Figure III.1. On each of these graphs, the vertical axis represents the time steps and the horizontal axis represent the notes, which there are 59 in total as we omitted the top 37 notes and bottom 32 notes. In left most figure, each row represents the notes that are being played at a particular time. In the middle figure, the "velocity", or the strength of a note, is denoted by the intensity of its color, where brighter yellow means higher intensity and darker blue means lower intensity. Finally, in the right figure, we have a representation of the exact moment when a note is played.

[1]We used the same representation format but we wrote all the code ourselves. You can reference the code written on our github

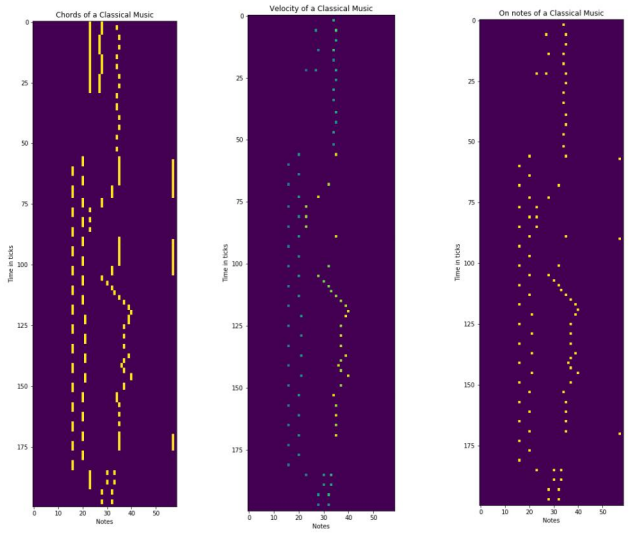

Figure III.1: Classical Chords, Velocities heat map, and notes on/off matrices

IV. Models

A. Velocity Style Transfer

Given a 2D Numpy array representing the chords at each time step (corresponding to the chord matrix in figure III.1), our first model attempts to predict the velocity for each note (corresponding to the velocity matrix in figure III.1). Thus, both the input and output of our model is a 2D Numpy array of the same shape. The architecture that we use is called a *Dilated Convolutional Network*. A diagram is shown in IV.1. In this toy example we only have two layers, with the input layer on the bottom and the output layer on top. For the first layer, the filter size is 2 and the dilation rate is 1, which means that each blue node depends on 2 yellow nodes that are next to each other. For the second layer, the filter size is 2 and dilation rate is 2, which means that each green node depends on 2 yellow nodes that have a distance of 2 (or a gap of 1). In the actual model, each input node would also have 59 channels, corresponding to the 59 notes, and the total length of the input (number of nodes) would be equal to T, the number of time steps. Thus our input is a matrix of dimensions $T \times 59$, whose elements are either 0 or 1. The filter size is fixed, but the dilation rate doubles each layer. Finally, notice that using a dilation rate ≥ 1 will cause the length of the layers to decrease, but we want to output to be the same shape as the input. Therefore, we need to pad the input to be sufficiently long so that even after dilation, the actual length remains the same. The advantage of this architecture compared to the traditional CNN is twofold: first, we will have fewer number of weights (due to dilation), making the model easier to train. Second, dilation allows for nodes that are far away (in terms of separation in time) to be more strongly correlated. This is more ideal for sequential data like music.

For optimization, we use a modified version of the L^2 loss defined by

$$J(\theta) = \frac{1}{n}\sum_i \mathbb{1}\{\text{notes-on}_i\}(\hat{y}_i - y_i)^2,$$

where Θ is the weights in our neural network and $\mathbb{1}\{\text{notes-on}_i\}$ is 1 if $y_i > 0$ and 0 otherwise. This indicator function essentially applies a mask to the output $\hat{y}$ and computes the L^2 using the new $\hat{y}$. The reason for defining the loss this way is that we really only care about the velocity at the moment when a note is struck, assuming that the velocity will remain the same during the whole duration of the note. We optimize over Θ using the `adam` optimizer, which combines gradient descent with momentum and RMSprop.

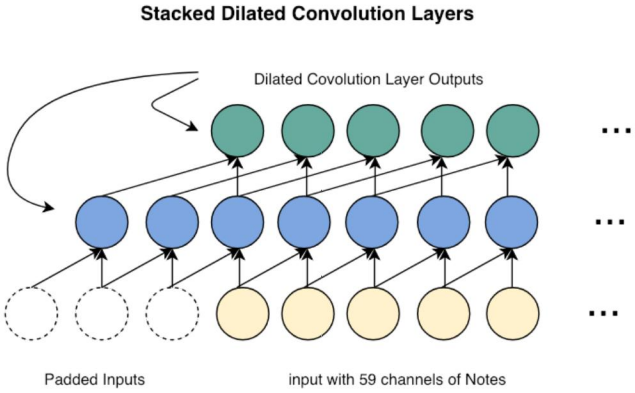

Figure IV.1: Toy Example of an Dilated Convolutional Neural Network

To accomplish a *style transfer* for velocity, we train **two** Dilated CNNs, one for Jazz and one for classical music. The result is if we are given a piece of Jazz music encoded as an input matrix corresponding to chords at different time steps, such as the one show in Figure III, we would be able use the Dilated CNN trained on classical music to predict classical-like dynamics for the Jazz piece. This also works the other way around.

B. Chords Style Transfer

We first trained a Dilated CNN using the following architecture depicted in figure IV.2[2] Essentially it is a 2 layer 1 dimensional convolution layer followed by average pool, batch norm[1], and relu activation. This is repeated once, and connected to a 50 node layer, then connected to a 1 node output layer with sigmoid activation, outputting the probability of whether it is classical or jazz.

Then we use the light blue layer activation outputs as our feature encoding, and feed that with the original music into an LSTM. Instead of concatenating the feature vector with the music themselves, we used a fully connected layer with learnable weights and transformed it into the shape of initial cell state of the LSTM. Then we trained our LSTM The architecture is shown in figure IV.3. One reason for not

[2]We used similar figure for our CS229 project since we are using the same technique

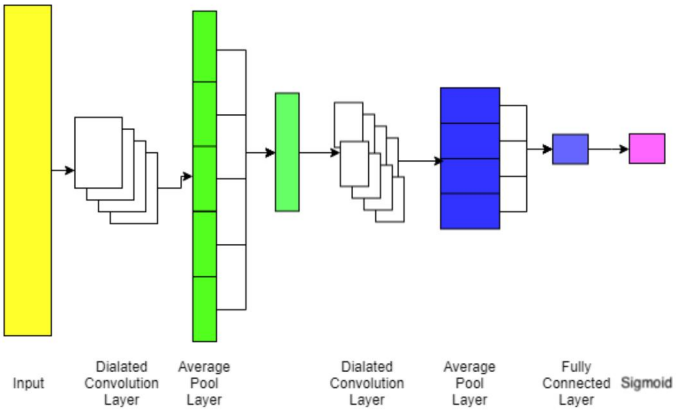

Figure IV.2: Dilated Convolutional Network Architecture for classification

concatenating the feature vector is that if the music style changes in the middle of the music we want the LSTM to be able to change dynamically. We also chose sigmoid as our output activation instead of softmax activation because multiple notes out of the 59 notes can be on at the same time.

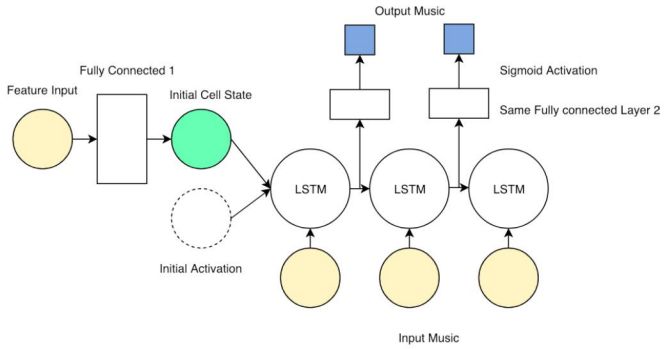

Figure IV.3: LSTM training architecture. Each yellow node on the bottom is a single time stamp input of 59 depth binary array (each depth is a note). Each blue node is a 59 depth array of probabilities from sigmoid activation

To generate music, we used the same LSTM cells and Fully Connected layers with the structure depicted in figure IV.4. The architecture is very similar to original architectures for LSTM generation. However when the output is being fed back into the network, each of the 59 notes is sampled through a uniform distribution of probability equal to the output of the sigmoid activation. this allows us to always have a binary array as input to next time stamp of the network.

V. Results and Discussion[3]

A. Velocity Style Transfer

Since we are transferring velocity style, we have 59 filters at each level corresponding to 59 notes. We have experimented

[3]All experiments here are part of the repository https://github.com/LithiumH/CS230-Music-Style-Transfer

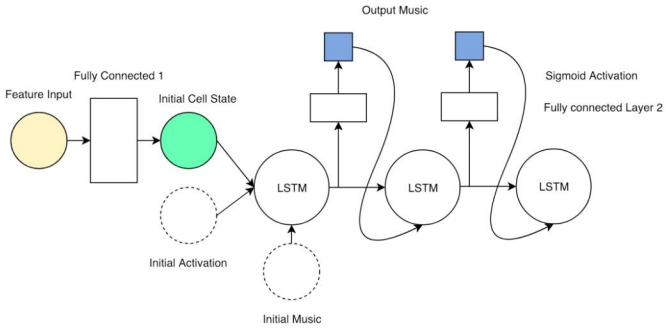

Figure IV.4: LSTM generating architecture

with different hyper parameters and ultimately decided on filter size 64 and dilation rate of 1, then filter size 32 and dilation rate of 2, then filter size 16 and dilation rate of 4. We padded only the beginning of the chords matrix, which means that each output time stamp output looks at 64 time stamps directly before it. (In retrospect, we should have padded both front and back so each time stamp output will look at 32 steps before and 32 steps after.) This achieves the fastest convergence rate and lowest cost. We then used standard ADAM optimizer with learning rate 0.001. Using an ADAM optimizer allows us to speed up learning. The exact architecture is the depicted in figure IV.1.

Even though we could "overfit" the data since we are not looking to make the model more generalizable, we still splitted our data into 3750 training examples and 417 testing examples. Out of the 417 classical test examples, the classical CNN achieved a mean squared loss of 3.34, which is extremely well.

We have also received promising results on Velocity Style Transfer using Dilated CNN. The true and predicted velocity heat maps for a single jazz piece of music passed through a classical Dilated CNN is shown in figure V.1. As we can see from the picture, higher notes of the jazz music is emphasized by the neural network. This make sense because as we can see in figure III.1, classical music generally have higher notes (or right hand notes) emphasized because the lower notes (or left hand notes) are complementary to the higher notes. However, in Jazz music as evident in V.1 each note of a single time stamp is equally emphasized. Moreover, Jazz music notes velocities are not consistent across time stamps like classical musics. This is also realized by the neural network as evident by the fact that note differences in velocity are more similar to each other in the generated velocity matrix.

We have generated the actual MIDI on our GitHub repo and if you are interested you can take a look at our MIDI files generated called "vel_generate_j2c.midi". The original music can be heard in "vel_original_j2c.midi". There is a subtle difference in the music.

B. Chords Style Transfer

First we trained a dilated CNN to classify whether the music is classical or jazz. The hyper parameters we used are as follows. The first layer has 4 filters of size 32 with a

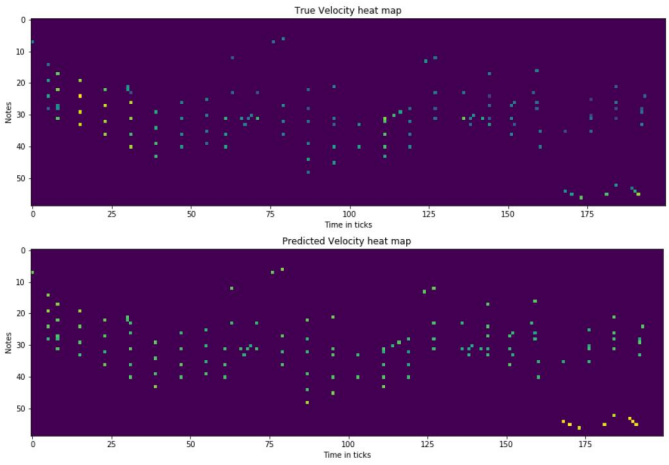

Figure V.1: True velocity of a Jazz music and classical dynamic predictions of the same Jazz Music

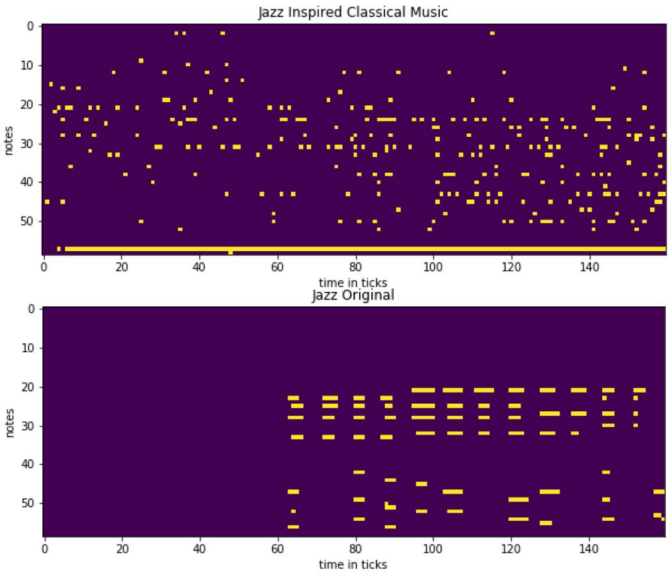

Figure V.2: Jazz Inspired Classical Music and Original Jazz chords

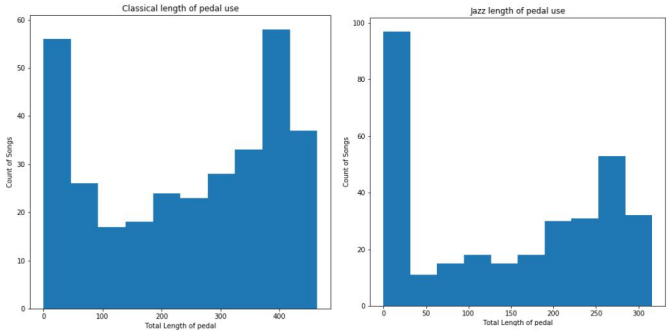

Figure V.3: Classical and Jazz pedal press length

dilation rate of 4. Then we have an average pooling layer of window size 4. This is followed by 16 filters of size 16 with a dilation rate of 2. This is then followed by an average pooling layer of window size 2. Lastly, they are connected to a fully connected layer of size 50 (which will be our feature encoding) and then connected to a layer with output size 1. We also used ADAM optimizer with learning rate 0.001 and standard parameters. We also splitted the data into 3750 training examples and 417 test examples. This CNN achieved a test accuracy of 93%.

We have some very interesting results from our experiment. We trained the network for a day and it was able to predict reliable music to some extent. The final loss for training the LSTM is 100 with average binary accuracy of 95%. This is still not the best accuracy considering that most of the values are 0. Thus as expected, the resulting audio is not as pleasing as we expected it to be. The chords of the classical music inspired from jazz is shown in the figure V.2.

There are some patterns worth discussing. For example, we see that even though the original music does not have the pedal pressed at all, the generated music has the pedal pressed all the way through. This is because more classical music have padel pressed for longer than Jazz music. The histograms of pedal length is shown in figure V.3. Thus we hypothesize that the inspired music's pedal pressed is an indication that the style transfer is somewhat working.

VI. Conclusions

In conclusion, we have successfully realized music style transfer through velocity by using a dilated CNN architecture. The classical CNN is able to predict classical music's velocity very well, while the jazz CNN is able to predict jazz music's velocity very well. And when a jazz music is fed into the classical CNN, the network is able to produce a velocity similar to other classical music.

We have only limited results for music style transfer through chord differences. We suspect it is because we did not train the network for enough epochs, and we will continue to train and improve the network with more hidden states and activation states. However, the current result is promising in that we can find distinct style transfer from an jazz inspired classical music.

VII. Future Work

In addition to more epochs, we would also add a discriminator next to our generator to force the music generated to be more "real". We would also experiment with GANs with our feature extraction/generation approach. GANs have proven to produce more realistic results and they would definitely work better than the LSTM by themselves.

References

[1] Ioffe, Sergey and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.” *ICML* (2015).

[2] Jayakumar, Suraj, et al. âĂIJToneNet : A Musical Style Transfer âĂŞ Suraj Jayakumar âĂŞ Medium.âĂİ Medium.com, Medium, 28 Nov. 2017, medium.com/@suraj.jayakumar/tonenet-a-musical-style-transfer-c0a18903c910.

[3] Malik, Iman, and Carl Henrik Ek. "Neural translation of musical style." arXiv preprint arXiv:1708.03535 (2017).

[4] Van Den Oord, AÃd'ron, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. "WaveNet: A generative model for raw audio." In SSW, p. 125. 2016.

[5] Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "Image style transfer using convolutional neural networks." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2414-2423. 2016.

VIII. Contributions

- Haojun - Dilated CNN modeling and experiments and LSTM experiments. Feature extraction. Poster lead.
- Jialun - Report lead and composition. Generating results.