DeepMix

Joseph Ballouz, Ashwin Neerabail, William Park
Department of Computer Science
Stanford University
{jballouz, ashwinn, wpark73}@stanford.edu

Abstract

This paper investigates usage of CNNs in generating audios containing the content (i.e. tempo, melody) of one audio clip and the style (i.e.timbre of the primary instrument) of another audio clip. The solution to this problem can potentially allow generation of new audio through the mixture of two completely different types of audio content. We try to achive this by testing 4 CNN architectures: 1-layer Random Weights, 2-layer Random Weights, VGGnet-16, and WaveNet. The 1-layer Random Weights works best and gives descent results for certain style and content files.

1 Introduction

With the use of convolutional neural networks (CNNs), we saw a vast rise in works involving style transfer to create a new image involving the content from one image and the style from a different image. In this project, our goal is to evaluate these techniques for Audio. Style for audio files is not a well defined paradigm. Style could mean compositional style of a particular artist , it could mean specific arrangement of instruments used or it could mean the timbre of the instrument. For this paper, melody and tempo is referred as the content and the timbre of the instrument is used as the style. The way this is done is by applying a Fourier transform to generate a spectrogram (frequency vs. time) that is fed to a CNN and optimized to reduce the loss (combination of style loss and content loss similar to style transfer for images but using the spectrogram as our image). The output audio is then generated by taking the inverse fourier transform of the output spectrogram.

2 Related work

Style Transfer on Images:
Seminal work for Style transfer was done by Gatys et al. [4] primarily for generating a new image containing the content of one image rendered in the style of a different image.
Audio Texture Synthesis using Random Shallow Network:
Ivan et al. [7] showed that Shallow network initialized with random weights can perform as well or better than pre-trained networks for Audio Texture synthesis.
Style Transfer on Audio:
Dmitry et al.[1] applied style transfer technique on spectrogram images to successfully transfer style for audio files using 1 Layer Shallow CNN. In this paper , we evaluate 1-layer and 2-layer Random Shallow CNN against pre-trained networks, namely a. VGGNet trained on images (to operate on spectrogram) b. Wavenet trained on audio files (to operate directly on audio files)

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

3 Dataset and Features

3.1 Dataset

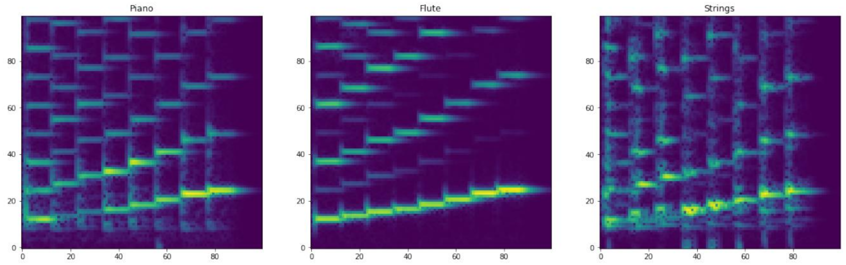
For this project we have used Random weighted CNN and pre-trained models - VGGNet and Wavenet. So we did not need any data for training. For mixing , we have collected style and content clips from the royalty free website www.bensound.com.

3.2 Pre-procesing

First we trim the given audio using the ffmpeg library to 10s. We then convert the audio signal to spectrogram via Short-Time Fourier Transform (STFT).

3.3 Spectogram

A spectrogram is a visual representation of the frequencies that make up the sound, from low to high, and how they change over time, from left to right. Different instruments vary in the overtones they produce though they have the same pitch and loudness.
Spectrograms of same notes played on Piano, Flute and Strings shown below:



3.4 Post-processing and Evaluation of Outputs

The output audio is obtained from the output spectrogram via the inverse Fourier Transform. We noticed that the loss function is not a good absolute measure for the quality of the generated audio, it only indicates approximately how well the model converged. To evaluate our results, we manually listen to the generated output clips and validate whether the content and the style were mixed in an aesthetically pleasing way while still retaining the melody and tempo of the original content. We use the following color scale to judge (see this scale in use in section 5: Results)

noise	bad	ok	good

4 Methods

Based on the work done by [8], shallow convolutional neural networks with the filter values set to random weights are a good way synthesizing the texture of an image which is equivalent to the 'content' of the audio file.
Thus, the first network architecture is a 1-layer CNN with Random Weights. The diagram below explains the model architecture and defines the content and style matrices.

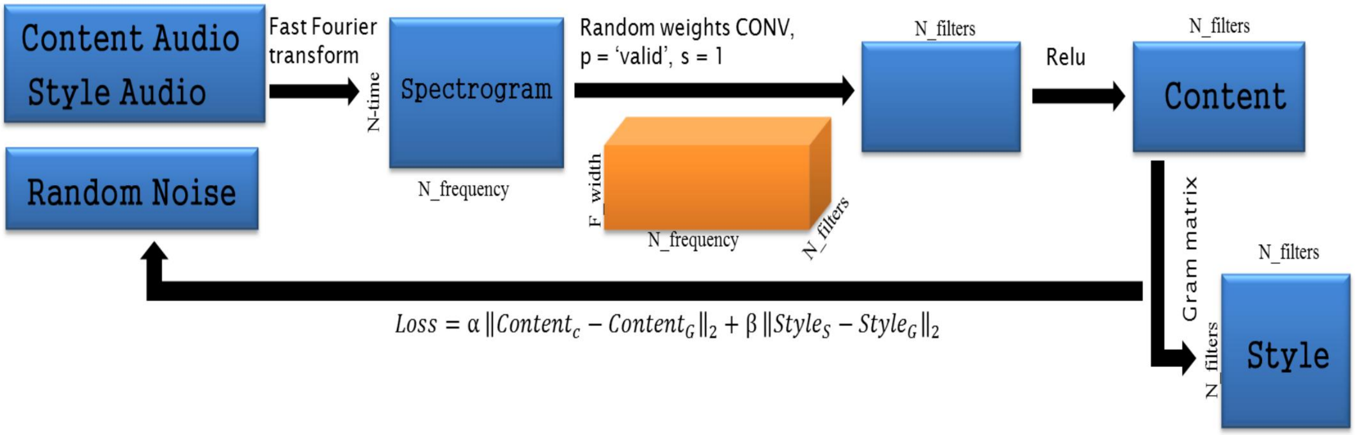


Figure 1: 1-layer Random Weights

As is shown by the loss function, the model tries to minimize the L2 distances between the content matrices of the content audio and the generated audio as well as between the style matrices (gram matrix of the content matrix in this case) of the style audio and the generated audio. The hyper-parameters α and β are used to tune the presence of more content or more style in the generated audio.
The second architecture we used is a 2-layer Random Weights architecture as is shown below. The loss function is the same as before but the content and style matrices have different definitions because of the added random weights convolutional layer.

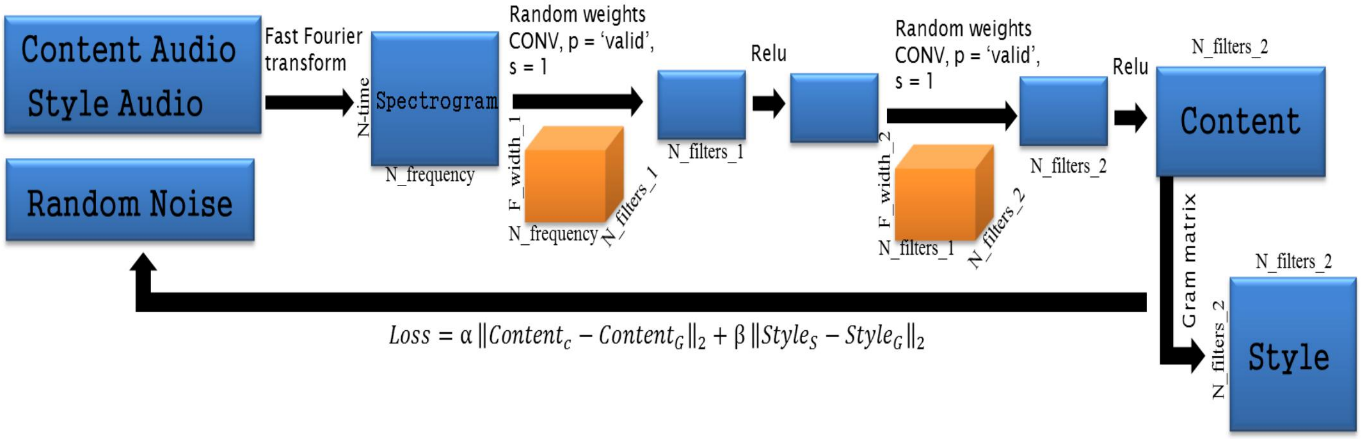


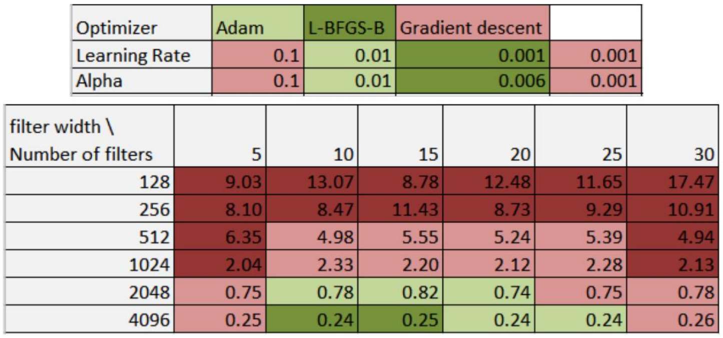
Figure 2: 2-layer Random Weights

The third architecture is the VGG-19 pre-trained network (taken from the Coursera Assignment 'Art Generation with Neural Style Transfer')where the style matrix is the average gram matrix of the layers conv1-1, conv2-1, conv3-1, conv4-1 and conv5-1, and the content matrix is the output of conv4-2.
The fourth architecture is the WaveNet pretrained network (see ref. [10]) where the network has an encode and a synthesize (decode) feature. The content matrix is the output of the model (the 'encode' of the input audio), and the style matrix is the gram matrix of the output (gram matrix of the content matrix). After running the model, The generated decoded audio synthesized (decoded) to obtain the generated audio.

5 Experiments/Results/Discussion

5.1 Hyperparameter Tuning

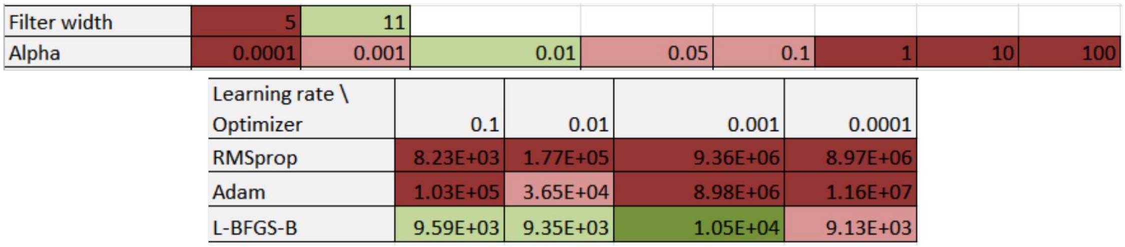
The results for tuning hyperparameters number of filters,filter width,learning rate, Optimization Algorithms are shown below.
1-layer Random Weights:



Optimizer	Adam	L-BFGS-B	Gradient descent	
Learning Rate	0.1	0.01	0.001	0.001
Alpha	0.1	0.01	0.006	0.001

filter width \ Number of filters	5	10	15	20	25	30
128	9.03	13.07	8.78	12.48	11.65	17.47
256	8.10	8.47	11.43	8.73	9.29	10.91
512	6.35	4.98	5.55	5.24	5.39	4.94
1024	2.04	2.33	2.20	2.12	2.28	2.13
2048	0.75	0.78	0.82	0.74	0.75	0.78
4096	0.25	0.24	0.25	0.24	0.24	0.26

2-layer Random Weights:



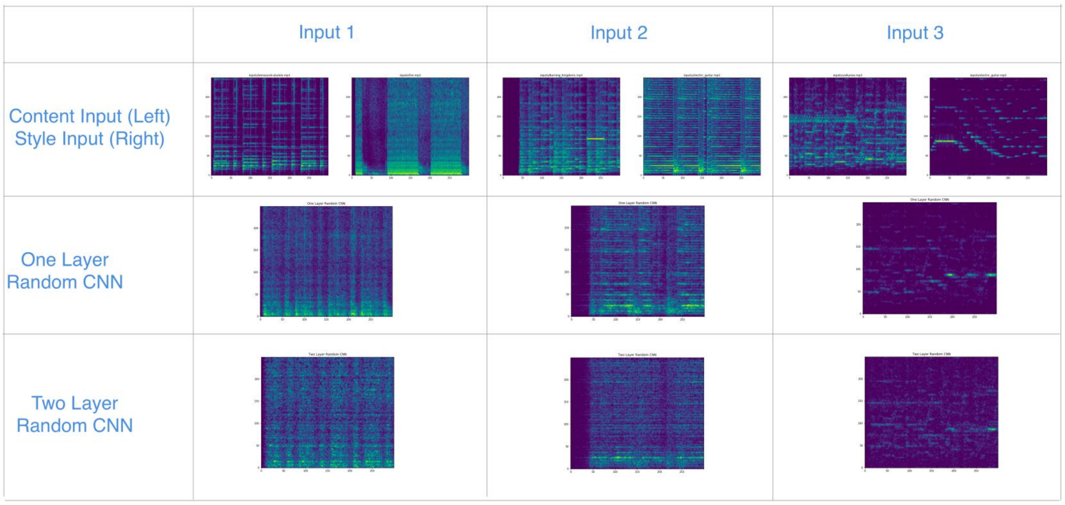
Filter width	5	11						
Alpha	0.0001	0.001	0.01	0.05	0.1	1	10	100

Learning rate \ Optimizer	0.1	0.01	0.001	0.0001
RMSprop	8.23E+03	1.77E+05	9.36E+06	8.97E+06
Adam	1.03E+05	3.65E+04	8.98E+06	1.16E+07
L-BFGS-B	9.59E+03	9.35E+03	1.05E+04	9.13E+03

VGGnet-16 and WaveNet:
The outputs from these two pretrained network were mainly noise and there was nothing to tune. Althoug VGGnet-16 should be well suited for extracting the styles of images, it is not well suited for spectograms (a specific type of image).

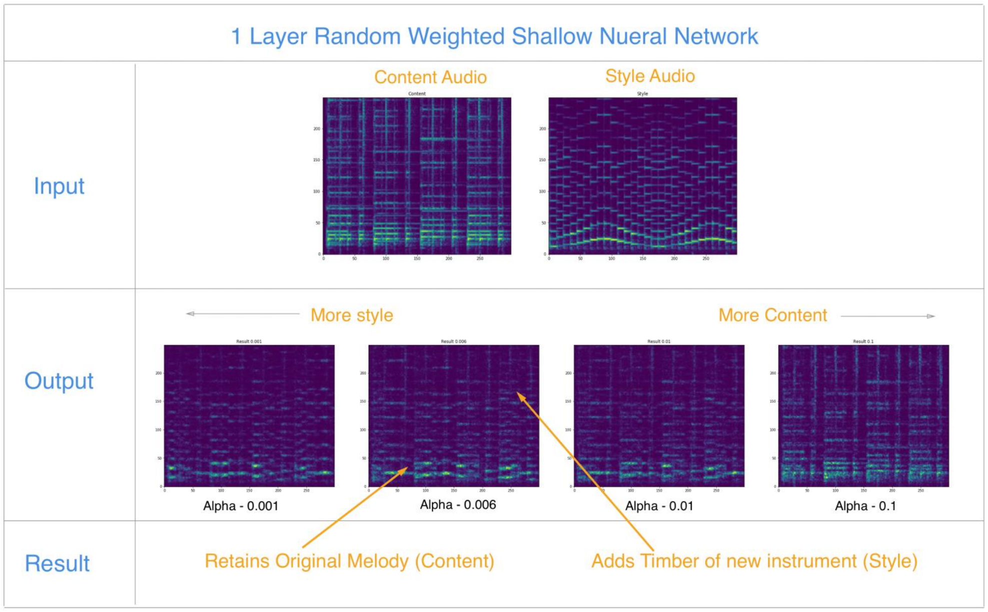
5.2 Results

Best results were obtained for 1-layer Random Weights with 4096 filters of width 11 using L-BFGS-B optimizer with learning rate of 0.001 and Alpha of 0.006. The 2-layer Random Weights with 2048 filters of width 11 using using L-BFGS-B optimizer with learning rate of 0.001 and Alpha of 0.001 gave ok results. Pretrained networks like VGGNet and AudioNet did not perform as well and did not produce meaningful output.
We get really good results for monophonic sounds like generating original melody played in the style of a different instrument.Polyphonic sounds and human voice transfer don't work well.



5.3 Discussion

By increasing the Alpha (Weight used for Content Loss) we notice more of the Content gets retained. By lowering values of Alpha , more of the style is retained. 0.005-0.006 gives the best result where it retains the original melody (Content) while adding the timber of the instrument used as the style.



6 Conclusion/Future Work

Shallow CNN with Random-Weights performs really well for texture synthesis for audio content extraction. We have used timbre of the instrument to represent style in our work. This works well for monophonic sounds. Further research is needed to extract human voice and polyphonic sounds. Using a pre-trained network for audio like Google Magenta to learn compositional style of an artist to generate music interpretation from a different composer for the same input song is something that needs further exploration.

7 Contributions And Code Link

Ashwin worked on implementing VGGnet-16 and WaveNet as understanding the effects of alpha on the output. Joseph worked on the 2-layer Random Weights and its hyper-parameter tuning. William worked on the 1-layer Random Weights and its hyper-parameter tuning. All members worked on writing the report.

References

[1] Audio texture synthesis and style transfer by Dmitry Ulyanov and Vadim Lebedev https://dmitryulyanov.github.io/audio-texture-synthesis-and-style-transfer/

[2] Ulyanov, D. & Lebedev, V. Audio texture synthesis and style transfer. Retrieved from `dmitryulyanov.github.io/audio-texture-synthesis-and-style-transfer/`, December 2016.

[3] Grinstein, E. & Duong, N.Q.K. & Ozerov, A. & Perez, P. Audio Style Transfer. *ArXiv e-prints*, November 2018.

[4] Gatys, L.A. & Ecker A. S. & Bethge M. Image Style Transfer Using Convolutional Neural Networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[5] He, K. & Wang, Y. & Hopcroft, J. A Powerful Generative Model Using Random Weights for Deep Image Representation. *ArXiv e-prints*, June 2016.

[6] Luan, F. & Paris, S. & Shechtman, E. & Bala, K. Deep Photo Style Transfer. *ArXiv e-prints*, April 2017.

[7] Ustyuzhaninov, I. & Brendel, W. & Gatys, L.A. & Bethge, M. Texture Synthesis Using Shallow Convolutional Networks with Random Filters. *ArXiv e-prints*, May 2016.

[8] Coursera Convolutional Neural Networks Programming Assignment: Art generation with Neural Style Transfer