Time Domain Neural Audio Style Transfer

Parag K. Mital
Kadenze, Inc.*
parag@kadenze.com

Abstract

A recently published method for audio style transfer has shown how to extend the process of image style transfer to audio. This method synthesizes audio "content" and "style" independently using the magnitudes of a short time Fourier transform, shallow convolutional networks with randomly initialized filters, and iterative phase reconstruction with Griffin-Lim. In this work, we explore whether it is possible to directly optimize a time domain audio signal, removing the process of phase reconstruction and opening up possibilities for real-time applications and higher quality syntheses. We explore a variety of style transfer processes on neural networks that operate directly on time domain audio signals and demonstrate one such network capable of audio stylization.

1 Introduction

Audio style transfer (1) attempts to extend the technique of image style transfer (2) to the domain of audio, allowing "content" and "style" to be independently manipulated. Ulyanov et al. demonstrates the process using the magnitudes of a short time Fourier transform representation of an audio signal as the input to a shallow untrained neural network, following similar work in image style transfer (3), storing the activations of the content and Gram activations of the style. A noisy input short time magnitude spectra is then optimized such that its activations through the same network resemble the target content and style magnitude's activations. The optimized magnitudes are then inverted back to an audio signal using an iterative Griffin-lim phase reconstruction process (4).

Using phase reconstruction ultimately means the stylization process is not modeling the audio signal's fine temporal characteristics contained in its phase information. For instance, if a particular content or style audio source were to contain information about vibrato or the spatial movement or position of the audio source, this would likely be lost in a magnitude-only representation. Further, by relying on phase reconstruction, some error during the phase reconstruction is likely to happen, and developing real-time applications are also more difficult (5), though not impossible (6). In any case, any networks which discard phase information, such as (5), which build on Ulyanov's approach, or recent audio networks such as (7) will still require phase reconstruction for stylization/synthesis applications.

Rather than approach stylization/synthesis via phase reconstruction, this work attempts to directly optimize a raw audio signal. Recent work in Neural Audio Synthesis has shown it is possible to take as input a raw audio signal and apply blending of musical notes in a neural embedding space on a trained WaveNet autoencoder (8). Though their work is capable of synthesizing raw audio from its embedding space, there is no separation of content and style using this approach, and thus they cannot be independently manipulated. However, to date, it is not clear whether this network's encoder or decoder could be used for audio stylization using the approach of Ulyanov/Gatys.

To understand better whether it is possible to perform audio stylization in the time domain, we investigate a variety of networks which take a time domain audio signal as input to their network:

*http://kadenze.com

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

using the real and imaginary components of a Discrete Fourier Transform (DFT); using the magnitude and unwrapped phase differential components of a DFT; using combinations of real, imaginary, magnitude, and phase components; using the activations of a pre-trained WaveNet decoder (9; 8); and using the activations of a pre-trained NSynth encoder (8). We then apply audio stylization similarly to Ulyanov using a variety of parameters and report our results.

2 Experiments[2]

We explore a variety of computational graphs which use as their first operation a discrete Fourier transform in order to project an audio signal into its real and imaginary components. We then explore manipulations on these components, including directly applying convolutional layers, or undergoing an additional transformation of the typical magnitude and phase components, as well as combinations of each these components. For representing phase, we also explored using the original phase, the phase differential, and the unwrapped phase differentials. From here, we apply the same techniques for stylization as described in (1), except we no longer have to optimize a noisy magnitude input, and can instead optimize a time domain signal. We also explore combinations of using content/style layers following the initial projections and after fully connected layers.

We also explore two pre-trained networks: a pre-trained WaveNet decoder, and the encoder portion of an NSynth network as provided by Magenta (8), and look at the activations of each of these networks at different layers, much like the original image style networks did with VGG. We also include Ulyanov's original network as a baseline, and report our results as seen through spectrograms and through listening. Our code is also available online[3].

3 Results

Only one network was capable of producing meaningful audio reconstruction through a stylization process where both the style and content appeared to be retained: including the real, imaginary, and magnitude information as concatenated features in height and using a kernel size 3 height convolutional filter. This process also includes a content layer which includes the concatenated features before any linear layer, and a style layer which is simply the magnitudes, and then uses a content and style layer following each nonlinearity. This network produces distinctly different stylizations to Ulyanov's original network, despite having similar parameters, often including quicker and busier temporal changes in content and style. The stylization also tends to produce what seems like higher fidelity syntheses, especially in lower frequencies, despite having the same sample rate. Lastly, this approach also tends to produce much less noise than Ulyanov's approach, most likely due to errors in the phase reconstruction/lack of phase representation.

Every other combination of input manipulations we tried tended towards a white noise signal and did not appear to drop in loss. The only other network that appeared to produce something recognizable, though with considerable noise was using the magnitude and unwrapped phase differential information with a kernel size 2 height convolutional filter. We could not manage to stylize any meaningful sounding synthesis using the activations in a WaveNet decoder or NSynth encoder.

4 Discussion and Conclusion

This work explores neural audio style transfer of a time domain audio signal. Of these networks, only two produced any meaningful results: the magnitude and unwrapped phase network, which produced distinctly noisier syntheses, and the real, imaginary, and magnitude network which was capable of resembling both the content and style sources in a similar quality to Ulyanov's original approach, though with interesting differences. It was especially surprising that we were unable to stylize with NSynth's encoder or decoder, though this is perhaps to due to the limited number of combinations of layers and possible activations we explored, and is worth exploring more in the future.

[2]Further details are described in the Supplementary Materials

[3]https://github.com/pkmital/neural-audio-style-transfer

References

[1] D. Ulyanov and V. Lebedev, "Audio texture synthesis and style transfer," 2016.

[2] L. A. Gatys, A. S. Ecker, M. Bethge, and C. V. Sep, "A Neural Algorithm of Artistic Style," *Arxiv*, p. 211839, 2015.

[3] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky, "Texture Networks: Feed-forward Synthesis of Textures and Stylized Images," 2016. [Online]. Available: http://arxiv.org/abs/1603.03417

[4] D. W. Griffin and J. S. Lim, "Signal Estimation from Modified Short-Time Fourier Transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[5] L. Wyse, "Audio Spectrogram Representations for Processing with Convolutional Neural Networks," in *Proceedings of the First International Workshop on Deep Learning and Music joint with IJCNN*, vol. 1, no. 1, 2017, pp. 37–41. [Online]. Available: http://arxiv.org/abs/1706.09559

[6] Z. Průša and P. Rajmic, "Toward High-Quality Real-Time Signal Reconstruction from STFT Magnitude," *IEEE Signal Processing Letters*, vol. 24, no. 6, pp. 892–896, 2017.

[7] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, K. Wilson, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson, "CNN Architectures for Large-Scale Audio Classification," *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4–8, 2016. [Online]. Available: http://arxiv.org/abs/1609.09430

[8] J. Engel, C. Resnick, A. Roberts, S. Dieleman, D. Eck, K. Simonyan, and M. Norouzi, "Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders," in *Proceedings of the 34th International Conference on Machine Learning*, 2017. [Online]. Available: http://arxiv.org/abs/1704.01279

[9] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A Generative Model for Raw Audio," *arxiv*, pp. 1–15, 2016. [Online]. Available: http://arxiv.org/abs/1609.03499

5 Supplementary Material

5.1 Input Data

Each of the shallow untrained networks we used take as input a raw audio signal sampled at 22050 Hz with a frame size of 2048 samples, an alpha of 0.01, and use 150 iterations of the Adam optimizer. We explored manipulations in sample rate including [44100, 22050, and 16000]. For frame size (DFT size was always set to half frame size with no padding or centering), we explored [1024, 2048, 4096, 8192], with hop sizes of [128, 256, 512]. The resulting projections from a discrete Fourier basis set were then sliced to half width to remove their symmetric projections.

For the NSynth and WaveNet networks, we used the native sampling rate they were trained on, 16000 Hz. For the shallow untrained networks, we explored a combination of networks that varied in their initial input processing, depth, and the number layers and information we used for content and stylization. We tested networks which incorporated the real, imaginary, magnitude, and phase information of an audio source signal's DFT, as computed with a computational graph capable of automatic differentiation. This enabled us to apply stylization by optimizing an input noise signal, while keeping the rest of the network untrained.

5.2 Network

Ulyanov's original stylization network uses depth-wise convolution as the first layer operating on the magnitudes. We employ the same technique here, except using combinations of the real, imaginary, magnitude, and phase information as input, stacked along the height dimension. For kernel sizes, we tried a variety of widths, including $[4, 8, 16]$, and for heights, depending on the number of components included we tried $[1, H]$, where H is the total number of components included in the model. For instance, for a model incorporating real and imaginary components, we set $H = 2$, and stacked the real and imaginary comopnents in rows. For number of layers, we tried $[1, 2, 3]$. And finally, for representing phase, we tried the original phase, the phase differential, and the unwrapped phase differentials. We used a stride of 1 and a ReLu activation for all convolutional layers, and followed the weight initialization used by Ulyanov's baseline audio stylization network. Finally, we explored alphas including [0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001].

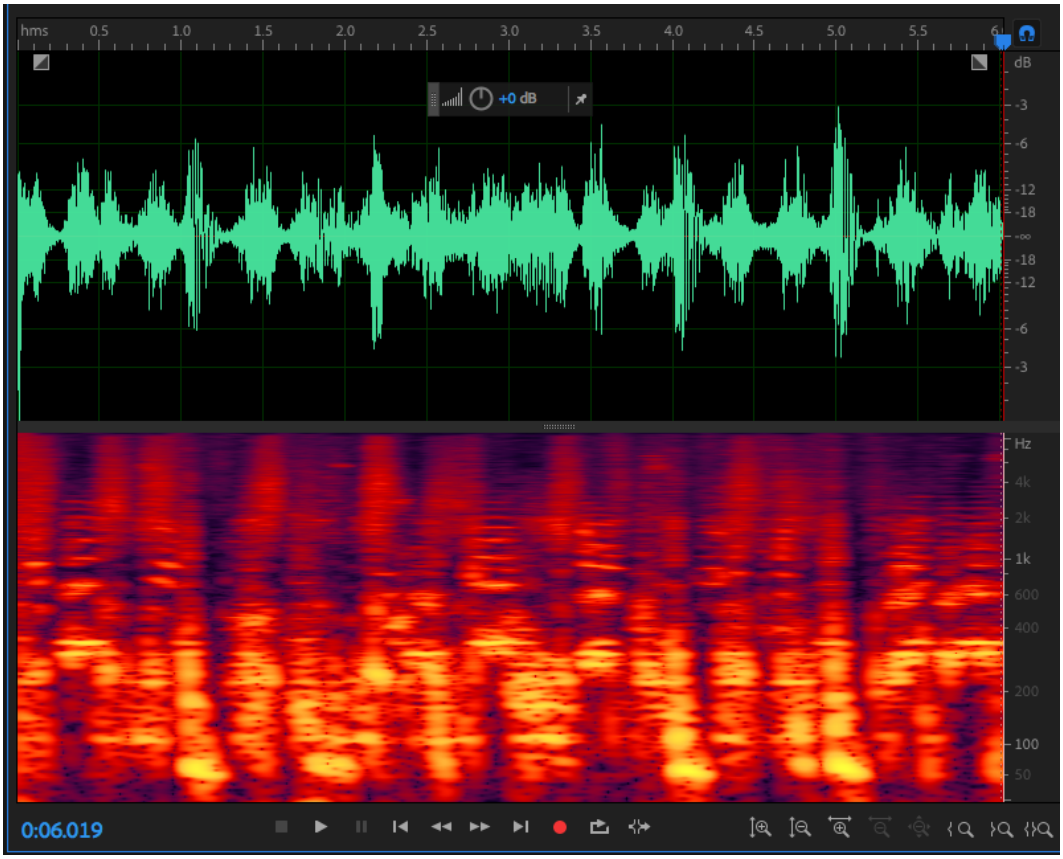

Figure 1: Example synthesis optimizing audio directly with both the source content and style audible.