Style Transfer for Prosodic Speech

Anthony Perez
Stanford University
Stanford, CA - 94305
aperez8@cs.stanford.edu

Chris Proctor
Stanford University
Stanford, CA - 94305
cproctor@stanford.edu

Archa Jain
Stanford University
Stanford, CA - 94305
archa@stanford.edu

Abstract

Neural style transfer on images captures independent representations of style and content, such that style from one image can be transferred onto content from another. We present an extension of style transfer to speech using spectrogram representations of speech as the image. We report success in transferring low-level textural features, but difficulty in transferring high-level prosody such as emotion or accent. We discuss several improvements made to the model and propose others which could improve results. Comparing our approach with other recent results, we hypothesize that a system capable of prosodic style transfer will require the incorporation of a language model.

1 Introduction

Neural Style transfer in images aims to capture the "style" or texture of an image and apply it to another. An example with images can be seen in Figure 1, where the style of Van Gogh's *Starry Night* is applied to an image of the San Francisco skyline. We can see in the image that the high level features and contours, like building edges and brightly-lit areas, are preserved from the content image, but local textural features, like the overcast sky, are effaced by van Gogh's characteristic swirls.

In the context of speech, style transfer would mean reproducing the content of an audio clip in another speaker's voice. In this work, we study the effects of applying the ideas from the work done on style transfer of images to audio, and explore what might be a successful approach to achieve prosodic style transfer.

Figure 1: The style from van Gogh's *Starry Night* is transferred onto a photograph of San Francisco.

2 Background

Our work is largely inspired by the neural style transfer work done on images (Gatys et al., 2015). The methods in this paper are explained in more detail in the following sections, but at a high level, this paper proposes using a pretrained CNN to extract a content representation from one audio spectrogram and a style representation from another spectrogram and then produce a spectrogram (from which audio can be recovered) that matches the content and style representations of the inputs.

Pretrained models are plentiful in the vision domain, but pretrained speech models are scarce. Thus, we seek to train our own model that we will then use to extract content and style representations from audio data. We explored using autoencoders for unsupervised pretraining of our CNN over speech samples.

Autoencoders are models that learn a lower dimensional representation of the data that can be used to recover an approximation of the original input to the model. The work in (Dai and Le, 2015) shows that recurrent autoencoders based on the *seq2seq* model are useful in pretraining LSTM weights for subsequent use in supervised tasks. The method described in (Dai and Le, 2015) uses

an LSTM-based encoder, which encodes the input sequence into a fixed length vector and an LSTM-based decoder, which recovers the input sequence from the fixed length vector encoding. While recurrent autoencoders have been shown to perform demonstrably well in sequence based, tasks, the base approach behind neural style transfer requires a pretrained convolutional neural net. Thus we have experimented with convolutional autoencoders.

3 Dataset

We use the VCTK corpus (Veaux et al., 2010) for pretraining the autoencoder, which consists of 109 English speakers, each reading 400 sentences from newspaper articles sampled at 48kHZ. While the speech clips in the corpus are of varying lengths, we clip them at one second when training the autoencoder. We separated our training and testing data based on speakers, so there are no speakers common to both sets. We preprocess each audio clip to using a Short-time Fourier transform (FFT window size = 2048) to create a spectrogram, which was more conducive than the raw audio signal to training a CNN.

4 Methods

4.1 Autoencoder

Style transfer requires a pretrained neural network from which to extract features. To this end, we train a convolutional autoencoder on the spectrograms of our input data. We use symmetric 4-layer CNNs as the encoder and decoder for our autoencoder model (Fig 2). In the baseline model, each convolutional layer is followed by a ReLU nonlinearity and a Batch Normalization layer.

The model is trained by minimizing the L2 norm of the difference between the source spectrogram and its reconstruction. After the model is trained, features are extracted from the layers in blue in Figure 2.

We experimented with alternative autoencoder architectures, including a recurrent autoencoder based of the *seq2seq* model and a convolutional autoencoder that convolved only in the time dimension of the input spectrogram. However, we found that a convolutional autoencoder that convolved in both the time and spectrogram channel dimensions had the best performance in reconstructing the input.

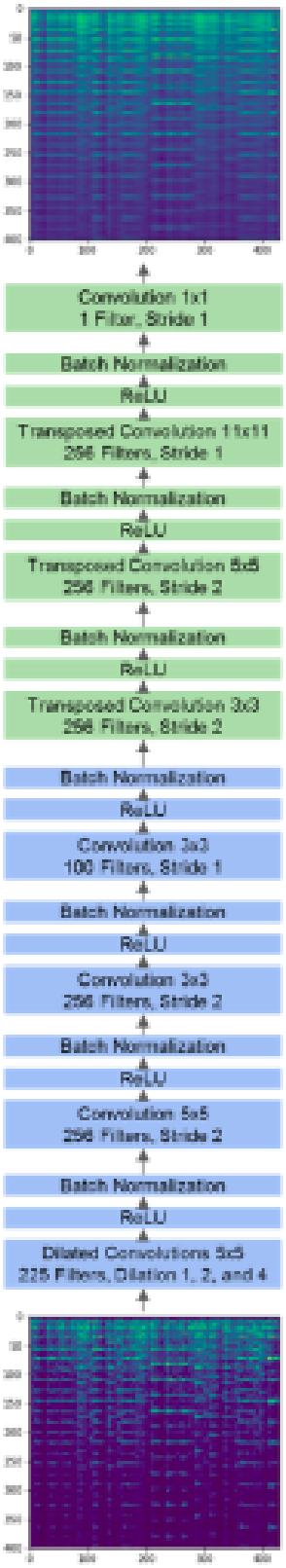

Figure 2: Convolutional autoencoder trained offline using reconstruction loss. Style and content features were extracted from the encoding layers, shown in blue.

4.2 Style Transfer

To perform style transfer, we first assume that the following is available: a pretrained neural network, a content reference, and a style reference. The objective is to combine content and style from the respective references. Figure 3 shows a visualization of this workflow.

At a high level, the style transfer begins by extracting features from some layer in the network for both the content reference ($F_{content}$) and style reference (F_{Style}) reference and by initializing the output spectrogram from noise. (In the next section, we explore the effect of extracting style features from different layers of the network.)

We then define the loss functions for the content and style of the output we seek to generate as such:

$$L_{content} = ||F^{Opt}_{content} - F_{content}||^2$$
$$L_{style} = ||(F^{Opt}_{style})^T F^{Opt}_{style} - (F_{style})^T F_{style}||^2$$

where $F_i \in \mathbb{R}^{S_i \times C_i}$, S_i is the product of the width and height of the layer from which F_i is extracted, and C_i is the number of feature maps in the layer from which F_i is extracted.

The content loss encourages the output to match the content features at every time step while the style loss encourages the model to match the covariance statistics across convolutional filters in the style layer (or layers). The intuition is that the autoencoder has been pretrained to optimize for more and more compact representations at each layer; therefore convolutional filters in early layers encode fine-grained and widely-distributed features, while filters in deeper layers encode abstract, context-sensitive, structural features. The style loss is computed from the Gram matrix of the filter values rather than the values themselves to allow for various representations of content at deeper levels while insisting on certain distributions of surface-level features. The Gram matrix is akin to the covariance of the CNN channels across the time and spectrogram channel dimensions.

Finally, the ADAM optimizer is used to minimize a weighted sum of the content and style losses for the output spectrogram, with the weights being used for tuning the final performance.

5 Results and Analysis

5.1 Autoencoder

The trained autoencoder performs fairly well at reconstructing the audio clips. Figure 4 shows original (left) and reconstructed (right) spectrograms for example content and style images. As is expected with lower dimensional representations of content, we lose some detail, but the reconstructed audio is recognizable as a reproduction of the original.

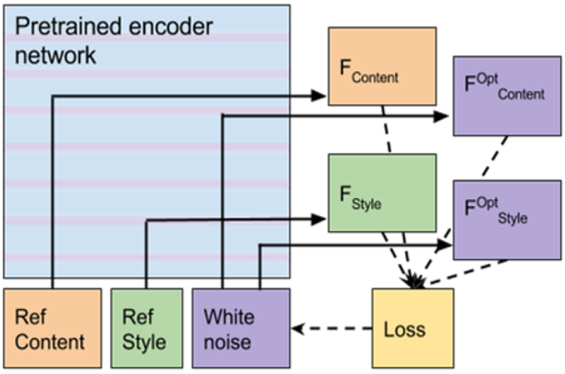

Figure 3: In style transfer, content features are extracted from reference audio and from an optimization target, using a deep layer in the network. Style features are extracted from another reference and the target, using one or more shallow layers. These features contribute to a loss function, which is used to train the target to match the content of one reference and the style of the other.

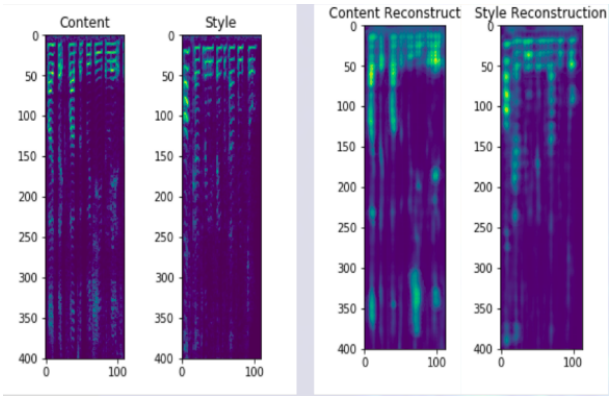

Figure 4: Input spectrograms and reconstructions from our previously-trained model.

5.2 Style Transfer

Audio style transfer was successful when we drew our style features exclusively from the first layer of the model, however the texture transferred is not recognizable as prosodic. Human judges clearly recognize the content from then resulting audio clips, while prosodic features from the style source are clearly not transferred. Nevertheless, listeners

describe the resulting audio as having a continuous buzz with the "flavor" or "color" of the style sample. Changes to the weighting of content and style loss yield the predictable effect of increasing or attenuating this buzz. An example of spectrograms for the content, style, and output of the style transfer process can be see in Figure 5.

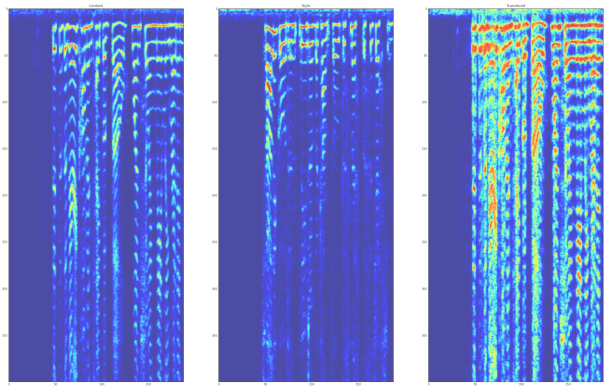

Figure 5: This figure shows the content (left) and style (middle) spectrograms, as well as the spectrogram of the output of the style transfer process (right).

The following strategies improved the model, as judged by the content and style quality of output audio:

- Adding dilated convolutions (over both the time and frequency dimension) at the bottom layer in an attempt to capture less-local texture (Sercu et al., 2016).
- Optimizing log-values of the frequency spectrogram, and then re-exponentiating the results before reconstructing waveforms.
- Learning rate decay during optimization.
- Tuning weight parameters between losses to achieve balance between content and style.
- Using the ADAM optimizer (Kingma and Ba, 2014) rather than Gradient Descent (GD), GD with Momentum, or RMSProp (Tieleman and Hinton, 2012).

One of the main challenges we faced was optimizing the generated audio during style transfer. We tried the following methods to improve the optimization of the generated audio.

- Removing batch normalization.
- Using leaky ReLU to propagate error signal.
- Computing the gram matrix (covariance of feature values in style layers) only across time rather than across time and spectrogram channels.
- Adding an L2 loss regularization on the magnitude of frequency channels during optimization.
- Clipping the values of the generated audio between iterations.
- Initializing the generated audio to the content spectrogram rather than noise.

Of these optimization tricks, we found that clipping the values of the generated audio to be extremely helpful, but none of the other methods had a significant impact. We were unable to achieve stable gradient descent when using layers beyond the first as style features. Figure 6 is a loss plot for two style transfer instances on the same content and style references. It illustrates the optimization difficulties that occur when optimizing a layer beyond the first.

6 Future Work

Our results validate the use of image style transfer for audio spectrograms of speech, while surfacing challenges in separating style representation from content and optimization. The style we were able to isolate from speech was generally uniformally distributed, similar to results achieved by (Ulyanov and Lebedev, 2016) in modeling audio textures such as typing or musical harmonies. In this section, we first suggest incremental improvements that appear promising for our approach, and we then consider possible responses to a more fundamendal issue in the relationship between style and content in speech.

Our most immediate obstacle was in achieving stable gradient descent when using layers beyond the first for style features. We hypothesize several possible changes beyond those we describe above. First, we could change the loss function on the pretrained autoencoder to optimize for meaningful feature representations of style. A representation of style isolated from content should not vary much from sentence to sentence for the same speaker, assuming the same speaking context and emotional state. Drawing on the VCTK corpus (Veaux et al., 2010), we could minimize style difference between pairs of different sentences from

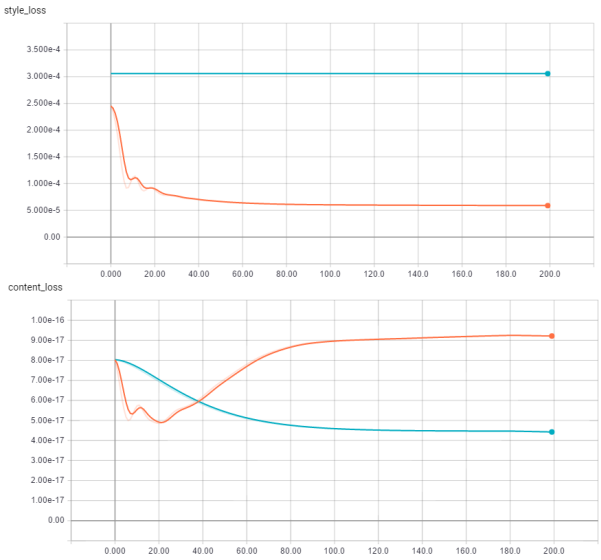

Figure 6: This figure shows the loss plots of both the style (top) and content (bottom) losses as the style transfer optimization proceeds. The orange curve specifies style transfer where the style layer is the first layer in the network. The blue or teal curve specifies style transfer where the style layer is the second layer in the network. The horizontal axis is the optimization step while the vertical axis is the loss value. Note that style was weighted very highly relative to content to illustrate the optimization issues that arise when using a layer beyond the first.

the same speaker while minimizing content difference between the same sentence spoken by different speakers. Since we use weights generated from this network to influence the style transfer loss, a network that is trained to capture a better representation of the style could potentially improve performance.

It may also be that spectrograms are not well-suited for representating speech style. If the important characteristics of speech could be parametrized, this would provide a much more compact representation and thus a more easily-trained model. For example, emotional speech can be characterized to some extent by coarse features such as mean F_0, mean RMS, and F_0 and RMS within the nuclear stressed syllable of an utterance (Hirschberg et al., 2003). Pitch and speaking rate are also important parameters of speakers, and could be the basis for warping the input signal to surface more fine-grained individual differences. Our inclusion of convolutional dilation in the first layer of our model was an attempt to capture some of these features, but it seems likely that preprocessing at the signal processing level would be more effective.

In addition to preprocessing the input, we could change our model to encourage it to learn and encode these parameters. Presently, the style which our model is capable of transferring is too local. We hypothesize that including total variation loss (Chambolle, 2004) would optimize for smoother output at the fine-grained level with change concentrated in higher-level structures such as formants.

Since the resulting audio does not sound like natural speech, and we have no measure of "real speech" in the model, adding a generative adversarial network (Goodfellow et al., 2014) to select for natural-sounding output might be helpful. Specifically, the output of the style transfer process should be optimized to fool the discriminator of a well trained generative adversarial network.

Beyond these possible incremental improvements, our difficulty in isolating higher-level speech prosody, such as accent or emotion, suggests a fundamental difference between how we see artistic visual style and how we hear style in speech. While visual style is somewhat sensitive to the context of image content–the application of Van Gogh's swirls to a city scene maintains crisp edges and the intensity of foreground color contrast (Figure 1)–we can apply different styles without attending much to what the image depicts. In contrast, it may be that how we speak is deeply intertwined with what we are saying, or at least that models of speech style will take content as input in determining the probability of stylistic features.

If this is true, we anticipate that progress on modeling and synthesizing style in speech (where style broadly includes dialect, emotion, and other prosody) will employ a language model. Our model represents style as correlations between the values of convolutional filters, each of which learns to recognize features in the underlying speech representation. The inclusion of a language model would allow associations between stylistic features and content context. It would thus be much easier to capture contextual stylistic features such as uptalk, emphatic stress on adjectives, and semantically-appropriate pauses. This appears to be the strategy employed by Lyrebird, a startup which has demonstrated the most successful synthesis of speech style to date. Lyrebird has not

disclosed its methods, but the company's founders were the lead authors of Char2Wav (Sotelo et al., 2017), an end-to-end speech synthesizer which stacks a RNN-based vocoder on top of an encoder-decoder language model with attention. While this approach tackles a different problem, it illustrates the potential utility of incorporating a language model into a style transfer system for speech.

7 Conclusion

The Spoken Language Processing community appears to be on the cusp of modeling and synthesizing prosodic speech. The ability to systematically analyze spoken prosody will be a transformative tool in sociolinguistic research investigating how language practices shape (and are shaped by) communities and cultures. These systems will allow dialogue agents to express personalities and emotions, as well as much more effectively engaging in speech acts such as making requests, establishing social situations, and in turn eliciting emotional speech from their human interlocutors. As such, dialogue systems may participate more fully as actors in social networks (Latour, 2005) in which agency, power, and personhood are assigned to human and nonhuman alike. The participation of computers as agents in human social worlds raises profound ethical and ontological questions.

References

Antonin Chambolle. 2004. An algorithm for total variation minimization and applications.

Andrew M. Dai and Quoc V. Le. 2015. Semi-supervised sequence learning. *CoRR* abs/1511.01432. http://arxiv.org/abs/1511.01432.

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2015. A neural algorithm of artistic style. *CoRR* abs/1508.06576. http://arxiv.org/abs/1508.06576.

I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. 2014. Generative Adversarial Networks. *ArXiv e-prints* .

Julia Hirschberg, Jackson Liscombe, and Jennifer Venditti. 2003. Experiments in emotional speech. In *ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980. http://arxiv.org/abs/1412.6980.

Bruno Latour. 2005. *Reassembling the social: An introduction to actor-network-theory*. Oxford university press.

Tom Sercu, Christian Puhrsch, Brian Kingsbury, and Yann Lecun. 2016. *Very deep multilingual convolutional neural networks for LVCSR*, Institute of Electrical and Electronics Engineers Inc., United States, volume 2016-May, pages 4955–4959. https://doi.org/10.1109/ICASSP.2016.7472620.

Jose Sotelo, Soroush Mehri, Kundan Kumar, Joao Felipe Santos, Kyle Kastner, Aaron Courville, and Yoshua Bengio. 2017. Char2wav: End-to-end speech synthesis .

Tijmen Tieleman and Geoffrey Hinton. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning.

Dmitry Ulyanov and Vadim Lebedev. 2016. Audio texture synthesis and style transfer. https://dmitryulyanov.github.io/audio-texture-synthesis-and-style-transfer/.

Christophe Veaux, Junichi Yamagishi, and Kirsten MacDonald. 2010. Cstr vctk corpus http://homepages.inf.ed.ac.uk/jyamagis/page3/page58/page58.html.