Audio Translation and Style Transfer

Murad Hüdavendigar Bozik - s2619822

May 2021

1 Abstract

Voice translation task is an important task in terms of style transfer on audio signals. New audio signals can able to generated using Generative Adversarial Networks (GANs). In this project, I investigated, studied and experimented with GAN architectured method called MELGAN-VC[3] introduced by Marco Pasini. This method is relatively new and it uses GAN models together with a siamese network on spectrograms.

2 Introduction

In the original paper two tasks are introduced. First is voice conversion and second is style transfer. These two tasks are actually the same. Voice translation aims to transfer the speaker voice while keeping the content of the sentences. Audio style transfer is introduced on music signals. It aims to convert music genres without changing the original instruments. In this method MELGAN-VC the author succeed this using GAN model on spectrograms. There are mainly two types of approach regarding these tasks. One is feature based methods and the other is spectrogram based methods. Some features can be extracted from audio signals and these features can be used to train neural networks. The main objective is to create a mapping function between source and target signals. Spectrogram based methods operates on image version of audio signals. Convolutional Neural Network (CNN) architectures and GANs along with other methods have been investigated. Passini developed GAN architecture together with a siamese network.

3 Related Work

GAN architecture proved its success on generating high quality realistic images, applying image translation. Similar to audio translation, image translation requires to preserve some content in source domain and translate it to target domain. Cycle-GAN used cycle consistency as a constraint. It calculated pixel-wise differences and forced source image to adapt target domain. This method limits flexibility and also requires paired datasets. Travel-GAN introduced image-to-image translation using transformation vector learning. These methods mainly focused on latent space representations and its statistics. Because latent space the core of this mapping process. Generator can map from source to target if the latent space is well-representative for target domain.

Audio translation can also be used in this domain using their spectrogram images. However, there are not wide range of datasets regarding paired audio signals. Also, is it hard to use their spectrograms because of the alignment requirements.

4 Method

Pasini used siamese network and introduced TraVeL loss to keep content information during translation task. Figure 1 shows model components and the training process of the model.

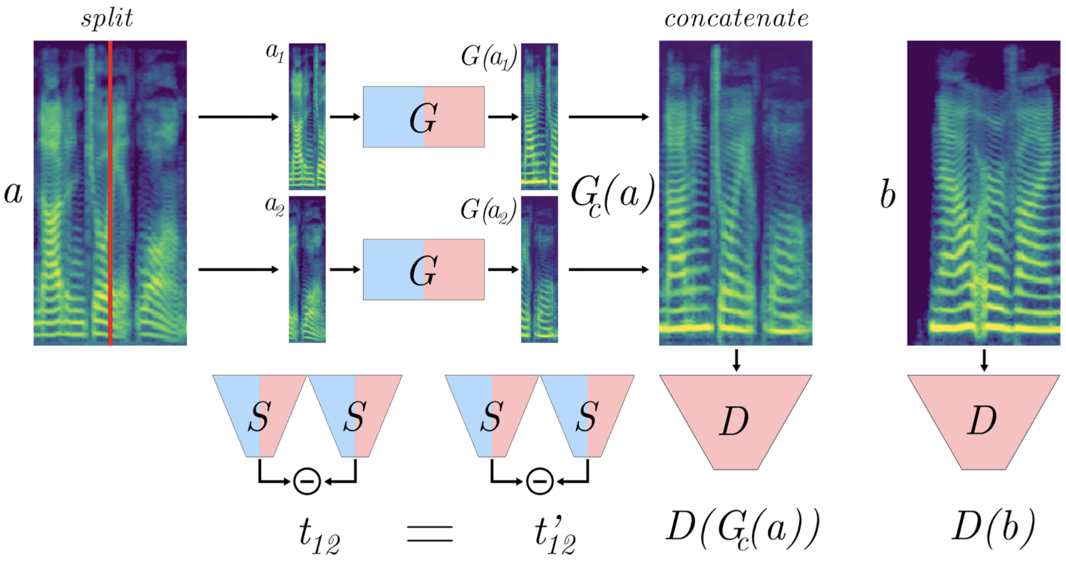

Figure 1: Model

Different audio signals provide different length spectrograms. In order to leverage spectrograms of arbitrarily long audio signals, the method converts audio signal into mel-spectrogram, divides it into multiple parts and last part is padded to read a certain length. Splitting operation which is showed in Figure 1 indicates that operation performed on one part of spectrogram. The model includes single generator, discriminator and additional siamese network.

For training ARCTIC dataset is used. This dataset includes 1132 sentences pronounced by male and female speakers. With this dataset training has been conducted on paired samples for audio translation task.

For male to female audio translation, first spectrum from male speaker split into two parts and individually processed by generator. These split parts and generated parts are fed into siamese network and calculated transformation vectors. From these vectors TraVeL loss is calculated and this is used for siamese and generator networks.

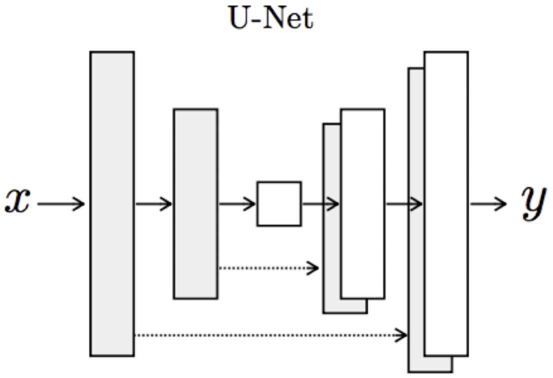

Figure 2: U-net generator architecture. Taken from the GAN with Python book[2]

Generator network is u-type generator similar to the model in paper[1]. Figure 2 shows the architecture of U-net generator.

Encoder: CSN256-CSN256-CSNS256
Decoder: UCSN256-UCSN256-CSNT

CSN denote to Extended_Convolution-BatchNorm-LeakyReLU layer. Convolution2D layers are extended with an additional weight and using this weight spectral normalization is computed. UCSN denotes to Upsampling-CSN-Concat layer. First, input is fed to upsampling layer then follwed by CSN layer which is explained previously. Then, output is the concatenation of result with respective layer in encoder part. Last, CSNT denotes to extended Conv2DTranspose layer.

Discriminator architecture is similar to PatchGAN discriminator. It includes 3 Extended Convolution2D layers and one Dense layer. Siamese network architecture is same as discriminator. The only difference is its convolution layers are not extended version.

The losses used in this architecture is shown below;

$$\mathcal{L}_D = \mathcal{L}_{D,adv}$$
$$\mathcal{L}_G = \mathcal{L}_{G,adv} + \alpha\mathcal{L}_{G,id} + \beta\mathcal{L}_{(G,S),TraVeL}$$
$$\mathcal{L}_S = \beta\mathcal{L}_{(G,S),TraVeL} + \gamma\mathcal{L}_{S,margin}$$

This model reduces the limitations of cycle consistency. Using transformation vector helps generator to preserve contents from source domain. The weights of losses can be tuned. In inference period, audio signals turned into mel-spectrograms and the generated spectrograms turned back to audio files using Griffin-Lim algorithm.

5 Result

Even though it is not shared in the paper, I managed to find the author's code on github separately. This code is one jupyter notebook and designed with hard coded parameters. I created a repository on github and made this implementation more readable in a more structured way [1]. In this repository there are links to original repository and the paper itself. The problem I encountered was transfering the parameters and model itself between code partitions. Updating the model parameter caused duplication of the models in memory. I overcome this problem by implementing every parameters and model as an argument. StyleGAN repository uses EasyDict, an extended dictionary to provide dot notation, as its arguments object. I used this class and turned argument parser into EasyDict version. By doing this I managed to pass parameters between helper classes and also added models as parameters. This solved the problem of model duplication and updating its weights.

I have trained the model in a way to translate the content from male speaker to female speaker. The latest model is added to the repository. It is possible to perform inference using inference.py file. The most important feature of this method is flexibility.

This method splits audio signals into smaller spectrograms. By doing this it gains the capability of processing arbitrarily long audio signals. This method is also independent from linguistic features of the dataset it is trained on. I performed various tests with sentences from different languages. The results are quite surprising. The model is trained only with english sentences, however it can perform conversion task on other languages as well as on english. Turkish samples includes turkish letters which aren't present in english. But the result was best on turkish sentences. Some sentences were poorly translated such as Spanish sample. It included "rr" sound and this is translated quite bad. The samples can be found on repository.

[1] https://github.com/MuradBozik/audio-style-transfer

6 Conclusion & Future Work

To conclude, new audio signals with source content is possible with generative models. However, generative models needs high amount of data to represent target domain. With more data-centric approach generative models will perform as good as image domain. For the future work, I would like to try to use progressive GAN structure to create mel-spectrgorams. With very large amount of data it is possible to represent target domain very good. I have been working on noise optimization and blending effects on image domain and the effect is really big. With blending process the product can be pushed to create more reliable results. If we can acquire large amount of data of mel-spectrograms, generative models can be used to create mel-spectrograms of meaningful words, sentences. With blending translation task might be performed successfully without retraining the model for other speakers.

References

[1] Phillip Isola et al. "Image-to-image translation with conditional adversarial networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2017, pp. 1125–1134.

[2] Jason Brownlee. *Generative Adversarial Networks with Python: Deep Learning Generative Models for Image Synthesis and Image Translation.* Machine Learning Mastery, 2019.

[3] Marco Pasini. "Melgan-vc: Voice conversion and audio style transfer on arbitrarily long samples using spectrograms". In: *arXiv preprint arXiv:1910.03713* (2019).