VOICE CONVERSION

by using CycleGAN

Selahaddin HONİ
İsmail Melik TÜRKER
İmran Çağla EYÜBOĞLU

MLSP Term Project - Fall 2020

CONTENT

The Aim & Reference Paper

Brief Explanation of Reference Paper

Dataset

Implementation & Training

Result

A project publish page is deployed for the results, related links and more
mlsp2020.github.io

The Aim & Reference Paper

In the project, it is aimed to transfer the trained voice style of a famous person to given input voice.

A project which competed in Voice Conversion Challenge 2016 is selected as a reference work.

Takuhiro Kaneko and Hirokazu Kameoka

Parallel-Data-Free Voice Conversion Using Cycle-Consistent Adversarial Networks (2017)

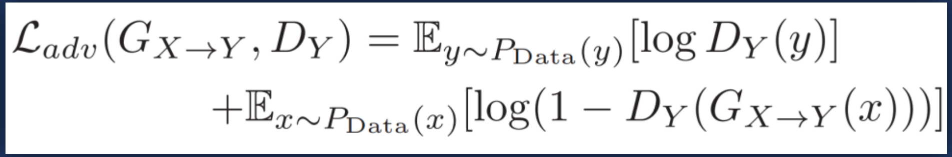
We tried to implement an adaptation of this network to Turkish language.

İSTANBUL TEKNİK ÜNİVERSİTESİ 1773

Brief Explanation of Reference Paper | CycleGAN 1/3

Purpose of the study is to obtain mapping by train source and target data which are not parallel. CycleGAN is preferred concept as a base network which generates mapping with the benefits of two essential loss term; adversarial loss and cycle-consistency loss.

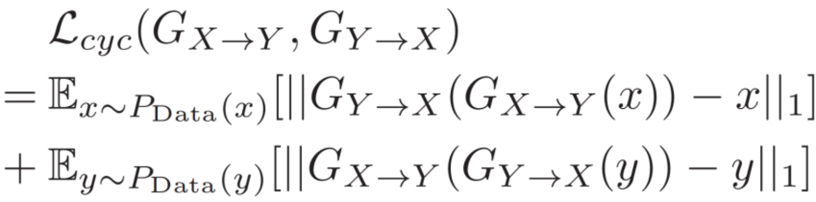
Adversarial Loss



$$\begin{aligned}\mathcal{L}_{adv}(G_{X\to Y}, D_Y) &= \mathbb{E}_{y\sim P_{\text{Data}}(y)}[\log D_Y(y)] \\ &+ \mathbb{E}_{x\sim P_{\text{Data}}(x)}[\log(1 - D_Y(G_{X\to Y}(x)))]\end{aligned}$$

Adversarial loss defines the difference between converted data and target data. As smaller as loss means the distribution of the converted data is more similar to the target data distribution.

Cycle-Consistency Loss



$$\begin{aligned}&\mathcal{L}_{cyc}(G_{X\to Y}, G_{Y\to X}) \\ &= \mathbb{E}_{x\sim P_{\text{Data}}(x)}[||G_{Y\to X}(G_{X\to Y}(x)) - x||_1] \\ &+ \mathbb{E}_{y\sim P_{\text{Data}}(y)}[||G_{X\to Y}(G_{Y\to X}(y)) - y||_1]\end{aligned}$$

Cycle consistency loss updates the generator models for each iteration considering the difference between generated data and input data.

Brief Explanation of Reference Paper | CycleGAN-VC

Gated CNN and identity mapping loss methods are used in order to apply CycleGAN for parallel-data-free VC.

Gated CNN

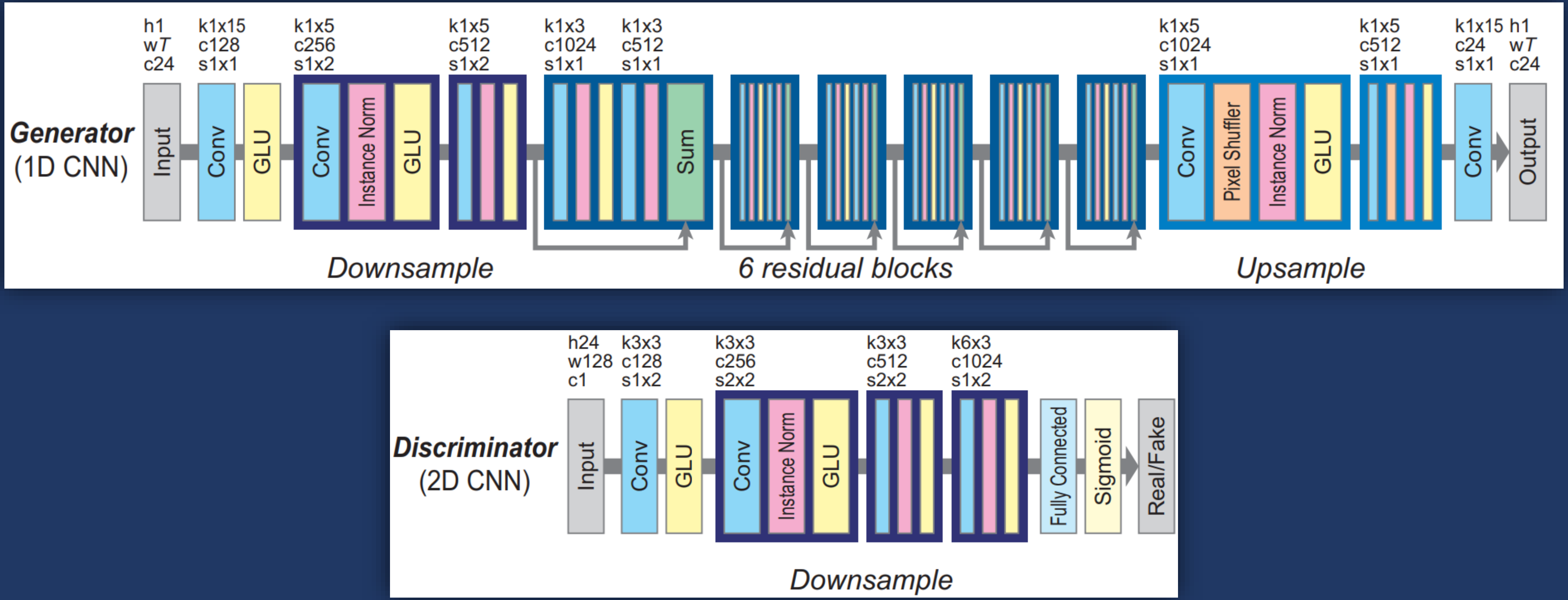
The data-driven activation function is gated linear units (GLUs). Data transmission from layer to layer is elaborately realized considering the previous layer by means of the Gated CNN model.

$$\boldsymbol{H}_{l+1} = (\boldsymbol{H}_l * \boldsymbol{W}_l + \boldsymbol{b}_l) \otimes \sigma(\boldsymbol{H}_l * \boldsymbol{V}_l + \boldsymbol{c}_l)$$

Identity-Mapping Loss

Cycle-consistency loss is not a sufficient solution for mappings to preserve linguistic-information. Identity-mapping loss provides generator to obtain mapping which preserves the arrangement between input and output.

Brief Explanation of Reference Paper | CycleGAN-VC Network 3/3



“Network architectures of generator and discriminator. In input or output layer, h, w, and c represent height, width, and number of channels, respectively. In each convolutional layer, k, c, and s denote kernel size, number of channels, and stride size, respectively.”

Dataset

Source

Google's text-to-speech voices are used to generate 13 audio clips (each in a duration of approx. 40 secs) in a total of at least 8 minutes for each speaker.

- Female Speaker: *WaveNet Turkish Female voice G*
- Male Speaker: *WaveNet Turkish Male voice E*

Target

Similarly, 13 audio clips in a total duration of 8.8 minutes of Turkish news-presenter Ece Uner's speech is chosen.

Our custom dataset is shared online, link is given in the project web page.

Implementation & Training

We highly utilized from *Lei Mao's work* [leimao@github] while implementing this project.

Some updates are required to reduce the time-consuming process as a main reason. Detailed changelog is given in report; yet, here are the significant ones for performance gain:

Performance

- *After realized the training per epoch is so slow because of model-saving and validation operations; 'check_epoch' parameter is added to control them.*
- *With the help of another if condition, epoch duration is decreased from approx. 55 seconds to 4 seconds. (NVIDIA Tesla T4)*
- *Validation functions for conversion from B-to-A is removed. (We only need A-to-B)*

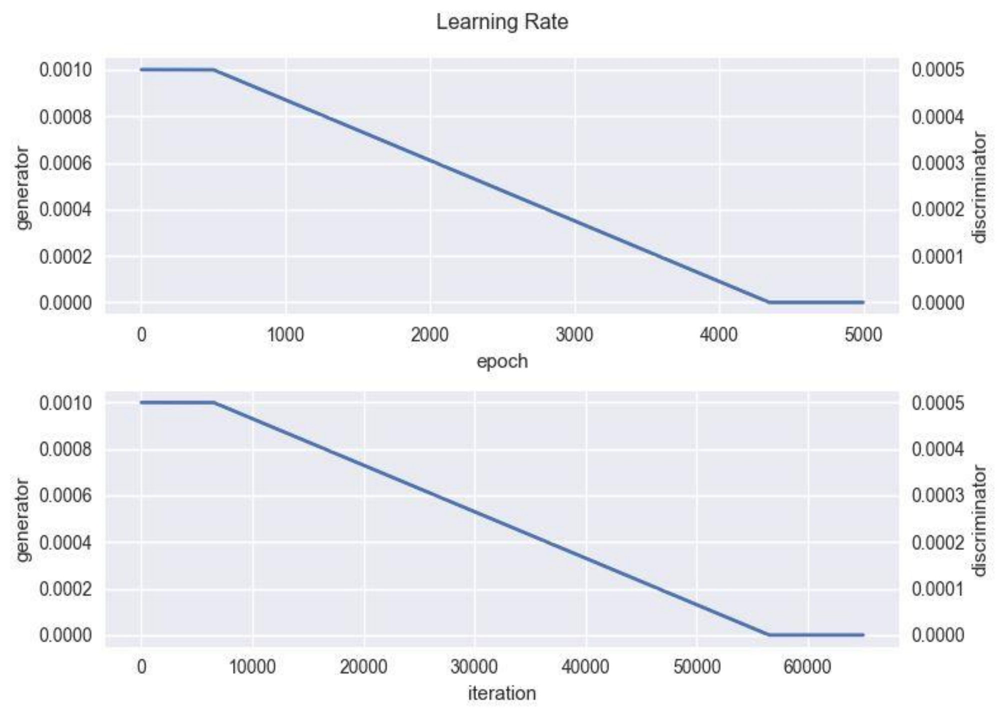
Two models, female-to-female and male-to-female, trained for voice conversion in Google's Colaboratory; the training for one model took approximately 5.5 hours with NVIDIA Tesla T4 GPUs after above updates were applied.

Implementation & Training

Hyper-Parameter Tuning

In the reference implementation, iteration size dependent on the number of given training audio files not the length; therewithal, learning rate decays with iterations to converge to global minima.

However, our dataset and file organization are different and old hyper-parameters result in stop of learning. Therefore, the figure on the right is the plot of new learning rates for both generator and discriminator over growing epochs and iterations.



Result

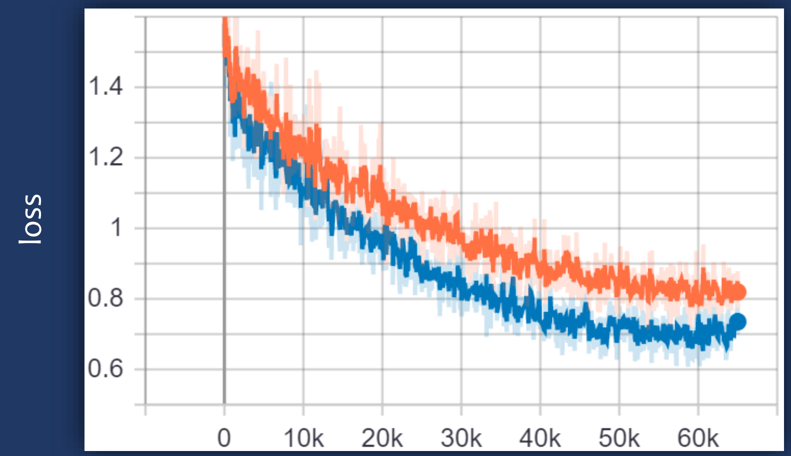
It is not possible to demonstrate audio samples on this PDF presentation with sound; thus, synthesized fake voices are uploaded to deployed project web page. The demo allows you to follow the progress on the conversion of input voice over number of epochs trained models.

mlsp2020.github.io

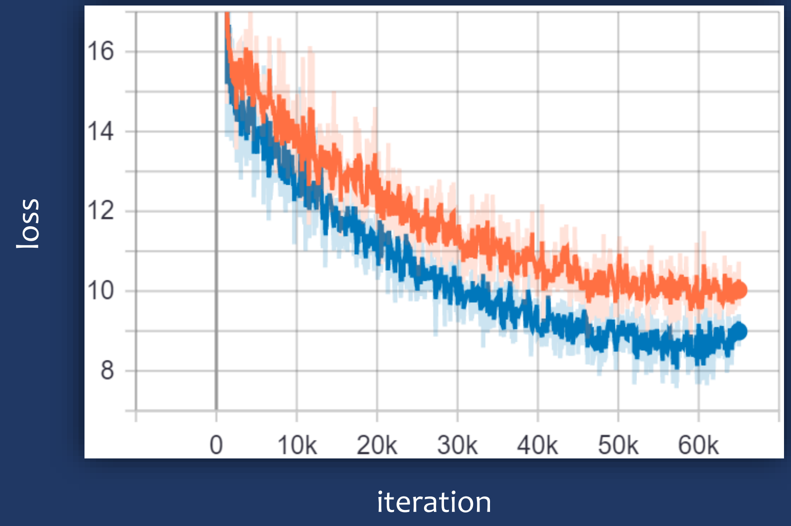
The figures on the right, a simple evidence of our successful training that loss reduces over growing iterations. However, detailed observation is out of scope of this short presentation.

More figures and comments included in report

Cycle Loss



Generator Loss


loss
10k
20k
30k
40k
50k
60k
iteration