Generative Adversarial Networks for Speech Applications

Nagaraj Adiga

University of Crete, Dept of Computer Science
nagaraj@csd.uoc.gr

April 22, 2019
OUTLINE

1. Introduction
2. Speech Enhancement
3. Voice conversion
4. Conclusion
GANs in Speech applications

- Speech Generation
  - Speech Enhancement
  - Speech Synthesis
  - Voice Conversion
  - Speech dereverberation

- Speech Recognition
  - Speech-to-text (ASR)
  - Speaker Recognition
  - Emotion Recognition

nagaraj@csd.uoc.gr
Speech Enhancement

- Enhancement involves two processes: denoising and improving intelligibility.
- Multiple types of noises: 

![Clean signal](image1)

![Noisy signal](image2)
Speech Enhancement

- Enhancement involves two processes: denoising and improving intelligibility.
- Multiple types of noises: , ,

![Clean Signal](image1)

![Noisy Signal](image2)
## INTRODUCTION

## SPEECH ENHANCEMENT

## VOICE CONVERSION

## CONCLUSION
**Speech Enhancement Generative Adversarial Network (SEGAN)**

- SEGAN is an end-to-end model for speech enhancement using GAN.

- It works with the raw audio. Therefore, no hand-crafted features are extracted and no explicit assumptions about the raw data while modeling.

- It learns from different speakers (28 speakers) and noise types (40 different noise conditions), and incorporates them together into the same shared parametrization.

- It consists of two neural networks: Generator and Discriminator.

- Generator network is structured similar to an auto-encoder.
In the encoder, input signal is projected and compressed through convolutional layers followed by rectified linear units (ReLUs).

- Compressed representation "c" is concatenated with latent vector "z".
- The encoding process is reversed in the decoder by means of transposed convolutions.
- G network also features skip connections to maintain phase and alignments.
In the encoder, input signal is projected and compressed through convolutional layers followed by rectified linear units (ReLUs).

Compressed representation "c" is concatenated with latent vector "z".

The encoding process is reversed in the decoder by means of transposed convolutions.

G network also features skip connections to maintain phase and alignments.
In the encoder, input signal is projected and compressed through convolutional layers followed by rectified linear units (ReLUs).

Compressed representation "c" is concatenated with latent vector "z".

The encoding process is reversed in the decoder by means of transposed convolutions.

G network also features skip connections to maintain phase and alignments.
In the encoder, input signal is projected and compressed through convolutional layers followed by rectified linear units (ReLUs).

Compressed representation "c" is concatenated with latent vector "z".

The encoding process is reversed in the decoder by means of transposed convolutions.

G network also features skip connections to maintain phase and alignments.

nagaraj@csd.uoc.gr
Loss function and training process:

- Training process uses least-square GAN approach with binary coding (1 for real, 0 for fake).

- Discriminator loss is given as follows:
  \[
  \min_D L(D) = \\
  \mathbb{E}_{(x_c, x_n) \sim p_d(x_c, x_n)}[(D(x_c, x_n) - 1)^2] + \\
  \mathbb{E}_{z \sim p_z(z), x_n \sim p_d(x_n)}[(D(G(z, x_n), x_n))^2]
  \]

- Generator loss is given as follows:
  \[
  \min_G L(G) = \\
  \mathbb{E}_{z \sim p_z(z), x_n \sim p_d(x_n)}[(D(G(z, x_n), x_n) - 1)^2]
  \]

- A secondary structure similarity L1 norm is added to Loss function of G with controlling parameter \( \lambda \)
  \[
  \min_G L(G) = \mathbb{E}_{z \sim p_z(z), x_n \sim p_d(x_n)}[(D(G(z, x_n), x_n) - 1)^2] + \\
  \lambda \| G(z, x_n) - x_c \|_1
  \]
Results for Highly Noise Scenarios

- Database: Voice Bank corpus (28 speakers) of CSTR Lab.
- Different types of noises: speech shaped noise, babble, kitchen, meeting room, cafeteria, restaurant, subway station, car, metro, and a street noise.
- The signal-to-noise (SNR) values added to training were: 15 dB, 10 dB, 5 dB and 0 dB.
Wasserstein GAN

- The loss function for discriminator with Wasserstein distance with gradient penalty is defined as follows:

$$\min_D L_W(D) = \mathbb{E}_{(x_c, x_n) \sim p_d(x_c, x_n)}[(D(x_c, x_n))]$$

$$- \mathbb{E}_{z \sim p_z(z), x_n \sim p_d(x_n)}[(D(G(z, x_n), x_n))]$$

$$+ \lambda_1 \mathbb{E}_{z \sim p_z(z), \tilde{x} \sim p_d(\tilde{x})}[\|\nabla \tilde{x} D(\tilde{x})\|_2^2 - 1]^2] \tag{1}$$

- Similarly, the loss function for $G$ is given by:

$$\min_G L_W(G) = -\mathbb{E}_{z \sim p_z(z), x_n \sim p_d(x_n)}[D(G(z, x_n), x_n)]$$

$$+ \lambda_2 \mathbb{E}_{z \sim p_z(z), (x_c, x_n) \sim p_d(x_c, x_n)}[\|G(z, x_n) - x_c\|_1] \tag{2}$$

- Parameter $\lambda_1$ and $\lambda_2$ is a hyper-parameter which controls the similarity term in the loss function of $D$ and $G$, respectively.
**Wasserstein GAN**

- The loss function for discriminator with Wasserstein distance with gradient penalty is defined as follows:

\[
\min_D L_W(D) = \mathbb{E}_{(x_c, x_n) \sim p_d(x_c, x_n)}[(D(x_c, x_n))] \\
- \mathbb{E}_{z \sim p_z(z), x_n \sim p_d(x_n)}[(D(G(z, x_n), x_n))] \\
+ \lambda_1 \mathbb{E}_{z \sim p_z(z), \tilde{x} \sim p_d(\tilde{x})}[(||\nabla_{\tilde{x}} D(\tilde{x})||_2 - 1)^2]
\]  

(1)

- Similarly, the loss function for \( G \) is given by:

\[
\min_G L_W(G) = -\mathbb{E}_{z \sim p_z(z), x_n \sim p_d(x_n)}[D(G(z, x_n), x_n)] \\
+ \lambda_2 \mathbb{E}_{z \sim p_z(z), (x_c, x_n) \sim p_d(x_c, x_n)}[|| G(z, x_n) - x_c ||_1]
\]

(2)

- Parameter \( \lambda_1 \) and \( \lambda_2 \) is a hyper-parameter which controls the similarity term in the loss function of \( D \) and \( G \), respectively.
Wasserstein GAN

- The loss function for discriminator with Wasserstein distance with gradient penalty is defined as follows:

\[
\min_D L_W(D) = \mathbb{E}_{(x_c,x_n) \sim p_d(x_c,x_n)}[(D(x_c, x_n))] - \mathbb{E}_{z \sim p_z(z), x_n \sim p_d(x_n)}[(D(G(z, x_n), x_n))] + \lambda_1 \mathbb{E}_{z \sim p_z(z), \tilde{x} \sim p_d(\tilde{x})}[\|\nabla_{\tilde{x}} D(\tilde{x})\|_2^2 - 1]^2
\]  

(1)

- Similarly, the loss function for \( G \) is given by:

\[
\min_G L_W(G) = -\mathbb{E}_{z \sim p_z(z), x_n \sim p_d(x_n)}[D(G(z, x_n), x_n)] + \lambda_2 \mathbb{E}_{z \sim p_z(z), (x_c, x_n) \sim p_d(x_c,x_n)}[\| G(z, x_n) - x_c \|_1]
\]  

(2)

- Parameter \( \lambda_1 \) and \( \lambda_2 \) is a hyper-parameter which controls the similarity term in the loss function of \( D \) and \( G \), respectively.
WGAN with Gated activation function

- Conventionally, either regular rectified linear units (ReLUs) or parametric ReLUs are used as activation function.

- Using gated linear units (GLUs) activation function have been reported to capture both long- and short-term dependencies which are present in speech.

- The output of the $l^{th}$ hidden layer of a gated CNN is described as:

$$H_l = (H_{l-1} \ast W_l) \otimes \sigma(H_{l-1} \ast V_l)$$  \hspace{1cm} (3)

- where $\sigma$ is the sigmoid function and $\otimes$ represents element-wise product. $W_l$ and $V_l$ are parameters to be learned during training.
**WGAN with Gated activation function**

- Conventionally, either regular rectified linear units (ReLUs) or parametric ReLUs are used as activation function.

- Using gated linear units (GLUs) activation function have been reported to capture both long- and short-term dependencies which are present in speech.

- The output of the $l^{th}$ hidden layer of a gated CNN is described as:

$$H_l = (H_{l-1} * W_l) \otimes \sigma(H_{l-1} * V_l)$$

where $\sigma$ is the sigmoid function and $\otimes$ represents element-wise product. $W_l$ and $V_l$ are parameters to be learned during training.
**WGAN with Gated activation function**

- Conventionally, either regular rectified linear units (ReLUs) or parametric ReLUs are used as activation function.

- Using gated linear units (GLUs) activation function have been reported to capture both long- and short-term dependencies which are present in speech.

- The output of the $l^{th}$ hidden layer of a gated CNN is described as:

$$H_l = (H_{l-1} * W_l) \otimes \sigma(H_{l-1} * V_l)$$  \hspace{1cm} (3)

where $\sigma$ is the sigmoid function and $\otimes$ represents element-wise product. $W_l$ and $V_l$ are parameters to be learned during training.
WGAN with Gated Activation Function

- Conventionally, either regular rectified linear units (ReLUs) or parametric ReLUs are used as activation function.

- Using gated linear units (GLUs) activation function have been reported to capture both long- and short-term dependencies which are present in speech.

- The output of the $l^{th}$ hidden layer of a gated CNN is described as:

$$H_l = (H_{l-1} \ast W_l) \otimes \sigma(H_{l-1} \ast V_l)$$  \hspace{1cm} (3)

- where $\sigma$ is the sigmoid function and $\otimes$ represents element-wise product. $W_l$ and $V_l$ are parameters to be learned during training.
# Results of WGAN for Low Noisy Scenarios

<table>
<thead>
<tr>
<th>Noisy files</th>
<th>SEGAN</th>
<th>WGAN</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Audio Sample" /></td>
<td><img src="image2" alt="Audio Sample" /></td>
<td><img src="image3" alt="Audio Sample" /></td>
</tr>
<tr>
<td><img src="image4" alt="Audio Sample" /></td>
<td><img src="image5" alt="Audio Sample" /></td>
<td><img src="image6" alt="Audio Sample" /></td>
</tr>
<tr>
<td><img src="image7" alt="Audio Sample" /></td>
<td><img src="image8" alt="Audio Sample" /></td>
<td><img src="image9" alt="Audio Sample" /></td>
</tr>
<tr>
<td><img src="image10" alt="Audio Sample" /></td>
<td><img src="image11" alt="Audio Sample" /></td>
<td><img src="image12" alt="Audio Sample" /></td>
</tr>
</tbody>
</table>

nagaraj@csd.uoc.gr
OUTLINE

1 INTRODUCTION

2 SPEECH ENHANCEMENT

3 VOICE CONVERSION

4 CONCLUSION
Voice conversion

Source speaker

Target speaker

Objective function

Output

G

nagaraj@csd.uoc.gr

GANs
Voice conversion using WGAN

- VAW-GAN [Hsu et al., Interspeech 2017]
## Results of VC using WGAN

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>VAE</th>
<th>WGAN</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="source.png" alt="Image" /></td>
<td><img src="target.png" alt="Image" /></td>
<td><img src="VAE.png" alt="Image" /></td>
<td><img src="WGAN.png" alt="Image" /></td>
</tr>
</tbody>
</table>
Voice conversion using CycleGAN

GANs
Three type of Losses: Adversarial loss, Cycle-consistency loss, Identity loss.

\[ L_{full} = L_{adv}(G_{S\rightarrow T}, D_T) + L_{adv}(G_{T\rightarrow S}, D_S) + \lambda_{cyc} L_{cyc}(G_{S\rightarrow T}, G_{T\rightarrow S}) + \lambda_{id} L_{id}(G_{S\rightarrow T}, G_{T\rightarrow S}) \]
Outline

1. Introduction

2. Speech Enhancement

3. Voice conversion

4. Conclusion
Speech enhancement with GANs gives impressive results for speech enhancement in both high and low SNR conditions.

GANs helped in reducing the robotic speech quality present in Voice conversion application.

GANs have the potential to further improve the performance results.

GANs can be used in other Speech application or Speech transformation or signal transformation.
Speech enhancement with GANs gives impressive results for speech enhancement in both high and low SNR conditions.

GANs helped in reducing the robotic speech quality present in voice conversion application.

GANs have the potential to further improve the performance results.

GANs can be used in other speech application or speech transformation or signal transformation.
Speech enhancement with GANs gives impressive results for Speech enhancement in both high and low SNR conditions.

GANs helped in reducing the robotic speech quality present in Voice conversion application.

GANs have the potential to further improve the performance results.

GANs can be used in other Speech application or Speech transformation or signal transformation.
Conclusion

- Speech enhancement with GANs gives impressive results for Speech enhancement in both high and low SNR conditions.

- GANs helped in reducing the robotic speech quality present in Voice conversion application.

- GANs have the potential to further improve the performance results.

- GANs can be used in other Speech application or Speech transformation or signal transformation.


Pantazis et al. ,”https://www.csd.uoc.gr/~spcc/".


Thank You