Generative Adversarial Network and its Applications to Signal Processing and Natural Language Processing

Hung-yi Lee and Yu Tsao
Outline

Part I: General Introduction of Generative Adversarial Network (GAN)

Part II: Applications on Signal Processing

Part III: Applications on Natural Language Processing
Generative Adversarial Network
and its Applications to Signal Processing
and Natural Language Processing

Part I: General Introduction
All Kinds of GAN ... https://github.com/hindupuravinash/the-gan-zoo

GAN
ACGAN
BGAN
CGAN
DCGAN
EBGAN
fGAN
GoGAN
...

Cumulative number of named GAN papers by month

2 We use the Greek $\alpha$ prefix for $\alpha$-GAN, as AEGAN and most other Latin prefixes seem to have been taken
https://deephunt.in/the-gan-zoo-79597dc8c347.

Mihaela Rosca, Balaji Lakshminarayanan, David Warde-Farley, Shakir Mohamed, "Variational Approaches for Auto-Encoding Generative Adversarial Networks", arXiv, 2017
Number of papers whose titles include the keyword

ICASSP

Keyword search on session index page, so session name is included.

- Generative
- Adversarial
- Reinforcement
Outline of Part 1

- Generation by GAN
- Conditional Generation
- Unsupervised Conditional Generation
- Relation to Reinforcement Learning
Outline of Part 1

Generation by GAN
- Image Generation as Example
- Theory behind GAN
- Issues and Possible Solutions

Conditional Generation

Unsupervised Conditional Generation

Relation to Reinforcement Learning
Anime Face Generation

Examples

Generator

Draw
Basic Idea of GAN

- **Generator**: It is a neural network (NN), or a function.
- **Image**: High dimensional vector
- **Input Vector**: Each dimension represents some characteristics.

Examples:
- Longer hair: 
  - Input: 
  - Output: 
- Blue hair: 
  - Input: 
  - Output: 
- Open mouth: 
  - Input: 
  - Output: 

Powered by: http://mattya.github.io/chainer-DCGAN/
Basic Idea of GAN

It is a neural network (NN), or a function.

Larger value means real, smaller value means fake.
Algorithm

- Initialize generator and discriminator
- In each training iteration:

**Step 1:** Fix generator G, and update discriminator D

Discriminator learns to assign high scores to real objects and low scores to generated objects.
Algorithm

- Initialize generator and discriminator
- In each training iteration:

**Step 2**: Fix discriminator D, and update generator G

Generator learns to "fool" the discriminator
Algorithm

- Initialize generator and discriminator
- In each training iteration:

Sample some real objects:
Generate some fake objects:

Learning G

Learning D

Update D

update

fix
Anime Face Generation

Source of training data: https://zhuanlan.zhihu.com/p/24767059
Anime Face Generation

1000 updates
Anime Face Generation

2000 updates
Anime Face Generation

5000 updates
Anime Face Generation

10,000 updates
Anime Face Generation

20,000 updates
Anime Face Generation

50,000 updates
The faces generated by machine.

The images are generated by Yen-Hao Chen, Po-Chun Chien, Jun-Chen Xie, Tsung-Han Wu.
(Variational) Auto-encoder

As close as possible

Randomly generate a vector as code

= Generator

= Generator
Auto-encoder v.s. GAN

**Auto-encoder**

- Code
- NN Decoder
- As close as possible
- Just copy an image
- = Generator
- Fuzzy ...

**GAN**

- Code
- Generator
- Discriminator
- If discriminator does not simply memorize the images,
  Generator learns the patterns of faces.
FID: Smaller is better
Outline of Part 1

Generation
- Image Generation as Example
- Theory behind GAN
- Issues and Possible Solutions

Conditional Generation

Unsupervised Conditional Generation

Relation to Reinforcement Learning
Generator

• A generator $G$ is a network. The network defines a probability distribution $P_G$

$G^* = \arg \min_G \text{Div}(P_G, P_{data})$

Divergence between distributions $P_G$ and $P_{data}$

How to compute the divergence?
Discriminator

\[ G^* = \arg \min_G \text{Div}(P_G, P_{data}) \]

Although we do not know the distributions of \( P_G \) and \( P_{data} \), we can sample from them.
Discriminator \( G^* = \arg \min_G \text{Div}(P_G, P_{\text{data}}) \)

\( \star \) : data sampled from \( P_{\text{data}} \)
\( \star \star \) : data sampled from \( P_G \)

Using the example objective function is exactly the same as training a binary classifier.

**Example** Objective Function for \( D \)

\[
V(G, D) = E_{x \sim P_{\text{data}}} \left[ \log D(x) \right] + E_{x \sim P_G} \left[ \log (1 - D(x)) \right]
\]

\( G \) is fixed

**Training:** \( D^* = \arg \max_D V(D, G) \)

The maximum objective value is related to JS divergence.

[Goodfellow, et al., NIPS, 2014]
Discriminator \[ G^* = \arg \min_G Div(P_G, P_{data}) \]

Discriminator:
- Blue stars: data sampled from \( P_{data} \)
- Orange stars: data sampled from \( P_G \)

Training:
\[ D^* = \arg \max_D V(D, G) \]

Small divergence:
- Hard to discriminate
  (cannot make objective large)

Large divergence:
- Easy to discriminate
\[ G^* = \arg \min_G \max_D V(G, D) \]

\[ D^* = \arg \max_D V(D, G) \]

The maximum objective value is related to the divergence.

- Initialize generator and discriminator
- In each training iteration:
  - **Step 1**: Fix generator G, and update discriminator D
  - **Step 2**: Fix discriminator D, and update generator G

[Goodfellow, et al., NIPS, 2014]
Can we use other divergence?

<table>
<thead>
<tr>
<th>Name</th>
<th>$D_f(P|Q)$</th>
<th>Generator $f(u)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total variation</td>
<td>$\frac{1}{2} \int</td>
<td>p(x) - q(x)</td>
</tr>
<tr>
<td>Kullback-Leibler</td>
<td>$\int p(x) \log \frac{p(x)}{q(x)} , dx$</td>
<td>$u \log u$</td>
</tr>
<tr>
<td>Reverse Kullback-Leibler</td>
<td>$\int q(x) \log \frac{q(x)}{p(x)} , dx$</td>
<td>$- \log u$</td>
</tr>
<tr>
<td>Pearson $\chi^2$</td>
<td>$\int \frac{(p(x) - q(x))^2}{p(x)} , dx$</td>
<td>$(u - 1)^2$</td>
</tr>
<tr>
<td>Neyman $\chi^2$</td>
<td>$\int \frac{(p(x) - q(x))^2}{q(x)} , dx$</td>
<td>$\frac{(1-u)^2}{u}$</td>
</tr>
<tr>
<td>Squared Hellinger</td>
<td>$\int \left( \sqrt{p(x)} - \sqrt{q(x)} \right)^2 , dx$</td>
<td>$(\sqrt{u} - 1)^2$</td>
</tr>
<tr>
<td>Jeffrey</td>
<td>$\int (p(x) - q(x)) \log \left( \frac{p(x)}{q(x)} \right) , dx$</td>
<td>$(u - 1) \log u$</td>
</tr>
<tr>
<td>Jensen-Shannon</td>
<td>$\frac{1}{2} \int p(x) \log \frac{2p(x)}{p(x)+q(x)} + q(x) \log \frac{2q(x)}{p(x)+q(x)} , dx$</td>
<td>$-(u + 1) \log \frac{1+u}{2} + u \log u$</td>
</tr>
<tr>
<td>Jensen-Shannon-weighted</td>
<td>$\int p(x) \pi \log \frac{\pi p(x)+(1-\pi)q(x)}{p(x)+q(x)} + (1-\pi)q(x) \log \frac{q(x)}{p(x)+(1-\pi)q(x)} , dx$</td>
<td>$\pi u \log u - (1 - \pi + \pi u) \log(1 - \pi + \pi u)$</td>
</tr>
<tr>
<td>GAN</td>
<td>$\int p(x) \log \frac{2p(x)}{p(x)+q(x)} + q(x) \log \frac{2q(x)}{p(x)+q(x)} , dx - \log(4)$</td>
<td>$u \log u - (u + 1) \log(u + 1)$</td>
</tr>
</tbody>
</table>

Using the divergence you like 😊

[Sebastian Nowozin, et al., NIPS, 2016]
Outline of Part 1

Generation
- Image Generation as Example
- Theory behind GAN
- Issues and Possible Solutions

Conditional Generation

Unsupervised Conditional Generation

Relation to Reinforcement Learning

https://github.com/soumith/ganhacks
JS divergence is not suitable

- In most cases, $P_G$ and $P_{data}$ are not overlapped.
- 1. The nature of data
  - Both $P_{data}$ and $P_G$ are low-dim manifold in high-dim space.
  - The overlap can be ignored.
- 2. Sampling
  - Even though $P_{data}$ and $P_G$ have overlap.
  - If you do not have enough sampling ......
What is the problem of JS divergence?

JS divergence is log2 if two distributions do not overlap.

Intuition: If two distributions do not overlap, binary classifier achieves 100% accuracy.

Same objective value is obtained. Same divergence
Wasserstein distance

• Considering one distribution $P$ as a pile of earth, and another distribution $Q$ as the target.
• The average distance the earth mover has to move the earth.

$$W(P, Q) = d$$
Wasserstein distance

There are many possible “moving plans”.
Using the “moving plan” with the smallest average distance to define the Wasserstein distance.

Source of image: https://vincentherrmann.github.io/blog/wasserstein/
What is the problem of JS divergence?

\[ JS(P_{G0}, P_{data}) = \log_2 \]
\[ W(P_{G0}, P_{data}) = d_0 \]

\[ JS(P_{G1}, P_{data}) = \log_2 \]
\[ W(P_{G1}, P_{data}) = d_1 \]

......

\[ JS(P_{G100}, P_{data}) = 0 \]
\[ W(P_{G100}, P_{data}) = 0 \]

Better!
WGAN

Evaluate wasserstein distance between \( P_{data} \) and \( P_G \)

\[
V(G, D) = \max_{D \in 1-Lipschitz} \{ E_{x \sim P_{data}} [D(x)] - E_{x \sim P_G} [D(x)] \}
\]

D has to be smooth enough. How to fulfill this constraint?

Without the constraint, the training of D will not converge.

Keeping the D smooth forces D(x) become \( \infty \) and \(-\infty\)
\[ V(G, D) = \max_{D \in 1-Lipschitz} \left\{ E_{x \sim P_{data}}[D(x)] - E_{x \sim P_G}[D(x)] \right\} \]

- **Original WGAN \rightarrow Weight Clipping** [Martin Arjovsky, et al., arXiv, 2017]
  
  Force the parameters \( w \) between \( c \) and \( -c \)
  
  After parameter update, if \( w > c \), \( w = c \); if \( w < -c \), \( w = -c \)

- **Improved WGAN \rightarrow Gradient Penalty** [Ishaan Gulrajani, NIPS, 2017]

- **Spectral Normalization \rightarrow Keep gradient norm smaller than 1 everywhere** [Miyato, et al., ICLR, 2018]
Energy-based GAN (EBGAN)

- Using an autoencoder as discriminator D
  - Using the negative reconstruction error of auto-encoder to determine the goodness
  - **Benefit**: The auto-encoder can be pre-train by real images without generator.

[Junbo Zhao, et al., arXiv, 2016]
Mode Collapse

Training with too many iterations ....

★ : real data
★橙色: generated data
Generator switches mode during training

Generator at iteration $t$

Generator at iteration $t+1$

Generator at iteration $t+2$
Ensemble
Train a set of generators: \( \{G_1, G_2, \cdots, G_N\} \)
To generate an image
Random pick a generator \( G_i \)
Use \( G_i \) to generate the image
Objective Evaluation

\[ P(y|x) \]

\[ x: \text{image} \]
\[ y: \text{class (output of CNN)} \]

Concentrated distribution means higher visual quality

Uniform distribution means higher variety

\[ P(y) = \frac{1}{N} \sum_{n} P(y^n|x^n) \]

\[ x \rightarrow \text{Off-the-shelf Image Classifier} \rightarrow P(y|x) \]

e.g. Inception net, VGG, etc.

\[ x^1 \rightarrow \text{CNN} \rightarrow P(y^1|x^1) \]
\[ x^2 \rightarrow \text{CNN} \rightarrow P(y^2|x^2) \]
\[ x^3 \rightarrow \text{CNN} \rightarrow P(y^3|x^3) \]

\[ \vdots \]

[Tim Salimans, et al., NIPS, 2016]
### Objective Evaluation

**Inception Score**

\[
P(y) = \frac{1}{N} \sum_{n} P(y^n | x^n)
\]

\[
\text{Negative entropy of } P(y|\mathbf{x})
\]

\[
\text{Entropy of } P(y)
\]

\[
\sum_{x} \sum_{y} P(y|\mathbf{x}) \log P(y|\mathbf{x})
\]

\[
- \sum_{y} P(y) \log P(y)
\]
Outline of Part 1

- Generation
- Conditional Generation
- Unsupervised Conditional Generation
- Relation to Reinforcement Learning
**Original Generator**

\[ P_G(x) \rightarrow P_{data}(x) \]

\[ x = G(z) \]

**Conditional Generator**

\[ P_G(x|c) \rightarrow P_{data}(x|c) \]

\[ x = G(c, z) \]

[Mehdi Mirza, et al., arXiv, 2014]

e.g. Text-to-Image

“Girl with red hair and red eyes”

“Girl with yellow ribbon”
Text-to-Image

- Traditional supervised approach

$c^1$: a dog is running

Text: “train”

A blurry image!

Target of NN output

as close as possible

a dog is running

a bird is flying
Conditional GAN

\[ x = G(c,z) \]

Normal distribution \( z \)  
\[ \text{x: train} \]
\[ \text{G} \]
\[ \text{Image} \]

\( x \) is real image or not

\[ x \]
\[ \text{D} \] (original)
\[ \text{scalar} \]

Real images: 1  
Generated images: 0

Generator will learn to generate realistic images ....  
But completely ignore the input conditions.

[Scott Reed, et al, ICML, 2016]
Conditional GAN

$c$: train  \[
x = G(c, z)
\]

Normal distribution $z$

$D$: (better)  \[
x \text{ is realistic or not } +
\text{c and x are matched or not}
\]

True text-image pairs:  \[
(train, ) \quad (0)
\]
\[
(cat, ) \quad (0)
\]

[Scott Reed, et al, ICML, 2016]
Conditional GAN - Discriminator

object x → Network → Network

condition c → Network

(score)
x is realistic or not + c and x are matched or not

(almost every paper)

object x → Network

x is realistic or not

c and x are matched or not

[Augustus Odena et al., ICML, 2017]
[Takeru Miyato, et al., ICLR, 2018]
[Han Zhang, et al., arXiv, 2017]
Conditional GAN

paired data

blue eyes
red hair
short hair

Collecting anime faces and the description of its characteristics

red hair, green eyes

blue hair, red eyes

The images are generated by Yen-Hao Chen, Po-Chun Chien, Jun-Chen Xie, Tsung-Han Wu.
Conditional GAN - Image-to-image

$\mathbf{c} \xrightarrow{} G \xrightarrow{} x = G(\mathbf{c}, \mathbf{z})$

[Phillip Isola, et al., CVPR, 2017]

Image translation, or pix2pix
Conditional GAN - Image-to-image

• Traditional supervised approach

Testing:

<table>
<thead>
<tr>
<th>Input</th>
<th>L1</th>
<th>Image</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Input Image" /></td>
<td><img src="image2.png" alt="L1 Image" /></td>
<td><img src="image3.png" alt="Generated Image" /></td>
</tr>
</tbody>
</table>

It is blurry.

[Phillip Isola, et al., CVPR, 2017]
Conditional GAN - Image-to-image

Testing:

input  L1  GAN  GAN + L1
Conditional GAN - Video Generation

[Michael Mathieu, et al., arXiv, 2015]
Domain Adversarial Training

- Training and testing data are in different domains

Take digit classification as example
Domain Adversarial Training

- Feature extractor (Generator)
- Always output zero vectors
- Domain Classifier Fails
- Which domain?
- Discriminator (Domain classifier)

Input $x$

- Blue points
- Red points
Domain Adversarial Training

Not only cheat the domain classifier, but satisfying label predictor at the same time.

Successfully applied on image classification
[ Ganin et al, ICML, 2015 ] [ Ajakan et al. JMLR, 2016 ]

More speech-related applications in Part II.
Outline of Part 1

- Generation
- Conditional Generation
- Unsupervised Conditional Generation
- Relation to Reinforcement Learning
Unsupervised Conditional Generation

Transform an object from one domain to another **without paired data** (e.g. style transfer)

It is good.
It’s a good day.
I love you.

It is bad.
It’s a bad day.
I don’t love you.
Unsupervised Conditional Generation

• Approach 1: Direct Transformation

\[ \mathcal{G}_{X \rightarrow Y} \]

Domain X \rightarrow Domain Y

For texture or color change

• Approach 2: Projection to Common Space

Encoder of domain X \( \mathcal{E}_{N_X} \)

Decoder of domain Y \( \mathcal{D}_{E_Y} \)

Larger change, only keep the semantics
Direct Transformation

Domain X \( \xrightarrow{G_{X \rightarrow Y}} \) Domain Y

Become similar to domain Y

Input image belongs to domain Y or not

\( G_{X \rightarrow Y} \)

\( D_Y \) scalar
Direct Transformation

Domain X

\[ G_{X\rightarrow Y} \]

Become similar to domain Y

Not what we want!

Input image belongs to domain Y or not

Domain Y

\[ D_Y \]

scalar

ignore input
Direct Transformation

\[ G_{X \to Y} \]

Domain X → Domain Y

Become similar to domain Y

Not what we want!

Input image belongs to domain Y or not

The issue can be avoided by network design. Simpler generator makes the input and output more closely related.

[Tomer Galanti, et al. ICLR, 2018]
Direct Transformation

$$G_{X \rightarrow Y}$$

Encoder Network

pre-trained

Encoder Network

as close as possible

$$D_Y$$

Input image belongs to domain Y or not

scalar

Baseline of DTN [Yaniv Taigman, et al., ICLR, 2017]
Direct Transformation

\[ G_{X \rightarrow Y} \rightarrow D_Y \rightarrow G_{Y \rightarrow X} \]

as close as possible

Cycle consistency

Lack of information for reconstruction

Input image belongs to domain Y or not

Domains X and Y

Direct Transformation

\[ G_{X \to Y} \rightarrow \text{as close as possible} \rightarrow \text{as close as possible} \]

scalar: belongs to domain X or not

\[ D_X \]

scalar: belongs to domain Y or not

\[ D_Y \]

as close as possible
For multiple domains, considering starGAN
[Junjey Choi, arXiv, 2017]

Disco GAN

Dual GAN

Cycle GAN

Issue of Cycle Consistency

• CycleGAN: a Master of Steganography

[Casey Chu, et al., NIPS workshop, 2017]

The information is hidden.
Unsupervised Conditional Generation

• Approach 1: Direct Transformation

\[ G_{X \rightarrow Y} \]

Domain X \rightarrow \text{For texture or color change} \rightarrow \text{Domain Y}

• Approach 2: Projection to Common Space

\[ EN_X \]

\[ DE_Y \]

Domain X \rightarrow \text{Encoder of domain X} \rightarrow \text{Face Attribute} \rightarrow \text{Decoder of domain Y} \rightarrow \text{Domain Y}

Larger change, only keep the semantics
Projection to Common Space

Target

$EN_X$ → $DE_X$ → image

$EN_Y$ → $DE_Y$ → image

Domain X

Domain Y
Projection to Common Space

Training

Minimizing reconstruction error
Because we train two auto-encoders separately ... 

The images with the same attribute may not project to the same position in the latent space.
Projection to Common Space

Training

Sharing the parameters of encoders and decoders

Couple GAN [Ming-Yu Liu, et al., NIPS, 2016]
UNIT [Ming-Yu Liu, et al., NIPS, 2017]
Projection to Common Space

Training

Minimizing reconstruction error

Discriminator of X domain

Discriminator of Y domain

$EN_X$ and $EN_Y$ fool the domain discriminator

The domain discriminator forces the output of $EN_X$ and $EN_Y$ have the same distribution. [Guillaume Lample, et al., NIPS, 2017]
Projection to Common Space

Training

Minimizing reconstruction error

Discriminator of X domain

$EN_x$ $DE_x$ $DX$

Discriminator of Y domain

$EN_y$ $DE_y$ $DY$

Cycle Consistency:

Used in ComboGAN [Asha Anoosheh, et al., arXiv, 017]
Projection to Common Space

Training

To the same latent space

Discriminator of X domain

Discriminator of Y domain

Semantic Consistency:

Outline of Part 1

- Generation
- Conditional Generation
- Unsupervised Conditional Generation
- Relation to Reinforcement Learning
Basic Components

<table>
<thead>
<tr>
<th>Actor</th>
<th>Env</th>
<th>Reward Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video Game</td>
<td>Get 20 scores when killing a monster</td>
<td></td>
</tr>
<tr>
<td>Go</td>
<td>The rule of GO</td>
<td>You cannot control</td>
</tr>
</tbody>
</table>

You cannot control
Neural network as Actor

- Input of neural network: the observation of machine represented as a vector or a matrix
- Output neural network: each action corresponds to a neuron in output layer

Take the action based on the probability.

Score of an action:
- left: 0.7
- right: 0.2
- fire: 0.1
Example: Playing Video Game

Start with observation $s_1$

Observation $s_2$

Observation $s_3$

Action $a_1$: “right”

Obtain reward $r_1 = 0$

Action $a_2$: “fire” (kill an alien)

Obtain reward $r_2 = 5$
Example: Playing Video Game

Start with observation $s_1$

Observation $s_2$

Observation $s_3$

After many turns

Game Over (spaceship destroyed)

Obtain reward $r_T$

Action $a_T$

This is an episode.

Total reward:

$$ R = \sum_{t=1}^{T} r_t $$

We want the total reward be maximized.
Actor, Environment, Reward

Trajectory

\[ \tau = \{ s_1, a_1, s_2, a_2, \ldots, s_T, a_T \} \]
Reinforcement Learning v.s. GAN

Actor → Generator
Reward Function → Discriminator

Reward \( r_1 \)
\[ R(\tau) = \sum_{t=1}^{T} r_t \]
Imitation Learning

We have demonstration of the expert.

\[
\text{Env} \rightarrow s_1 \rightarrow a_1 \rightarrow \text{Env} \rightarrow s_2 \rightarrow a_2 \rightarrow \text{Env} \rightarrow s_3 \rightarrow a_2 \\
\text{Env} \rightarrow s_1 \rightarrow a_1 \rightarrow \text{Env} \rightarrow s_2 \rightarrow a_2 \rightarrow \text{Env} \rightarrow s_3 \rightarrow a_2 \\
\cdots
\]

reward function is not available

Self driving: record human drivers

Robot: grab the arm of robot

We have demonstration of the expert.

Each \( \hat{t} \) is a trajectory of the expert.

\[ \{\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_N \} \]
Inverse Reinforcement Learning

- Using the reward function to find the optimal actor.
- Modeling reward can be easier. Simple reward function can lead to complex policy.

\[ \{\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_N\} \]

Environment

Reward Function

Reinforcement Learning

Demonstration of the expert

Expert
Framework of IRL

The expert is always the best.

\[ \sum_{n=1}^{N} R(\hat{\pi}_n) > \sum_{n=1}^{N} R(\tau) \]
GAN

High score for real, low score for generated

D

Find a G whose output obtains large score from D

IRL

Expert

\{\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_N\}

\{\tau_1, \tau_2, \ldots, \tau_N\}

Actor

Larger reward for \(\hat{t}_n\), Lower reward for \(\tau\)

Reward Function

Find a Actor obtains large reward
Concluding Remarks

- Generation
- Conditional Generation
- Unsupervised Conditional Generation
- Relation to Reinforcement Learning
Reference

• **Generation**
  
  • Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, Generative Adversarial Networks, NIPS, 2014
  
  
  • Martin Arjovsky, Soumith Chintala, Léon Bottou, Wasserstein GAN, arXiv, 2017
  
  • Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, Aaron Courville, Improved Training of Wasserstein GANs, NIPS, 2017
  
  • Junbo Zhao, Michael Mathieu, Yann LeCun, Energy-based Generative Adversarial Network, arXiv, 2016
  
  • Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, Olivier Bousquet, “Are GANs Created Equal? A Large-Scale Study”, arXiv, 2017
  
  • Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen Improved Techniques for Training GANs, NIPS, 2016
  
  • Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Sepp Hochreiter, GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium, NIPS, 2017
Reference

• **Generation**
  • Xiang Wei, Boqing Gong, Zixia Liu, Wei Lu, Liqiang Wang, Improving the Improved Training of Wasserstein GANs: A Consistency Term and Its Dual Effect, ICLR, 2018
  • Takeru Miyato, Toshiki Kataoka, Masanori Koyama, Yuichi Yoshida, Spectral Normalization for Generative Adversarial Networks, ICLR, 2018
Reference

• **Generational Generation**
  • Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, Honglak Lee, Generative Adversarial Text to Image Synthesis, ICML, 2016
  • Michael Mathieu, Camille Couprie, Yann LeCun, Deep multi-scale video prediction beyond mean square error, arXiv, 2015
  • Mehdi Mirza, Simon Osindero, Conditional Generative Adversarial Nets, arXiv, 2014
  • Takeru Miyato, Masanori Koyama, cGANs with Projection Discriminator, ICLR, 2018
  • Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, Dimitris Metaxas, StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks, arXiv, 2017
  • Augustus Odena, Christopher Olah, Jonathon Shlens, Conditional Image Synthesis With Auxiliary Classifier GANs, ICML, 2017
Reference

• **Generational Generation**
  
  • Yaroslav Ganin, Victor Lempitsky, Unsupervised Domain Adaptation by Backpropagation, ICML, 2015
  
  • Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, Domain-Adversarial Training of Neural Networks, JMLR, 2016
Reference

• **Unsupervised Conditional Generation**
  
  
  • Zili Yi, Hao Zhang, Ping Tan, Minglun Gong, DualGAN: Unsupervised Dual Learning for Image-to-Image Translation, ICCV, 2017
  
  • Tomer Galanti, Lior Wolf, Sagie Benaim, The Role of Minimal Complexity Functions in Unsupervised Learning of Semantic Mappings, ICLR, 2018
  
  • Yaniv Taigman, Adam Polyak, Lior Wolf, Unsupervised Cross-Domain Image Generation, ICLR, 2017
  
  • Asha Anoosheh, Eirikur Agustsson, Radu Timofte, Luc Van Gool, ComboGAN: Unrestrained Scalability for Image Domain Translation, arXiv, 2017
  
  • Amélie Royer, Konstantinos Bousmalis, Stephan Gouws, Fred Bertsch, Inbar Mosseri, Forrester Cole, Kevin Murphy, XGAN: Unsupervised Image-to-Image Translation for Many-to-Many Mappings, arXiv, 2017
Reference

• **Unsupervised Conditional Generation**
  
  • Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, Marc'Aurelio Ranzato, Fader Networks: Manipulating Images by Sliding Attributes, NIPS, 2017
  
  • Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, Jiwon Kim, Learning to Discover Cross-Domain Relations with Generative Adversarial Networks, ICML, 2017
  
  • Ming-Yu Liu, Oncel Tuzel, “Coupled Generative Adversarial Networks”, NIPS, 2016
  
  • Ming-Yu Liu, Thomas Breuel, Jan Kautz, Unsupervised Image-to-Image Translation Networks, NIPS, 2017
  
  • Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, Jaegul Choo, StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation, arXiv, 2017
Generative Adversarial Network
and its Applications to Signal Processing
and Natural Language Processing

Part II: Speech Signal Processing
Outline of Part II

Speech Signal Generation

- Speech enhancement
- Postfilter, speech synthesis, voice conversion

Speech Signal Recognition

- Speech recognition
- Speaker recognition
- Speech emotion recognition
- Lip reading

Conclusion
Speech Signal Generation (Regression Task)

Paired

Objective function

Conditional GAN

\[ x = G(c,z) \]

Prior distribution \( z \)

True text-image pairs:

- (train, ) 1
- (cat, ) 0
- (train, Image ) 0

Cycle-GAN

\[ G_{X \rightarrow Y} \]

\[ D_X \]

\[ D_Y \]

\[ G_{Y \rightarrow X} \]

\[ G_{X \rightarrow Y} \]

scalar: belongs to domain X or not

scalar: belongs to domain Y or not

as close as possible
Speech, Speaker, Emotion Recognition and Lip-reading (Classification Task)

Domain Adversarial Training

**feature extractor (Generator)**

Not only cheat the domain classifier, but satisfying label predictor at the same time

Successfully applied on image classification

[Ganin et al, ICML, 2015][Ajakan et al. JMLR, 2016]

Acoustic Mismatch

\[ \tilde{x} \]

Channel distortion

\[ \bar{x} \]

Accented speech

\[ \bar{\bar{x}} \]

Noisy data

\[ x \]

Clean data

\[ y \]

Output label

\[ \tilde{z} = g(\bar{x}) \]

\[ g(\cdot) \]

\[ h(\cdot) \]
Outline of Part II

Speech Signal Generation
- Speech enhancement
- Postfilter, speech synthesis, voice conversion

Speech Signal Recognition
- Speech recognition
- Speaker recognition
- Speech emotion recognition
- Lip reading

Conclusion
Speech Enhancement

- Typical objective function

  Model structures of $G$: DNN [Wang et al. NIPS 2012; Xu et al., SPL 2014], DDAE [Lu et al., Interspeech 2013], RNN (LSTM) [Chen et al., Interspeech 2015; Weninger et al., LVA/ICA 2015], CNN [Fu et al., Interspeech 2016].

- Typical objective function

  - Mean square error (MSE) [Xu et al., TASLP 2015], L1 [Pascual et al., Interspeech 2017], likelihood [Chai et al., MLSP 2017], STOI [Fu et al., TASLP 2018].
  - GAN is used as a new objective function to estimate the parameters in $G$. 
Speech Enhancement

- Speech enhancement GAN (SEGAN) [Pascual et al., Interspeech 2017]
Speech Enhancement (SEGAN)

- Experimental results

Table 1: Objective evaluation results.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Noisy</th>
<th>Wiener</th>
<th>SEGAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>PESQ</td>
<td>1.97</td>
<td>2.22</td>
<td>2.16</td>
</tr>
<tr>
<td>CSIG</td>
<td>3.35</td>
<td>3.23</td>
<td>3.48</td>
</tr>
<tr>
<td>CBAK</td>
<td>2.44</td>
<td>2.68</td>
<td>2.94</td>
</tr>
<tr>
<td>COVL</td>
<td>2.63</td>
<td>2.67</td>
<td>2.80</td>
</tr>
<tr>
<td>SSNR</td>
<td>1.68</td>
<td>5.07</td>
<td>7.73</td>
</tr>
</tbody>
</table>

Table 2: Subjective evaluation results.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Noisy</th>
<th>Wiener</th>
<th>SEGAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOS</td>
<td>2.09</td>
<td>2.70</td>
<td>3.18</td>
</tr>
</tbody>
</table>

Fig. 1: Preference test results.

SEGAN yields better speech enhancement results than Noisy and Wiener.
Speech Enhancement

- Pix2Pix [Michelsanti et al., Interpsech 2017]
Speech Enhancement (Pix2Pix)

- Spectrogram analysis

Fig. 2: Spectrogram comparison of Pix2Pix with baseline methods.

Pix2Pix outperforms STAT-MMSE and is competitive to DNN SE.
Speech Enhancement (Pix2Pix)

- Objective evaluation and speaker verification test

Table 3: Objective evaluation results.

<table>
<thead>
<tr>
<th>Babble</th>
<th>SNR</th>
<th>PESQ</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>(a)</td>
<td>1.20</td>
<td>1.42</td>
</tr>
<tr>
<td>(b)</td>
<td>1.14</td>
<td>1.31</td>
</tr>
<tr>
<td>(c)</td>
<td><strong>1.25</strong></td>
<td>1.51</td>
</tr>
<tr>
<td>(d)</td>
<td>1.20</td>
<td>1.48</td>
</tr>
<tr>
<td>(e)</td>
<td>1.24</td>
<td><strong>1.52</strong></td>
</tr>
<tr>
<td>(f)</td>
<td>1.20</td>
<td>1.49</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Speech Enhancement (Pix2Pix)</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Objective evaluation and speaker verification test</td>
</tr>
</tbody>
</table>

Table 4: Speaker verification results.

<table>
<thead>
<tr>
<th>Airplane</th>
<th>SNR</th>
<th>0</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>clean</th>
<th>mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Babble</td>
<td>(a)</td>
<td>21.09</td>
<td>15.99</td>
<td>13.61</td>
<td>11.66</td>
<td>9.18</td>
<td>6.99</td>
<td><strong>13.08</strong></td>
</tr>
<tr>
<td>(b)</td>
<td>17.69</td>
<td>12.58</td>
<td>8.17</td>
<td>6.53</td>
<td>6.27</td>
<td>5.80</td>
<td>9.51</td>
<td></td>
</tr>
<tr>
<td>(c)</td>
<td>16.99</td>
<td>10.55</td>
<td>7.48</td>
<td>6.99</td>
<td>6.15</td>
<td>6.12</td>
<td>9.05</td>
<td></td>
</tr>
<tr>
<td>(d)</td>
<td>17.19</td>
<td>8.84</td>
<td><strong>5.44</strong></td>
<td>5.05</td>
<td><strong>4.63</strong></td>
<td>3.74</td>
<td>7.48</td>
<td></td>
</tr>
<tr>
<td>(e)</td>
<td>15.99</td>
<td>8.99</td>
<td>6.12</td>
<td>6.12</td>
<td>5.58</td>
<td>5.67</td>
<td>8.08</td>
<td></td>
</tr>
<tr>
<td>(f)</td>
<td><strong>15.31</strong></td>
<td>7.89</td>
<td>5.58</td>
<td><strong>4.77</strong></td>
<td>4.76</td>
<td>5.44</td>
<td><strong>7.29</strong></td>
<td></td>
</tr>
</tbody>
</table>

1. From the objective evaluations, Pix2Pix outperforms Noisy and MMSE and is competitive to DNN SE.
2. From the speaker verification results, Pix2Pix outperforms the baseline models when the clean training data is used.
Speech Enhancement

- Frequency-domain SEGAN (FSEGAN) [Donahue et al., ICASSP 2018]
Speech Enhancement (FSEGAN)

- Spectrogram analysis

Fig. 2: Spectrogram comparison of FSEGAN with L1-trained method.

(a) Noisy speech input $\alpha$
(b) L1-trained output $G'(\alpha)$
(c) Clean speech target $y$
(d) FSEGAN output $G(\alpha)$

FSEGAN reduces both additive noise and reverberant smearing.
Speech Enhancement (FSEGAN)

• ASR results

1. From Table 5, (1) FSEGAN improves recognition results for ASR-Clean.
   (2) FSEGAN outperforms SEGAN as front-ends.

2. From Table 6, (1) Hybrid Retraining with FSEGAN outperforms Baseline;
   (2) FSEGAN retraining slightly underperforms L1–based retraining.

### Table 5: WER (%) of SEGAN and FSEGAN.

<table>
<thead>
<tr>
<th>Test Set</th>
<th>Enhancer</th>
<th>ASR-Clean WER</th>
<th>ASR-MTR WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean</td>
<td>None</td>
<td>11.9</td>
<td>14.3</td>
</tr>
<tr>
<td>MTR</td>
<td>None</td>
<td>72.2</td>
<td>20.3</td>
</tr>
<tr>
<td></td>
<td>SEGAN</td>
<td>80.7</td>
<td>52.8</td>
</tr>
<tr>
<td></td>
<td>FSEGAN</td>
<td>33.3</td>
<td>25.4</td>
</tr>
</tbody>
</table>

### Table 6: WER (%) of FSEGAN with retrain.

<table>
<thead>
<tr>
<th>Model</th>
<th>WER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTR Baseline *</td>
<td>20.3</td>
</tr>
<tr>
<td>+ Stereo</td>
<td>19.0</td>
</tr>
<tr>
<td>MTR + FSEGAN Enhancer * + Hybrid Retraining</td>
<td>17.6</td>
</tr>
<tr>
<td>MTR + L1-trained Enhancer * + Hybrid Retraining</td>
<td>17.1</td>
</tr>
</tbody>
</table>
Speech Enhancement

• Adversarial training based mask estimation (ATME)  
  [Higuchi et al., ASRU 2017]

\[ V_{Mask} = E_{s_{fake}}[\log(1 - D_S(s_{fake}, \theta))] + E_{n_{fake}}[\log(1 - D_N(n_{fake}, \theta))] \]
Speech Enhancement (ATME)

• Spectrogram analysis

Fig. 3: Spectrogram comparison of (a) noisy; (b) MMSE with supervision; (c) ATMB without supervision.

The proposed adversarial training mask estimation can capture speech/noise signals without supervised data.
**Speech Enhancement (ATME)**

- Mask-based beamformer for robust ASR

  - The estimated mask parameters are used to compute spatial covariance matrix for MVDR beamformer.

  \[ \hat{s}_{f,t} = w_f^H y_{f,t} \]

  where \( \hat{s}_{f,t} \) is the enhanced signal, and \( y_{f,t} \) denotes the observation of \( M \) microphones, \( f \) and \( t \) are frequency and time indices; \( w_f \) denotes the beamformer coefficient.

  - The MVDR solves \( w_f \) by:

    \[ w_f = \frac{h_f (R_f^{(s+n)})^{-1}}{h_f^H (R_f^{(s+n)})^{-1} h_f} \]

  - To estimate \( h_f \), the spatial covariance matrix of the target signal, \( R_f^{(s)} \), is computed by:

    \[ R_f^{(s)} = R_f^{(s+n)} - R_f^{(n)} \]

    where \( R_f^{(n)} = \frac{M_f^{(n)} y_{f,t} y_{f,t}^H}{\Sigma_{f,t} M_f^{(n)}} \)

    \( M_f^{(n)} \) was computed by AT.
Speech Enhancement (ATME)

• ASR results

Table 7: WERs (%) for the development and evaluation sets.

<table>
<thead>
<tr>
<th>systems</th>
<th>avg</th>
<th>bus</th>
<th>caf</th>
<th>ped</th>
<th>str</th>
<th>avg</th>
<th>bus</th>
<th>caf</th>
<th>ped</th>
<th>str</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unprocessed</td>
<td>9.01</td>
<td>14.00</td>
<td>7.94</td>
<td>6.03</td>
<td>8.05</td>
<td>15.60</td>
<td>22.55</td>
<td>16.21</td>
<td>12.89</td>
<td>10.74</td>
</tr>
<tr>
<td>Adversarial Training</td>
<td>5.00</td>
<td>7.60</td>
<td>4.09</td>
<td>4.03</td>
<td>4.29</td>
<td>7.58</td>
<td>10.24</td>
<td>7.51</td>
<td>6.20</td>
<td>6.39</td>
</tr>
</tbody>
</table>

1. ATME provides significant improvements over Unprocessed.
2. Unsupervised ATME slightly underperforms supervised MMSE.
Speech Enhancement (AFT)

• Cycle-GAN-based acoustic feature transformation (AFT)
  [Mimura et al., ASRU 2017]

\[
V_{Full} = V_{GAN}(G_{X\rightarrow Y}, D_Y) + V_{GAN}(G_{X\rightarrow Y}, D_Y) + \lambda V_{Cyc}(G_{X\rightarrow Y}, G_{Y\rightarrow X})
\]
Speech Enhancement (AFT)

- ASR results on noise robustness and style adaptation

1. $G_{T \rightarrow S}$ can transform acoustic features and effectively improve ASR results for both noisy and accented speech.
2. $G_{S \rightarrow T}$ can be used for model adaptation and effectively improve ASR results for noisy speech.

### Table 8: Noise robust ASR.

<table>
<thead>
<tr>
<th>acoustic model</th>
<th>feature</th>
<th>cycle loss</th>
<th>$\lambda$ and $\mu$</th>
<th>WER</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>no adapt.</td>
<td>no adapt.</td>
<td>-</td>
<td>-</td>
<td>41.08</td>
<td>(1)</td>
</tr>
<tr>
<td>no adapt.</td>
<td>adapt. with $G_{T \rightarrow S}$</td>
<td>no</td>
<td>1, 1</td>
<td>55.45</td>
<td>(2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>yes</td>
<td>1, 1</td>
<td>37.34</td>
<td>(3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>yes</td>
<td>trained</td>
<td>36.56</td>
<td>(4)</td>
</tr>
<tr>
<td>adapt. with $G_{S \rightarrow T}$</td>
<td>no adapt.</td>
<td>yes</td>
<td>1, 1</td>
<td>35.98</td>
<td>(5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>yes</td>
<td>trained</td>
<td>34.31</td>
<td>(6)</td>
</tr>
</tbody>
</table>

**S**: Clean; **T**: Noisy

### Table 9: Speaker style adaptation.

<table>
<thead>
<tr>
<th>source</th>
<th>target</th>
<th>feature</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>JNAS</td>
<td>CSJ-SPS</td>
<td>no adapt.</td>
<td>26.47</td>
</tr>
<tr>
<td></td>
<td></td>
<td>adapt. with $G_{T \rightarrow S}$</td>
<td>25.93</td>
</tr>
<tr>
<td>CSJ-APS</td>
<td>CSJ-SPS</td>
<td>no adapt.</td>
<td>17.15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>adapt. with $G_{T \rightarrow S}$</td>
<td>16.60</td>
</tr>
</tbody>
</table>

**JNAS**: Read; **CSJ-SPS**: Spontaneous (relax); **CSJ-APS**: Spontaneous (formal);
Outline of Part II

**Speech Signal Generation**
- Speech enhancement
- Postfilter, speech synthesis, voice conversion

**Speech Signal Recognition**
- Speech recognition
- Speaker recognition
- Speech emotion recognition
- Lip reading

**Conclusion**
• Postfilter for synthesized or transformed speech

- Conventional postfilter approaches for $G$ estimation include global variance (GV) [Toda et al., IEICE 2007], variance scaling (VS) [Sil’én et al., Interpseech 2012], modulation spectrum (MS) [Takamichi et al., ICASSP 2014], DNN with MSE criterion [Chen et al., Interspeech 2014; Chen et al., TASLP 2015].
- GAN is used a new objective function to estimate the parameters in $G$. 
Postfilter

- **GAN postfilter** [Kaneko et al., ICASSP 2017]

- Traditional MMSE criterion results in statistical averaging.
- GAN is used as a new objective function to estimate the parameters in $G$.
- The proposed work intends to further improve the naturalness of synthesized speech or parameters from a synthesizer.
Postfilter (GAN-based Postfilter)

- Spectrogram analysis

Fig. 4: Spectrograms of: (a) NAT (nature); (b) SYN (synthesized); (c) VS (variance scaling); (d) MS (modulation spectrum); (e) MSE; (f) GAN postfilters.

GAN postfilter reconstructs spectral texture similar to the natural one.
Postfilter (GAN-based Postfilter)

- Objective evaluations

Fig. 5: Mel-cepstral trajectories (GANv: GAN was applied in voiced part).

Fig. 6: Averaging difference in modulation spectrum per Mel-cepstral coefficient.

GAN postfilter reconstructs spectral texture similar to the natural one.
Postfilter (GAN-based Postfilter)

- Subjective evaluations

Table 10: Preference score (%). Bold font indicates the numbers over 30%.

<table>
<thead>
<tr>
<th></th>
<th>Former</th>
<th>Latter</th>
<th>Neutral</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GAN vs. SYN</strong></td>
<td>56.5 ± 4.9</td>
<td>22.0 ± 4.1</td>
<td>21.5 ± 4.0</td>
</tr>
<tr>
<td><strong>GAN vs. GANv</strong></td>
<td>11.3 ± 3.1</td>
<td><strong>37.3 ± 4.8</strong></td>
<td><strong>51.5 ± 4.9</strong></td>
</tr>
<tr>
<td><strong>GAN vs. NAT</strong></td>
<td>16.8 ± 3.7</td>
<td><strong>53.5 ± 4.9</strong></td>
<td>29.8 ± 4.5</td>
</tr>
<tr>
<td><strong>GANv vs. NAT</strong></td>
<td><strong>30.3 ± 4.5</strong></td>
<td><strong>34.5 ± 4.7</strong></td>
<td><strong>35.3 ± 4.7</strong></td>
</tr>
</tbody>
</table>

1. GAN postfilter significantly improves the synthesized speech.
2. GAN postfilter is effective particularly in voiced segments.
3. GANv outperforms GAN and is comparable to NAT.
Postfilter (GAN-postfilter-SFTF)

• GAN post-filter for STFT spectrograms [Kaneko et al., Interspeech 2017]

- GAN postfilter was applied on high-dimensional STFT spectrograms.
- The spectrogram was partitioned into $N$ bands (each band overlaps its neighboring bands).
- The GAN-based postfilter was trained for each band.
- The reconstructed spectrogram from each band was smoothly connected.
Postfilter (GAN-postfilter-SFTF)

- Spectrogram analysis

Fig. 7: Spectrograms of: (1) SYN, (2) GAN, (3) Original (NAT)

GAN postfilter reconstructs spectral texture similar to the natural one.
Speech Synthesis

- Speech synthesis with anti-spoofing verification (ASV)
  [Saito et al., ICASSP 2017]

\[
L(c, \hat{c}) = L_G(c, \hat{c}) + \omega_D \frac{E_{LG}}{E_{LD}} L_{D,1}(\hat{c})
\]
Minimum generation error (MGE) with adversarial loss.

\[
L_D(c, \hat{c}) = L_{D,1}(c) + L_{D,0}(\hat{c})
\]
\[
L_{D,1}(c) = -\frac{1}{T} \sum_{t=1}^{T} \log(D(c_t)) \ldots \text{NAT}
\]
\[
L_{D,0}(\hat{c}) = -\frac{1}{T} \sum_{t=1}^{T} \log(1 - D(\hat{c}_t)) \ldots \text{SYN}
\]
Speech Synthesis (ASV)

- Objective and subjective evaluations

1. The proposed algorithm generates MCCs similar to the natural ones.
2. The proposed algorithm outperforms conventional MGE training.
Speech Synthesis

- Speech synthesis with GAN (SS-GAN) [Saito et al., TASLP 2018]

\[ L(c, \hat{c}) = L_G(c, \hat{c}) + \omega_D \frac{E_{LG}}{E_{LD}} L_{D,1}(\hat{c}) \]

Minimum generation error (MGE) with adversarial loss.

\[
\begin{align*}
L_D(c, \hat{c}) &= L_{D,1}(c) + L_{D,0}(\hat{c}) \\
L_{D,1}(c) &= -\frac{1}{T} \sum_{t=1}^{T} \log(D(c_t)) \ldots \text{NAT} \\
L_{D,0}(\hat{c}) &= -\frac{1}{T} \sum_{t=1}^{T} \log(1 - D(\hat{c}_t)) \ldots \text{SYN}
\end{align*}
\]
Speech Synthesis (SS-GAN)

- Subjective evaluations

The proposed algorithm works for both spectral parameters and F0.
Speech Synthesis

• Speech synthesis with GAN glottal waveform model (GlottGAN) [Bollepalli et al., Interspeech 2017]
Speech Synthesis (GlottGAN)

- Objective evaluations

The proposed GAN-based approach can generate glottal waveforms similar to the natural ones.

Fig. 12: Glottal pulses generated by GANs.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Ref</td>
<td>GAN</td>
<td>CGAN</td>
<td>CGAN+CNN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CGAN+CNN+LS</td>
</tr>
</tbody>
</table>

The proposed GAN-based approach can generate glottal waveforms similar to the natural ones.
Speech Synthesis

- Speech synthesis with GAN & multi-task learning (SS-GAN-MTL) [Yang et al., ASRU 2017]

![Diagram showing speech synthesis process]

\[ V_{GAN}(G,D) = E_{x \sim p_{data}(x)}[logD(x|y)] + E_{z \sim p_z}[log(1 - D(G(z|y)))|y] \]

\[ V_{L2}(G) = E_{z \sim p_z}[G(z|y) - x]^2 \]
Speech Synthesis (SS-GAN-MTL)

- Speech synthesis with GAN & multi-task learning (SS-GAN-MTL) [Yang et al., ASRU 2017]

\[
V_{GAN}(G, D) = E_{x \sim p_{data}(x)}[\log D_{CE}(x | y, \text{label})] \\
+ E_{z \sim p_z} [\log (1 - D_{CE}(G(z | y)) | y, \text{label}]
\]

\[
V_{L2}(G) = E_{z \sim p_z} [G(z | y) - x]^2
\]
Speech Synthesis (SS-GAN-MTL)

- Objective and subjective evaluations

Table 11: Objective evaluation results.

<table>
<thead>
<tr>
<th>Methods</th>
<th>MCD (dB)</th>
<th>$F_0$ RMSE (Hz)</th>
<th>V/UV (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLSTM</td>
<td>4.624</td>
<td>18.544</td>
<td>6.447</td>
</tr>
<tr>
<td>GAN</td>
<td>4.633</td>
<td>18.678</td>
<td>6.492</td>
</tr>
<tr>
<td>GAN-PC</td>
<td>4.628</td>
<td>18.616</td>
<td>6.464</td>
</tr>
</tbody>
</table>

Fig. 13: The preference score (%).

1. From objective evaluations, no remarkable difference is observed.
2. From subjective evaluations, GAN outperforms BLSTM and ASV, while GAN-PC underperforms GAN.
Voice Conversion

• Convert (transform) speech from source to target

- Conventional VC approaches include Gaussian mixture model (GMM) [Toda et al., TASLP 2007], non-negative matrix factorization (NMF) [Wu et al., TASLP 2014; Fu et al., TBME 2017], locally linear embedding (LLE) [Wu et al., Interspeech 2016], restricted Boltzmann machine (RBM) [Chen et al., TASLP 2014], feed forward NN [Desai et al., TASLP 2010], recurrent NN (RNN) [Nakashika et al., Interspeech 2014].
Conventional MMSE approaches often encounter the “over-smoothing” issue.

GAN is used a new objective function to estimate $G$.

The goal is to increase the naturalness, clarity, similarity of converted speech.

$$V(G, D) = V_{GAN}(G, D) + \lambda V_{VAE}(x|y)$$
Voice Conversion (VAW-GAN)

- Objective and subjective evaluations

VAW-GAN outperforms VAE in terms of objective and subjective evaluations with generating more structured speech.
Voice Conversion

• Sequence-to-sequence VC with learned similarity metric (LSM) [Kaneko et al., Interspeech 2017]

\[ V(C, G, D) = V_{SVC}^{D_l}(C, D) + V_{GAN}(C, G, D) \]
Voice Conversion (LSM)

- Spectrogram analysis

Fig. 16: Comparison of MCCs (upper) and STFT spectrograms (lower).

The spectral textures of LSM are more similar to the target ones.
Voice Conversion (LSM)

- Subjective evaluations

Table 12: Preference scores for naturalness.

<table>
<thead>
<tr>
<th></th>
<th>Former</th>
<th>Latter</th>
<th>Neutral</th>
</tr>
</thead>
<tbody>
<tr>
<td>FVC vs. LSM</td>
<td>17.1 ± 6.3</td>
<td>72.9 ± 7.5</td>
<td>10.0 ± 5.0</td>
</tr>
<tr>
<td>MSE vs. LSM</td>
<td>10.0 ± 5.0</td>
<td>84.3 ± 6.1</td>
<td>5.7 ± 3.9</td>
</tr>
</tbody>
</table>

Table 12: Preference scores for clarity.

<table>
<thead>
<tr>
<th></th>
<th>Former</th>
<th>Latter</th>
<th>Neutral</th>
</tr>
</thead>
<tbody>
<tr>
<td>FVC vs. LSM</td>
<td>32.9 ± 7.9</td>
<td>54.3 ± 8.4</td>
<td>12.9 ± 5.6</td>
</tr>
<tr>
<td>MSE vs. LSM</td>
<td>27.1 ± 7.5</td>
<td>65.0 ± 8.0</td>
<td>7.9 ± 4.5</td>
</tr>
</tbody>
</table>

Fig. 17: Similarity of TGT and SRC with VCs.

Voice Conversion (LSM) outperforms FVC and MSE in terms of subjective evaluations.
Voice Conversion

- **CycleGAN-VC** [Kaneko et al., arXiv 2017]

\[
V_{Full} = V_{GAN}(G_{X \rightarrow Y}, D_Y) + V_{GAN}(G_{X \rightarrow Y}, D_Y) + \lambda V_{Cyc}(G_{X \rightarrow Y}, G_{Y \rightarrow X})
\]

Scalar: belongs to domain S or not

Scalar: belongs to domain T or not

as close as possible
Voice Conversion (CycleGAN-VC)

- Subjective evaluations

Fig. 18: MOS for naturalness.

Fig. 19: Similarity of to source and to target speakers. S: Source; T: Target; P: Proposed; B: Baseline

1. The proposed method uses **non-parallel** data.
2. For naturalness, the proposed method outperforms baseline.
3. For similarity, the proposed method is comparable to the baseline.
Voice Conversion

- Multi-target VC [Chou et al., arxiv 2018]

Stage-1

Stage-2
Voice Conversion (Multi-target VC)

• Subjective evaluations

1. The proposed method uses non-parallel data.
2. The multi-target VC approach outperforms one-stage only.
3. The multi-target VC approach is comparable to Cycle-GAN-VC in terms of the naturalness and the similarity.
Outline of Part II

Speech Signal Generation

• Speech enhancement
• Postfilter, speech synthesis, voice conversion

Speech Signal Recognition

• Speech recognition
• Speaker recognition
• Speech emotion recognition
• Lip reading

Conclusion
Speech, Speaker, Emotion Recognition and Lip-reading (Classification Task)

Domain Adversarial Training

Acoustic Mismatch

\[ y \]

Output label

\[ \tilde{z} = g(\tilde{x}) \]

\[ g(\cdot) \]

\[ E \]

\[ G \]

\[ \text{Emb.} \]

\[ x \]

Clean data

\[ \tilde{x} \]

Noisy data

\[ \bar{x} \]

Accented speech

\[ \hat{x} \]

Channel distortion

Not only cheat the domain classifier, but satisfying label predictor at the same time

Successfully applied on image classification

[Ganin et al, ICML, 2015][Ajakan et al. JMLR, 2016]
Speech Recognition

- Adversarial multi-task learning (AMT)
  [Shinohara Interspeech 2016]

Objective function

\[ V_y = - \sum_i \log P(y_i | x_i; \theta_E, \theta_G) \]
\[ V_z = - \sum_i \log P(z_i | x_i; \theta_E, \theta_D) \]

Model update

\[ \theta_G \leftarrow \theta_G - \epsilon \frac{\partial V_y}{\partial \theta_G} \]  \text{Max classification accuracy}
\[ \theta_D \leftarrow \theta_D - \epsilon \frac{\partial V_z}{\partial \theta_D} \]  \text{Max domain accuracy}
\[ \theta_E \leftarrow \theta_E - \epsilon \left( \frac{\partial V_y}{\partial \theta_E} \right) + \alpha \frac{\partial V_z}{\partial \theta_E} \]  \text{Max classification accuracy and Min domain accuracy}
Speech Recognition (AMT)

- ASR results in known (k) and unknown (unk) noisy conditions

Table 13: WER of DNNs with single-task learning (ST) and AMT.

<table>
<thead>
<tr>
<th>noise</th>
<th>ST</th>
<th>AMT</th>
<th>RERR</th>
</tr>
</thead>
<tbody>
<tr>
<td>k car 2000cc</td>
<td>5.83</td>
<td>5.56</td>
<td>4.63</td>
</tr>
<tr>
<td>k exhib. booth</td>
<td>6.80</td>
<td>6.66</td>
<td>2.06</td>
</tr>
<tr>
<td>k station</td>
<td>7.89</td>
<td>7.76</td>
<td>1.65</td>
</tr>
<tr>
<td>k crossing</td>
<td>6.96</td>
<td>6.65</td>
<td>4.45</td>
</tr>
<tr>
<td>unk car 1500cc</td>
<td>5.58</td>
<td>5.46</td>
<td>2.15</td>
</tr>
<tr>
<td>unk exhib. aisle</td>
<td>7.71</td>
<td>6.93</td>
<td>10.12</td>
</tr>
<tr>
<td>unk factory</td>
<td>12.17</td>
<td>12.92</td>
<td>-6.16</td>
</tr>
<tr>
<td>unk highway</td>
<td>9.73</td>
<td>9.52</td>
<td>2.16</td>
</tr>
<tr>
<td>unk crowd</td>
<td>6.72</td>
<td>6.40</td>
<td>4.76</td>
</tr>
<tr>
<td>unk server room</td>
<td>8.54</td>
<td>7.76</td>
<td>9.13</td>
</tr>
<tr>
<td>unk air cond.</td>
<td>6.96</td>
<td>6.98</td>
<td>-0.29</td>
</tr>
<tr>
<td>unk elev. hall</td>
<td>9.23</td>
<td>9.60</td>
<td>-4.01</td>
</tr>
<tr>
<td>average</td>
<td>7.84</td>
<td>7.68</td>
<td>2.04</td>
</tr>
</tbody>
</table>

The AMT-DNN outperforms ST-DNN with yielding lower WERs.
Speech Recognition

• Domain adversarial training for accented ASR (DAT)
  [Sun et al., ICASSP2018]

<Diagram>

Objective function

\[ V_y = - \sum_i \log P(y_i|x_i; \theta_E, \theta_G) \]
\[ V_z = - \sum_i \log P(z_i|x_i; \theta_E, \theta_D) \]

Model update

\[ \theta_G \leftarrow \theta_G - \epsilon \frac{\partial V_y}{\partial \theta_G} \]
\[ \theta_D \leftarrow \theta_D - \epsilon \frac{\partial V_z}{\partial \theta_D} \]
\[ \theta_E \leftarrow \theta_E - \epsilon \left( \frac{\partial V_y}{\partial \theta_E} \right) + \alpha \frac{\partial V_z}{\partial \theta_E} \]
Speech Recognition (DAT)

- ASR results on accented speech

Table 14: WER of the baseline and adapted model.

<table>
<thead>
<tr>
<th>training data</th>
<th>λ</th>
<th>STD</th>
<th>FJ</th>
<th>JS</th>
<th>JX</th>
<th>SC</th>
<th>GD</th>
<th>HN</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD</td>
<td>-</td>
<td>15.55</td>
<td>23.58</td>
<td>15.75</td>
<td>14.08</td>
<td>15.62</td>
<td>15.32</td>
<td>19.34</td>
<td>17.28</td>
</tr>
<tr>
<td>STD + (600hrs with trans)</td>
<td>-</td>
<td>14.22</td>
<td>14.84</td>
<td>9.41</td>
<td>8.68</td>
<td>9.13</td>
<td>9.62</td>
<td>11.89</td>
<td>10.60</td>
</tr>
<tr>
<td>STD + (600hrs no trans)</td>
<td>0.03</td>
<td>15.37</td>
<td>22.96</td>
<td>14.48</td>
<td>13.79</td>
<td>15.35</td>
<td>14.86</td>
<td>18.24</td>
<td>16.61</td>
</tr>
</tbody>
</table>

STD: standard speech

1. With labeled transcriptions, ASR performance notably improves.
2. DAT is effective in learning features invariant to domain differences with and without labeled transcriptions.
Speech Recognition

- Robust ASR using GAN enhancer (GAN-Enhancer)
  [Sriram et al., arXiv 2017]

Cross entropy with L1 Enhancer:
\[ H(h(\tilde{z}), y) + \lambda \frac{||z - \tilde{z}||_1}{||z||_1 + ||\tilde{z}||_1 + \epsilon} \]

Cross entropy with GAN Enhancer:
\[ H(h(\tilde{z}), y) + \lambda V_{adv}(g(x), g(\tilde{x})) \]
Speech Recognition (GAN-Enhancer)

- ASR results on far-field speech:

Fig. 15: WER of GAN enhancer and the baseline methods.

<table>
<thead>
<tr>
<th>Model</th>
<th>Near-Field</th>
<th></th>
<th>Far-Field</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CER</td>
<td>WER</td>
<td>CER</td>
</tr>
<tr>
<td>seq-to-seq</td>
<td>7.43%</td>
<td>21.18%</td>
<td>23.76%</td>
</tr>
<tr>
<td>seq-to-seq + far-field Augmentation</td>
<td>7.69%</td>
<td>21.32%</td>
<td>12.47%</td>
</tr>
<tr>
<td>seq-to-seq + $L^1$-Distance Penalty</td>
<td>7.54%</td>
<td>20.45%</td>
<td>12.00%</td>
</tr>
<tr>
<td>seq-to-seq + GAN Enhancer</td>
<td>7.78%</td>
<td>21.07%</td>
<td><strong>11.26%</strong></td>
</tr>
</tbody>
</table>

GAN Enhancer outperforms the Augmentation and L1-Enhancer approaches on far-field speech.
Outline of Part II

Speech Signal Generation
• Speech enhancement
• Postfilter, speech synthesis, voice conversion

Speech Signal Recognition
• Speech recognition
• Speaker recognition
• Speech emotion recognition
• Lip reading

Conclusion
Speaker Recognition

- Domain adversarial neural network (DANN)  
[Wang et al., ICASSP 2018]

Diagram:
- **Input** (Acoustic feature) $x$ to **Enroll** $E$ and **Test** $E$  
  - **Enroll** $E$ outputs $i$-vector to **GRL** and then to **DANN**  
  - **Test** $E$ outputs $i$-vector to **DANN**

- **Domain** adversarial neural network (DANN)
  - **Pre-processing** (GRL) $V_y$ to $G$ and $V_z$ to $D$
  - **Output 1** (Speaker ID)
  - **Output 2** (Domain)
Speaker Recognition (DANN)

- Recognition results of domain mismatched conditions

Table 16: Performance of DAT and the state-of-the-art methods.

<table>
<thead>
<tr>
<th>Systems#</th>
<th>Adaptation Methods</th>
<th>EER%</th>
<th>DCF10 [21]</th>
<th>DCF08</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>–</td>
<td>9.35</td>
<td>0.724</td>
<td>0.520</td>
</tr>
<tr>
<td>2</td>
<td>–</td>
<td>5.66</td>
<td>0.633</td>
<td>0.427</td>
</tr>
<tr>
<td>3</td>
<td>Interpolated [6] [12]</td>
<td>6.55</td>
<td>0.652</td>
<td>0.454</td>
</tr>
<tr>
<td>4</td>
<td>IDV [9] [12]</td>
<td>6.15</td>
<td>0.676</td>
<td>0.476</td>
</tr>
<tr>
<td>5</td>
<td>DICN [11] [12]</td>
<td>4.99</td>
<td>0.623</td>
<td>0.416</td>
</tr>
<tr>
<td>6</td>
<td>DAE [22] [12]</td>
<td>4.81</td>
<td>0.610</td>
<td>0.398</td>
</tr>
<tr>
<td>7</td>
<td>AEDA [12]</td>
<td>4.50</td>
<td>0.589</td>
<td>0.362</td>
</tr>
<tr>
<td>8</td>
<td>DAT</td>
<td>3.73</td>
<td><strong>0.541</strong></td>
<td><strong>0.335</strong></td>
</tr>
</tbody>
</table>

The DAT approach outperforms other methods with achieving lowest EER and DCF scores.
Outline of Part II

Speech Signal Generation

- Speech enhancement
- Postfilter, speech synthesis, voice conversion

Speech Signal Recognition

- Speech recognition
- Speaker recognition
- Speech emotion recognition
- Lip reading

Conclusion
Emotion Recognition

- Adversarial AE for emotion recognition (AAE-ER)
  [Sahu et al., Interspeech 2017]

$$AE \text{ with GAN : } H(h(z), x) + \lambda V_{GAN}(q, g(x))$$

The distribution of code vectors
Emotion Recognition (AAE-ER)

• Recognition results of domain mismatched conditions:

<table>
<thead>
<tr>
<th></th>
<th>OpenSmile features (1582-D)</th>
<th>Code vectors (2-D)</th>
<th>Auto-encoder (100-D)</th>
<th>LDA (2-D)</th>
<th>PCA (2-D)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UAR (%)</td>
<td>57.88</td>
<td>56.38</td>
<td>53.92</td>
<td>48.67</td>
<td>43.12</td>
</tr>
</tbody>
</table>

1. AAE alone could not yield performance improvements.
2. Using synthetic data from AAE can yield higher UAR.
Outline of Part II

Speech Signal Generation
- Speech enhancement
- Postfilter, speech synthesis, voice conversion

Speech Signal Recognition
- Speech recognition
- Speaker recognition
- Speech emotion recognition
- Lip reading

Conclusion
Lip-reading

- Domain adversarial training for lip-reading (DAT-LR)
  [Wand et al., arXiv 2017]

Output 1
Words

Output 2
Speaker

Objective function

\[ V_y = -\sum_i \log P(y_i | x_i; \theta_E, \theta_G) \]
\[ V_z = -\sum_i \log P(z_i | x_i; \theta_E, \theta_D) \]

Model update

\[ \theta_G \leftarrow \theta_G - \epsilon \frac{\partial V_y}{\partial \theta_G} \]
Max classification accuracy

\[ \theta_D \leftarrow \theta_D - \epsilon \frac{\partial V_z}{\partial \theta_D} \]
Max domain accuracy

\[ \theta_E \leftarrow \theta_E - \epsilon \left( \frac{\partial V_y}{\partial \theta_E} \right) + \alpha \frac{\partial V_z}{\partial \theta_E} \]
Max classification accuracy and Min domain accuracy

~80% WAC
Lip-reading (DAT-LR)

- Recognition results of speaker mismatched conditions

Table 19: Performance of DAT and the baseline.

<table>
<thead>
<tr>
<th>Adversarial Training on</th>
<th>Number of training spk</th>
<th>Target Test acc.</th>
<th>Relative Improvement</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>1</td>
<td>18.7%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>39.4%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>46.5%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>All Target Sequences</td>
<td>1</td>
<td>25.4%</td>
<td>35.8%</td>
<td>0.0030*</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>43.6%</td>
<td>10.7%</td>
<td>0.0261*</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>49.3%</td>
<td>6.0%</td>
<td>0.0266*</td>
</tr>
<tr>
<td>50 Target Sequences</td>
<td>1</td>
<td>24.1%</td>
<td>28.9%</td>
<td>0.0045*</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>41.5%</td>
<td>5.3%</td>
<td>0.1367</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>47.0%</td>
<td>1.1%</td>
<td>0.3555</td>
</tr>
</tbody>
</table>

The DAT approach notably enhances the recognition accuracies in different conditions.
Outline of Part II

Speech Signal Generation

• Speech enhancement
• Postfilter, speech synthesis, voice conversion

Speech Signal Recognition

• Speech recognition
• Speaker recognition
• Speech emotion recognition
• Lip reading

Conclusion
Speech Signal Generation (Regression Task)

Paired

Objective function

[Scott Reed, et al, ICML, 2016]

Conditional GAN

Prior distribution $z$

c: train

G

Image $x = G(c,z)$

c

D (better)

$D$ (better)

scalar $x$ is realistic or not + c and $x$ are matched or not

True text-image pairs: (train, ) 1

(cat, ) 0

(train, Image ) 0

Cycle-GAN

as close as possible

$G_{X \rightarrow Y}$

$D_X$

scalar: belongs to domain X or not

$D_Y$

scalar: belongs to domain Y or not

$G_{Y \rightarrow X}$

$G_{X \rightarrow Y}$

as close as possible
Speech, Speaker, Emotion Recognition and Lip-reading (Classification Task)

Domain Adversarial Training

Not only cheat the domain classifier, but satisfying label predictor at the same time

Successfully applied on image classification

[ Ganin et al, ICML, 2015][ Ajakan et al, JMLR, 2016 ]

Acoustic Mismatch

Channel distortion

Accented speech

Noisy data

Clean data

Output label

\[ g(\cdot) \]

\[ h(\cdot) \]

\[ \tilde{z} = g(\tilde{x}) \]

\[ E \]

\[ G \]
More GANs in Speech

**Diagnosis of autism spectrum**

**Emotion recognition**

**Robust ASR**

**Speaker verification**
References

Speech enhancement (conventional methods)

• Yuxuan Wang and Deliang Wang, Cocktail Party Processing via Structured Prediction, NIPS 2012.

Speech enhancement (GAN-based methods)

• Donahue Chris, Li Bo, and Prabhavalkar Rohit, Exploring Speech Enhancement with Generative Adversarial Networks for Robust Speech Recognition, ICASPP, 2018.
• Higuchi Takuya, Kinoshita Keisuke, Delcroix Marc, and Nakatani Tomohiro, Adversarial Training for Data-driven Speech Enhancement without Parallel Corpus, ASRU, 2017.
Postfilter (conventional methods)


Postfilter (GAN-based methods)

References

VC (conventional methods)


VC (GAN-based methods)

References

ASR
• Sining Sun, Ching-Feng Yeh, Mei-Yuh Hwang, Mari Ostendorf, and Lei Xie, Domain Adversarial Training for Accented Speech Recognition, ICASSP, 2018

Speaker recognition
• Qing Wang, Wei Rao, Sining Sun, Lei Xie, Eng Siong Chng, and Haizhou Li, Unsupervised Domain Adaptation via Domain Adversarial Training for Speaker Recognition, ICASSP, 2018.

Emotion recognition

Lipreading
GANs in ICASSP 2018

- Saurabh Sahu, Rahul Gupta, Ganesh Sivaraman, Carol Espy-Wilson, Smoothing Model Predictions using Adversarial Training Procedures for Speech Based Emotion Recognition
- Fuming Fang, Junichi Yamagishi, Isao Echizen, Jaime Lorenzo-Trueba, High-quality Nonparallel Voice Conversion Based on Cycle-consistent Adversarial Network
- Lauri Juvela, Bajibabu Bollepalli, Xin Wang, Hirokazu Kameoka, Manu Airaksinen, Junichi Yamagishi, Paavo Alku, Speech Waveform Synthesis from MFCC Sequences with Generative Adversarial Networks
- Zhong Meng, Jinyu Li, Yifan Gong, Biing-Hwang (Fred) Juang, Adversarial Teacher-Student Learning for Unsupervised Domain Adaptation
- Zhong Meng, Jinyu Li, Zhuo Chen, Yong Zhao, Vadim Mazalov, Yifan Gong, Biing-Hwang (Fred) Juang, Speaker-Invariant Training via Adversarial Learning
- Sen Li, Stephane Villette, Pravin Ramadas, Daniel Sinder, Speech Bandwidth Extension using Generative Adversarial Networks
- Qing Wang, Wei Rao, Sining Sun, Lei Xie, Eng Siong Chng, Haizhou Li, Unsupervised Domain Adaptation via Domain Adversarial Training for Speaker Recognition
- Hu Hu, Tian Tan, Yanmin Qian, Generative Adversarial Networks Based Data Augmentation for Noise Robust Speech Recognition
- Yuki Saito, Shinnosuke Takamichi, Hiroshi Saruwatari, Text-to-speech Synthesis using STFT Spectra Based on Low-/multi-resolution Generative Adversarial Networks
- Jiangyan Yi, Jianhua Tao, Zhengqi Wen, Ye Bai, Adversarial Multilingual Training for Low-resource Speech Recognition
- Taira Tsuchiya, Naohiro Tawara, Tetsuji Ogawa, Tetsunori Kobayashi, Speaker Invariant Feature Extraction for Zero-resource Languages with Adversarial Learning
GANs in ICASSP 2018

- Jing Han, Zixing Zhang, Zhao Ren, Fabien Ringeval, Bjoern Schuller, Towards Conditional Adversarial Training for Predicting Emotions from Speech
- Chenxing Li, Lei Zhu, Shuang Xu, Peng Gao, Bo Xu, CBLDNN-based Speaker-independent Speech Separation via Generative Adversarial Training
- Anuroop Sriram, Heewoo Jun, Yashesh Gaur, Sanjeev Satheesh, Robust Speech Recognition using Generative Adversarial Networks
- Cem Subakan, Paris Smaragdis, Generative Adversarial Source Separation,
- Ashutosh Pandey, Deliang Wang, On Adversarial Training and Loss Functions for Speech Enhancement
- Bin Liu, Shuai Nie, Yaping Zhang, Dengfeng Ke, Shan Liang, Wenju Liu, Boosting Noise Robustness of Acoustic Model via Deep Adversarial Training
- Yang Gao, Rita Singh, Bhiksha Raj, Voice Impersonation using Generative Adversarial Networks
- Aditay Tripathi, Aanchan Mohan, Saket Anand, Maneesh Singh, Adversarial Learning of Raw Speech Features for Domain Invariant Speech Recognition
A promising research direction and still has room for further improvements in the speech signal processing domain

Thank You Very Much

Tsao, Yu   Ph.D.

yu.tsao@citi.sinica.edu.tw

https://www.citi.sinica.edu.tw/pages/yu.tsao/contact_zh.html
Generative Adversarial Network
and its Applications to Signal Processing
and Natural Language Processing

Part III: Natural Language Processing
Outline of Part III

Conditional Sequence Generation

- RL (human feedback)
- GAN (discriminator feedback)

Unsupervised Conditional Sequence Generation

- Text Style Transfer
- Unsupervised Abstractive Summarization
- Unsupervised Translation
Conditional Sequence Generation

The generator is a typical seq2seq model.
With GAN, you can train seq2seq model in another way.
Review: Sequence-to-sequence

- Chat-bot as example

<table>
<thead>
<tr>
<th>Output:</th>
<th>Not bad</th>
<th>I’m John.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human</td>
<td>better</td>
<td></td>
</tr>
<tr>
<td>Training Criterion</td>
<td></td>
<td>better</td>
</tr>
</tbody>
</table>

Maximize likelihood

Training data:

A: How are you?
B: I’m good.

Input sentence c

How are you?

Generator

Encoder

Decoder
Outline of Part III

**Improving Supervised Seq-to-seq Model**
- RL (human feedback)
- GAN (discriminator feedback)

**Unsupervised Seq-to-seq Model**
- Text Style Transfer
- Unsupervised Abstractive Summarization
- Unsupervised Translation
Introduction

• Machine obtains feedback from user

   How are you?
   Bye bye 😊
   -10

   Hello
   Hi 😊
   3

• Chat-bot learns to maximize the **expected reward**
Maximizing Expected Reward

Learn to maximize expected reward

Input sentence c  \rightarrow  Chatbot  \rightarrow  response sentence x

Input sentence c  \rightarrow  Human  \rightarrow  \textbf{R}(c, x)

[Li, et al., EMNLP, 2016]
Policy Gradient - Implementation

\[ \theta^t \xrightarrow{\text{→}} \theta^t + \eta \nabla R_{\theta^t} \]

\[
\frac{1}{N} \sum_{i=1}^{N} R(c^i, x^i) \nabla \log P_{\theta^t}(x^i | c^i)
\]

- If \( R(c^i, x^i) \) is positive, updating \( \theta \) to increase \( P_{\theta}(x^i | c^i) \)
- If \( R(c^i, x^i) \) is negative, updating \( \theta \) to decrease \( P_{\theta}(x^i | c^i) \)

\( (c^1, x^1), R(c^1, x^1) \)
\( (c^2, x^2), R(c^2, x^2) \)
\( \vdots \)
\( (c^N, x^N), R(c^N, x^N) \)
## Comparison

<table>
<thead>
<tr>
<th></th>
<th>Maximum Likelihood</th>
<th>Reinforcement Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Objective Function</strong></td>
<td>[ \frac{1}{N} \sum_{i=1}^{N} \log P_\theta(\hat{x}^i</td>
<td>c^i) ]</td>
</tr>
<tr>
<td><strong>Gradient</strong></td>
<td>[ \frac{1}{N} \sum_{i=1}^{N} \nabla \log P_\theta(\hat{x}^i</td>
<td>c^i) ]</td>
</tr>
<tr>
<td><strong>Training Data</strong></td>
<td>{ (c^1, \hat{x}^1), \ldots, (c^N, \hat{x}^N) } \quad \text{( R(c^i, \hat{x}^i) = 1 )}</td>
<td>{ (c^1, x^1), \ldots, (c^N, x^N) } \quad \text{obtained from interaction weighted by } R(c^i, x^i) }</td>
</tr>
</tbody>
</table>
Outline of Part III

Improving Supervised Seq-to-seq Model

- RL (human feedback)
- GAN (discriminator feedback)

Unsupervised Seq-to-seq Model

- Text Style Transfer
- Unsupervised Abstractive Summarization
- Unsupervised Translation
Conditional GAN

Input sentence \( c \) \( \rightarrow \) En, De \( \rightarrow \) response sentence \( x \)

Input sentence \( c \) \( \rightarrow \) Discriminator \( \rightarrow \) Real or fake “reward”

response sentence \( x \) \( \rightarrow \) human dialogues

[Li, et al., EMNLP, 2017]
Algorithm

• Initialize generator G (chatbot) and discriminator D

• In each iteration:
  • Sample input $c$ and response $x$ from training set
  • Sample input $c'$ from training set, and generate response $\tilde{x}$ by $G(c')$
  • Update D to increase $D(c, x)$ and decrease $D(c', \tilde{x})$

• Update generator G (chatbot) such that

Training data:
Pairs of conditional input $c$ and response $x$
Can we use gradient ascent?

NO!

Due to the sampling process, "discriminator + generator" is not differentiable.
Three Categories of Solutions

Gumbel-softmax

• [Matt J. Kusner, et al., arXiv, 2016]

Continuous Input for Discriminator


“Reinforcement Learning”

Use the distribution as the input of discriminator

Avoid the sampling process

We can do backpropagation now.
What is the problem?

• Real sentence

• Generated

Can never be 1-of-N

Discriminator can immediately find the difference.

WGAN is helpful
Three Categories of Solutions

Gumbel-softmax


Continuous Input for Discriminator

- [Sai Rajeswar, et al., arXiv, 2017]
- [Ofir Press, et al., ICML workshop, 2017]
- [Zhen Xu, et al., EMNLP, 2017]
- [Alex Lamb, et al., NIPS, 2016]
- [Yizhe Zhang, et al., ICML, 2017]

“Reinforcement Learning”

- [Yu, et al., AAAI, 2017]
- [Li, et al., EMNLP, 2017]
- [Tong Che, et al, arXiv, 2017]
- [Jiaxian Guo, et al., AAAI, 2018]
- [Kevin Lin, et al, NIPS, 2017]
- [William Fedus, et al., ICLR, 2018]
Reinforcement Learning?

- Consider the output of discriminator as reward
  - Update generator to increase discriminator = to get maximum reward
  - Using the formulation of policy gradient, replace reward $R(c, x)$ with discriminator output $D(c, x)$

- Different from typical RL
  - The discriminator would update
\[ g\text{-step} \]

\[ d\text{-step} \]

\[ \theta^t \]

\[ \theta^{t+1} \leftarrow \theta^t + \eta \nabla \bar{R}_{\theta^t} \]

\[ \frac{1}{N} \sum_{i=1}^{N} D(c^i, x^i) \nabla \log P_{\theta^t}(x^i | c^i) \]

\[ D(c^i, x^i) \]

- is positive
- Updating \( \theta \) to increase \( P_{\theta}(x^i | c^i) \)

- is negative
- Updating \( \theta \) to decrease \( P_{\theta}(x^i | c^i) \)
Reward for Every Generation Step

\[ \nabla \bar{R}_\theta \approx \frac{1}{N} \sum_{i=1}^{N} D(c^i, x^i) \nabla \log P_\theta(x^i|c^i) \]

\( c^i = \text{“What is your name?”} \quad D(c^i, x^i) \text{ is negative} \)
\( x^i = \text{“I don’t know”} \quad \text{Update } \theta \text{ to decrease } \log P_\theta(x^i|c^i) \)

\[ \log P_\theta(x^i|c^i) = \log P(x_1^i|c^i) + \log P(x_2^i|c^i, x_1^i) + \log P(x_3^i|c^i, x_{1:2}^i) \]

\( P(\text{"I"}|c^i) \) \( \rightarrow \)

\( c^i = \text{“What is your name?”} \quad D(c^i, x^i) \text{ is positive} \)
\( x^i = \text{“I am John”} \quad \text{Update } \theta \text{ to increase } \log P_\theta(x^i|c^i) \)

\[ \log P_\theta(x^i|c^i) = \log P(x_1^i|c^i) + \log P(x_2^i|c^i, x_1^i) + \log P(x_3^i|c^i, x_{1:2}^i) \]

\( P(\text{"I"}|c^i) \) \( \rightarrow \)
**Reward for Every Generation Step**

\[ h^i = \text{"What is your name?"} \quad x^i = \text{"I don’t know"} \]

\[
\log P_\theta(x^i|h^i) = \log P(x^i_1|c^i) + \log P(x^i_2|c^i, x^i_1) + \log P(x^i_3|c^i, x^i_1:2)
\]

\[
P(\text{"I"}|c^i) \quad P(\text{"don’t"}|c^i, \text{"I"}) \quad P(\text{"know"}|c^i, \text{"I don’t"})
\]

\[
\nabla \tilde{R}_\theta \approx \frac{1}{N} \sum_{i=1}^{N} D(c^i, x^i) \nabla \log P_\theta(x^i|c^i)
\]

\[
\nabla \tilde{R}_\theta \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} (Q(c^i, x^i_{1:t}) - b) \nabla \log P_\theta(x^i_{t}|c^i, x^i_{1:t-1})
\]

Method 2. Discriminator For Partially Decoded Sequences [Li, et al., EMNLP, 2017]
**Experimental Results**

<table>
<thead>
<tr>
<th>Input</th>
<th>We've got to look for another route.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLE</td>
<td>I'm sorry.</td>
</tr>
<tr>
<td>GAN</td>
<td>You're not going to be here for a while.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input</th>
<th>You can save him by talking.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLE</td>
<td>I don't know.</td>
</tr>
<tr>
<td>GAN</td>
<td>You know what's going on in there, you know what I mean?</td>
</tr>
</tbody>
</table>

- MLE frequently generates “I’m sorry”, “I don’t know”, etc. (corresponding to fuzzy images?)
- GAN generates longer and more complex responses (however, no strong evidence shows that they are better)

Find more comparison in the survey papers.

More Applications


• Supervised abstractive summarization [Liu, et al., AAAI 2018]

• Image/video caption generation [Rakshith Shetty, et al., ICCV 2017][Liang, et al., arXiv 2017]

If you are using seq2seq models, consider to improve them by GAN.
Outline of Part III

Conditional Sequence Generation

- RL (human feedback)
- GAN (discriminator feedback)

Unsupervised Conditional Sequence Generation

- Text Style Transfer
- Unsupervised Abstractive Summarization
- Unsupervised Translation
Text Style Transfer

Domain X

male

It is good.
It’s a good day.
I love you.

positive sentences

Domain Y

female

It is bad.
It’s a bad day.
I don’t love you.

negative sentences
Direct Transformation

\[ G_{X \rightarrow Y} \rightarrow Y \rightarrow X \rightarrow G_{Y \rightarrow X} \rightarrow \text{scalar: belongs to domain } X \text{ or not} \rightarrow D_{Y} \rightarrow \text{scalar: belongs to domain } Y \text{ or not} \rightarrow G_{Y \rightarrow X} \rightarrow \text{as close as possible} \rightarrow G_{X \rightarrow Y} \rightarrow \text{as close as possible} \]
Direct Transformation

It is bad. negative

$G_{X \rightarrow Y}$

It is good. positive

$G_{Y \rightarrow X}$

It is bad. negative

$D_{Y}$

positive sentence?

$I$ love you. positive

$G_{Y \rightarrow X}$

$I$ hate you. negative

$G_{X \rightarrow Y}$

$I$ love you. positive

negative sentence?

$D_{X}$

as close as possible

as close as possible
Direct Transformation

\[ G_X \rightarrow Y \]

as close as possible

\[ G_Y \rightarrow X \]

negative sentence?

positive sentence?

Discrete?

Word embedding
[Lee, et al., ICASSP, 2018]
• **Negative** sentence to **positive** sentence:
  it's a crappy day → it's a great day
  i wish you could be here → you could be here
  it's not a good idea → it's good idea
  i miss you → i love you
  i don't love you → i love you
  i can't do that → i can do that
  i feel so sad → i happy
  it's a bad day → it's a good day
  it's a dummy day → it's a great day
  sorry for doing such a horrible thing → thanks for doing a great thing
  my doggy is sick → my doggy is my doggy
  my little doggy is sick → my little doggy is my little doggy

**Title:** SCALABLE SENTIMENT FOR SEQUENCE-TO-SEQUENCE CHATBOT RESPONSE WITH PERFORMANCE ANALYSIS

**Session:** Dialog Systems and Applications

**Time:** Wednesday, April 18, 08:30 - 10:30

**Authors:** Chih-Wei Lee, Yau-Shian Wang, Tsung-Yuan Hsu, Kuan-Yu Chen, Hung-Yi Lee, Lin-Shan Lee
Projection to Common Space

\[
\begin{align*}
EN_X & 
\rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & D_X \\
DN_Y & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & \rightarrow & D_Y \\
\end{align*}
\]

Discriminator of X domain
Discriminator of Y domain
Projection to Common Space

Decoder hidden layer as discriminator input

[Shen, et al., NIPS, 2017]

$EN_X$ and $EN_Y$ fool the domain discriminator

[Zhao, et al., arXiv, 2017]
[Fu, et al., AAAI, 2018]
Outline of Part III

**Improving Supervised Seq-to-seq Model**
- RL (human feedback)
- GAN (discriminator feedback)

**Unsupervised Seq-to-seq Model**
- Text Style Transfer
- Unsupervised Abstractive Summarization
- Unsupervised Translation
Abstractive Summarization

• Now machine can do **abstractive summary** by seq2seq (write summaries in its own words)

Supervised: We need lots of labelled training data.
Unsupervised Abstractive Summarization

The two seq2seq models are jointly learn to minimize the reconstruction error.

Only need a lot of documents to train the model.
Unsupervised Abstractive Summarization

This is a **seq2seq2seq auto-encoder**.

Using a sequence of words as latent representation.

Policy gradient is used.

---

**seq2seq2seq auto-encoder**

1. **long document**
2. **short document**
3. **long document**

**Summary?**
Unsupervised Abstractive Summarization

Human written summaries

Let Discriminator consider my output as real

Seq2seq

Summary?

Seq2seq

Real or not

long document

short document

long document
Semi-supervised Learning

The graph shows the performance of different learning methods on the ROUGE-2 metric. The methods compared are:

- WGAN
- Adversarial REINFORCE
- Supervised

The x-axis represents the amount of labeled data (0, 10K, 50K, 100K, 3.8M(full)), and the y-axis represents the ROUGE-2 score. The graph illustrates that semi-supervised learning (green line) performs better than unsupervised learning (red line) but is still inferior to supervised learning (blue line). The graph is unpublished.
Outline of Part III

Improving Supervised Seq-to-seq Model

- RL (human feedback)
- GAN (discriminator feedback)

Unsupervised Seq-to-seq Model

- Text Style Transfer
- Unsupervised Abstractive Summarization
- Unsupervised Translation
Unsupervised Machine Translation

Domain X

[Images of various domains]

male

Domain Y

[Images of various domains]

female

[Alexis Conneau, et al., ICLR, 2018]
[Guillaume Lample, et al., ICLR, 2018]
Unsupervised learning with 10M sentences = Supervised learning with 100K sentence pairs
Unsupervised Speech Recognition

Can we achieve unsupervised speech recognition?

Acoustic Pattern Discovery

Cycle GAN

The dog is ......
The cats are ......
The woman is ......
The cat is ......
The man is ......

$p_1$, $p_2$, $p_3$, $p_4$

$Liu$, et al., arXiv, 2018] [Chen, et al., arXiv, 2018]
Unsupervised Speech Recognition

- Phoneme recognition

Audio: TIMIT
Text: WMT

supervised

Gumbel-softmax

WGAN-GP
Concluding Remarks

Conditional Sequence Generation

- RL (human feedback)
- GAN (discriminator feedback)

Unsupervised Conditional Sequence Generation

- Text Style Transfer
- Unsupervised Abstractive Summarization
- Unsupervised Translation
To Learn More ...

You can learn more from the YouTube Channel

https://www.youtube.com/playlist?list=PLJV_el3uVTsMd2G9ZjcpJn1YfnM9wVOBf

(in Mandarin)
Reference

• **Conditional Sequence Generation**
  
  
  
  
  • Tong Che, Yanran Li, Ruixiang Zhang, R Devon Hjelm, Wenjie Li, Yangqiu Song, Yoshua Bengio, *Maximum-Likelihood Augmented Discrete Generative Adversarial Networks*, arXiv 2017
  
  • Lantao Yu, Weinan Zhang, Jun Wang, Yong Yu, *SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient*, AAAI 2017
Reference

• **Conditional Sequence Generation**
  • Sai Rajeswar, Sandeep Subramanian, Francis Dutil, Christopher Pal, Aaron Courville, Adversarial Generation of Natural Language, arXiv, 2017
  • Ofir Press, Amir Bar, Ben Bogin, Jonathan Berant, Lior Wolf, Language Generation with Recurrent Generative Adversarial Networks without Pre-training, ICML workshop, 2017
  • Zhen Xu, Bingquan Liu, Baoxun Wang, Chengjie Sun, Xiaolong Wang, Zhuoran Wang, Chao Qi, Neural Response Generation via GAN with an Approximate Embedding Layer, EMNLP, 2017
  • Alex Lamb, Anirudh Goyal, Ying Zhang, Saizheng Zhang, Aaron Courville, Yoshua Bengio, Professor Forcing: A New Algorithm for Training Recurrent Networks, NIPS, 2016
  • Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, Lawrence Carin, Adversarial Feature Matching for Text Generation, ICML, 2017
  • Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, Jun Wang, Long Text Generation via Adversarial Training with Leaked Information, AAAI, 2018
  • Kevin Lin, Dianqi Li, Xiaodong He, Zhengyou Zhang, Ming-Ting Sun, Adversarial Ranking for Language Generation, NIPS, 2017
  • William Fedus, Ian Goodfellow, Andrew M. Dai, MaskGAN: Better Text Generation via Filling in the______, ICLR, 2018
Reference

- Conditional Sequence Generation
  - Sidi Lu, Yaoming Zhu, Weinan Zhang, Jun Wang, Yong Yu, Neural Text Generation: Past, Present and Beyond, arXiv, 2018
  - Zhen Yang, Wei Chen, Feng Wang, Bo Xu, Improving Neural Machine Translation with Conditional Sequence Generative Adversarial Nets, NAACL, 2018
  - Linqing Liu, Yao Lu, Min Yang, Qiang Qu, Jia Zhu, Hongyan Li, Generative Adversarial Network for Abstractive Text Summarization, AAAI 2018
  - Rakshith Shetty, Marcus Rohrbach, Lisa Anne Hendricks, Mario Fritz, Bernt Schiele, Speaking the Same Language: Matching Machine to Human Captions by Adversarial Training, ICCV 2017
Reference

• **Text Style Transfer**
  
  • Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, Rui Yan, Style Transfer in Text: Exploration and Evaluation, AAAI, 2018
  
  • Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola, Style Transfer from Non-Parallel Text by Cross-Alignment, NIPS 2017
  
  • Chih-Wei Lee, Yau-Shian Wang, Tsung-Yuan Hsu, Kuan-Yu Chen, Hung-Yi Lee, Lin-shan Lee, Scalable Sentiment for Sequence-to-sequence Chatbot Response with Performance Analysis, ICASSP, 2018
  
  • Junbo (Jake) Zhao, Yoon Kim, Kelly Zhang, Alexander M. Rush, Yann LeCun, Adversarially Regularized Autoencoders, arxiv, 2017
Reference

• **Unsupervised Machine Translation**
  • Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, Hervé Jégou, Word Translation Without Parallel Data, ICRL 2018
  • Guillaume Lample, Ludovic Denoyer, Marc'Aurelio Ranzato, Unsupervised Machine Translation Using Monolingual Corpora Only, ICRL 2018

• **Unsupervised Speech Recognition**
  • Da-Rong Liu, Kuan-Yu Chen, Hung-Yi Lee, Lin-shan Lee, Completely Unsupervised Phoneme Recognition by Adversarially Learning Mapping Relationships from Audio Embeddings, arXiv, 2018
  • Yi-Chen Chen, Chia-Hao Shen, Sung-Feng Huang, Hung-yi Lee, Towards Unsupervised Automatic Speech Recognition Trained by Unaligned Speech and Text only, arXiv, 2018