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

Hung-yi Lee

²We 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.
Outline

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

Part II: Applications to Speech Processing

Part III: Applications to Natural Language Processing
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

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

Each dimension of input vector represents some characteristics.

- Longer hair
- Blue hair
- Open mouth
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

![Diagram](image)
**Algorithm**

- Initialize generator and discriminator
- In each training iteration:
  - Sample some real objects:
  - Generate some fake objects:
  - Update

Learning $D$

Learning $G$
The faces generated by machine.

The images are generated by Yen-Hao Chen, Po-Chun Chien, Jun-Chen Xie, Tsung-Han Wu.
Amazing Results!

[Tero Karras, et al., ICLR, 2018]
Amazing Results!

[Andrew Brock, et al., arXiv, 2018]
(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
- = Generator
- Fuzzy ...

**GAN**

- code
- Generator
- Discriminator
- If discriminator does not simply memorize the images, Generator learns the patterns of faces.
FID [Martin Heusel, et al., NIPS, 2017]: 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

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

\[ \star : \text{data sampled from } P_{data} \]
\[ \star : \text{data sampled from } P_G \]

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

Example Objective Function for D

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

(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 

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

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

Training:

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

Discriminator train easy to discriminate

Discriminator train hard to discriminate

small divergence

large divergence

: data sampled from \( P_{data} \)

: data sampled from \( P_G \)
The maximum objective value is related to JS divergence.

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

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

- 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 \( G \)
### 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>$\left(\frac{1-u}{u}\right)^2$</td>
</tr>
<tr>
<td>Squared Hellinger</td>
<td>$\int \left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2 , dx$</td>
<td>$\left(\sqrt{u} - 1\right)^2$</td>
</tr>
<tr>
<td>Jeffrey</td>
<td>$\int (p(x) - q(x)) \log \frac{p(x)}{q(x)} , dx$</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>$\int p(x) \pi \log \frac{q(x)}{q(x)+1-p(x)} + (1-\pi)q(x) \log \frac{q(x)}{q(x)+1-\pi} , dx$</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

More tips and tricks: https://github.com/soumith/gan hacks

Conditional Generation

Unsupervised Conditional Generation

Relation to Reinforcement Learning
GAN is hard to train ......

NO PAIN
NO GAN

(I found this joke from 陳柏文’s facebook.)
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?

\[
JS(P_{G_0}, P_{data}) = \log 2 \\
JS(P_{G_1}, P_{data}) = \log 2 \\
\cdots \\
JS(P_{G_{100}}, P_{data}) = 0
\]

JS divergence is \( \log 2 \) 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?

\[ JS(P_{G_0}, P_{data}) = \log_2 \]
\[ W(P_{G_0}, P_{data}) = d_0 \]

\[ JS(P_{G_1}, P_{data}) = \log_2 \]
\[ W(P_{G_1}, P_{data}) = d_1 \]

\[ \ldots \]

\[ JS(P_{G_{100}}, P_{data}) = 0 \]
\[ W(P_{G_{100}}, P_{data}) = 0 \]

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 → 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 → Gradient Penalty** [Ishaan Gulrajani, NIPS, 2017]
  Keep the gradient close to 1

  [Kodali, et al., arXiv, 2017]
  [Wei, et al., ICLR, 2018]

• **Spectral Normalization → 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]
**Tip: Improve Quality during Testing**

Some samples are poor.

Smaller Variance

The output would be more stable, but sacrifice the diversity.

This tip is also used in [Andrew Brock, et al., arXiv, 2018]
Mode Collapse

Training with too many iterations ......
Mode Dropping

Generator switches mode during training

Generator at iteration $t$

Generator at iteration $t+1$

Generator at iteration $t+2$

BEGAN on CelebA
Tip: Ensemble

To generate an image

Random pick a generator $G_i$, and then use $G_i$ to generate the image

Train a set of generators: $\{G_1, G_2, \cdots, G_N\}$
Objective Evaluation

$x$: image
$y$: class (output of CNN)

Concentrated distribution means higher visual quality

Uniform distribution means higher variety

$P(y|x) = \frac{1}{N} \sum_n P(y^n|x^n)$

$P(y^n|x^n)$

$P(y^1|x^1)$
$P(y^2|x^2)$
$P(y^3|x^3)$

$\cdot \cdot \cdot$

Off-the-shelf Image Classifier

e.g. Inception net, VGG, etc.
Objective Evaluation

\[ P(y|x) \]

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

**Inception Score** [Tim Salimans, et al., NIPS 2016]

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

Negative entropy of \( P(y|x) \)

Entropy of \( P(y) \)
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”

[Image of text-to-image example]
Text-to-Image

**Traditional supervised approach**

$c^1$: a dog is running

Text: “train”

Target of NN output

A blurry image!

a dog is running

a bird is flying

as close as possible
Conditional GAN

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

Normal distribution \( z \) → \( G \) → Image

\( c: \text{train} \)

\( x \) is real image or not

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

Real images: 1
Generated images: 0
Conditional GAN

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

Normal distribution \( z \) \( \rightarrow \) \( G \) \( \rightarrow \) \( \text{Image} \)

\( c: \text{train} \)

\( \text{D (better)} \)

\( x \) \( \rightarrow \) \( \text{scalar} \)

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

True text-image pairs:

- \( \text{(train , } \) \( \) 1
- \( \text{(cat , } \) \( ) \) 0
- \( \text{(train , Image} \) \( ) \) 0
Conditional GAN - Discriminator

Object x → Network → Network → score

Condition c → Network

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

Object x → Network → x is realistic or not

Condition c → Network → 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

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

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

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

- Traditional supervised approach

Testing:

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]
https://github.com/dyelax/Adversarial_Video_Generation
Conditional GAN - Sound-to-image

"a dog barking sound"

Training Data Collection

video
Conditional GAN - Sound-to-image

- Audio-to-image

The images are generated by Chia-Hung Wan and Shun-Po Chuang.
https://wjohn1483.github.io/audio_to_scene/index.html

Louder
Conditional GAN - Image-to-label

Multi-label Image Classifier

Input condition

Generated output

person, sports ball, baseball bat, baseball glove
The classifiers can have different architectures.

The classifiers are trained as conditional GAN.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>MS-COCO</th>
<th>NUS-WIDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG-16</td>
<td>56.0</td>
<td>33.9</td>
</tr>
<tr>
<td>+ GAN</td>
<td>60.4</td>
<td>41.2</td>
</tr>
<tr>
<td>Inception</td>
<td>62.4</td>
<td>53.5</td>
</tr>
<tr>
<td>+ GAN</td>
<td>63.8</td>
<td>55.8</td>
</tr>
<tr>
<td>Resnet-101</td>
<td>62.8</td>
<td>53.1</td>
</tr>
<tr>
<td>+ GAN</td>
<td>64.0</td>
<td>55.4</td>
</tr>
<tr>
<td>Resnet-152</td>
<td>63.3</td>
<td>52.1</td>
</tr>
<tr>
<td>+ GAN</td>
<td>63.9</td>
<td>54.1</td>
</tr>
<tr>
<td>Att-RNN</td>
<td>62.1</td>
<td>54.7</td>
</tr>
<tr>
<td>RLSD</td>
<td>62.0</td>
<td>46.9</td>
</tr>
</tbody>
</table>

[Tsai, et al., submitted to ICASSP 2019]
Conditional GAN - Image-to-label

The classifiers can have different architectures.

The classifiers are trained as conditional GAN.

Conditional GAN outperforms other models designed for multi-label.

<table>
<thead>
<tr>
<th></th>
<th>MS-COCO</th>
<th>NUS-WIDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG-16</td>
<td>56.0</td>
<td>33.9</td>
</tr>
<tr>
<td>+ GAN</td>
<td>60.4</td>
<td>41.2</td>
</tr>
<tr>
<td>Inception</td>
<td>62.4</td>
<td>53.5</td>
</tr>
<tr>
<td>+GAN</td>
<td>63.8</td>
<td>55.8</td>
</tr>
<tr>
<td>Resnet-101</td>
<td>62.8</td>
<td>53.1</td>
</tr>
<tr>
<td>+GAN</td>
<td>64.0</td>
<td>55.4</td>
</tr>
<tr>
<td>Resnet-152</td>
<td>63.3</td>
<td>52.1</td>
</tr>
<tr>
<td>+GAN</td>
<td>63.9</td>
<td>54.1</td>
</tr>
<tr>
<td>Att-RNN</td>
<td>62.1</td>
<td>54.7</td>
</tr>
<tr>
<td>RLSD</td>
<td>62.0</td>
<td>46.9</td>
</tr>
</tbody>
</table>
Domain Adversarial Training

- Training and testing data are in different domains

Take digit classification as example
Domain Adversarial Training

**feature extractor (Generator)**

input x

image

**features f**

Always output zero vectors

Domain Classifier Fails

**Which domain?**

Discriminator (Domain classifier)

blue points

red points

Discriminator

domain label d
Domain Adversarial Training

- feature extractor (Generator)
- Label predictor
- Discriminator (Domain classifier)

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 GAN

Transform an object from one domain to another
without paired data

Use image style transfer as example here

More Applications in Parts II and III
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

\[ \mathcal{E} \mathcal{N}_X \rightarrow \mathcal{D} \mathcal{E}_Y \]

Encoder of domain X \rightarrow Decoder of domain Y

Face Attribute

Larger change, only keep the semantics
Direct Transformation

Domain X

\[ G_{X \rightarrow Y} \]

Becomes similar to domain Y

Domain Y

\[ D_Y \]

Scalar

Input image belongs to domain Y or not
Direct Transformation

\[ G_{X \rightarrow Y} \]

Become similar to domain Y

Not what we want!

\[ D_Y \]

scalar

Input image belongs to domain Y or not

ignore input

Domain X

Domain Y
Direct Transformation

Domain X \[ \rightarrow \quad G_{X \rightarrow Y} \quad \rightarrow \quad \text{Become similar to domain Y} \]

Not what we want!

ignore input

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

\[ G_{X \rightarrow Y} \]

Domain X

Encoder Network

Become similar to domain Y

Encoder Network

pre-trained

as close as possible

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

\[ D_Y \]

scalar

Input image belongs to domain Y or not

Domain Y
Direct Transformation – Cycle GAN

- As close as possible
- Cycle consistency

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

Lack of information for reconstruction

Input image belongs to domain Y or not

Domain Y

Direct Transformation – Cycle GAN

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

as close as possible

scalar: belongs to domain X or not

Scalar: belongs to domain Y or not
For multiple domains, considering starGAN

[Zunjey Choi, arXiv, 2017]

Disco GAN


Dual GAN

[Taeksoo Kim, et al., ICML, 2017]

Cycle GAN

[Zili Yi, et al., ICCV, 2017]

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**

![Diagram of Direct Transformation](image)

**• Approach 2: Projection to Common Space**

![Diagram of Projection to Common Space](image)
**Projection to Common Space**

**Target**

```
image → EN_X → EN_Y → Face Attribute → DE_Y → image → Domain X → Domain Y
```

- **EN_X** and **EN_Y**: Features extracted from Domain X and Domain Y respectively.
- **DE_X** and **DE_Y**: Domain-specific transformations.
- **Face Attribute**: Common representation.

**Domain X** and **Domain Y** are the source domains with images and face attributes corresponding to each domain.
Projection to Common Space

Minimizing reconstruction error

Training

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]
Minimizing reconstruction error

$EN_X$ 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

Discriminator of Y domain

Cycle Consistency:

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

Training

To the same latent space

Semantic Consistency:

Outline of Part 1

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

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

- **Video Game**: Example of a video game where the agent interacts with the environment.
- **Actor**: The agent that makes decisions or actions.
- **Reward Function**: The function that determines the reward for the agent based on its actions.
- **Env**: The environment in which the agent operates, including the game rules and dynamics.

Example in **Go**:
- **AlphaGo**: A deep learning program developed by Google DeepMind.
- **Lee Sedol**: A professional Go player.

The rule of **GO**:
- **You cannot control the game entirely**.
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
Actor, Environment, Reward

Trajectory

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

Actor $\rightarrow$ Generator

Reward Function $\rightarrow$ Discriminator

$R(\tau) = \sum_{t=1}^{T} r_t$
Imitation Learning

We have demonstration of the expert.

Self driving: record human drivers
Robot: grab the arm of robot

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

reward function is not available

$\{\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_N\}$
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.
Framework of IRL

- Expert $\hat{\pi}$
- Obtain Reward Function $R$
- The expert is always the best.

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

Actor $\pi$

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

$\{\hat{\tau}_1, \hat{\tau}_2, \ldots, \hat{\tau}_N\}$

Actor $\rightarrow$ Generator

Reward function $\rightarrow$ Discriminator

By Reinforcement learning

The expert is always the best.
**GAN**

High score for real, low score for generated

**IRL**

Expert Actor

Find a G whose output obtains large score from D

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

Find a Actor obtains large reward

\( \tau_1, \tau_2, \ldots, \tau_N \)

\( \{\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_N\} \)
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
  - Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, Olivier Bousquet, “Are GANs Created Equal? A Large-Scale Study”, arXiv, 2017
  - 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
  - Tero Karras, Timo Aila, Samuli Laine, Jaakko Lehtinen, Progressive Growing of GANs for Improved Quality, Stability, and Variation, ICLR, 2018
  - Andrew Brock, Jeff Donahue, Karen Simonyan, Large Scale GAN Training for High Fidelity Natural Image Synthesis, arXiv, 2018
Reference

• **Conditional 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

• **Conditional 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
  • Che-Ping Tsai, Hung-Yi Lee, Adversarial Learning of Label Dependency: A Novel Framework for Multi-class Classification, submitted to ICASSP 2019
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, 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
Outline

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

Part II: Applications to Natural Language Processing

Part III: Applications to Speech Processing
Unsupervised Conditional Generation

**Image Style Transfer**

photos

Not Paired

Vincent van Gogh’s paintings

**Text Style Transfer**

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

positive

Not Paired

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

negative
Cycle GAN

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

as close as possible

scalar: belongs to domain X or not

scalar: belongs to domain Y or not

as close as possible
Cycle GAN

It is bad. (negative) $\rightarrow G_{X\rightarrow Y}$ $\rightarrow$ It is good. (positive) $\rightarrow G_{Y\rightarrow X}$ $\rightarrow$ It is bad. (negative)

negative sentence? $\leftarrow D_X$ $\rightarrow$ positive sentence?

I love you. (positive) $\rightarrow G_{Y\rightarrow X}$ $\rightarrow$ I hate you. (negative) $\rightarrow G_{X\rightarrow Y}$ $\rightarrow$ I love you. (positive)

as close as possible

as close as possible
Discrete Issue

It is bad.

negative

$G_{X\rightarrow Y}$

update

It is good.

politive

$D_Y$

fix

Seq2seq model

hidden layer

with discrete output

positive sentence?

large network
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]
Cycle GAN

Discrete?
Word embedding
[Lee, et al., ICASSP, 2018]
Cycle GAN

- **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
Cycle GAN

Negative sentence to positive sentence:

胃疼,沒睡醒,各種不舒服 ->  生日快樂,睡醒,超級舒服

我都想去上班了, 真夠賤的! ->  我都想去睡了, 真帥的！

暈死了, 吃燒烤、竟然遇到個變態狂 -> 哈哈好~, 吃燒烤~竟然遇到帥狂

我肚子痛的厲害 ->  我生日快樂厲害

感冒了, 難受的說不出話來了! ->  感冒了, 開心的說不出話來！
**Projection to Common Space**

Decoder hidden layer as discriminator input

[Shen, et al., NIPS, 2017]

Positive Sentence $\rightarrow EN_X$ $\rightarrow DE_X$ $\rightarrow$ Positive Sentence $\rightarrow DX$

Negative Sentence $\rightarrow EN_Y$ $\rightarrow DE_Y$ $\rightarrow$ Negative Sentence $\rightarrow DY$

$EN_X$ and $EN_Y$ fool the domain discriminator

[Zhao, et al., ICML 2018]

[Fu, et al., AAAI, 2018]

Domain Discriminator

From $EN_X$ or $EN_Y$
Unsupervised Conditional Generation

**Image Style Transfer**

Unpaired photos vs. Vincent van Gogh’s paintings

**Text Style Transfer**

Unpaired document vs. summary

This is **unsupervised abstractive summarization**.
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

- Now machine can do **abstractive summary** by seq2seq (write summaries in its own words)
Unsupervised Abstractive Summarization

Human written summaries

Document

Seq2seq

Word sequence

Summary?

Real or not

Discriminator
Unsupervised Abstractive Summarization

Human written summaries → Real or not

Document → Seq2seq → word sequence → Discriminator → Seq2seq → Document

minimize the reconstruction error
Unsupervised Abstractive Summarization

Only need a lot of documents to train the model

This is a \textit{seq2seq2seq auto-encoder}.
Using a sequence of words as latent representation.

\begin{itemize}
\item document \rightarrow \text{G} \rightarrow \text{Seq2seq} \rightarrow \text{word sequence} \rightarrow \text{R} \rightarrow \text{Seq2seq} \rightarrow \text{document}
\end{itemize}

Summary?
Unsupervised Abstractive Summarization

Let Discriminator considers my output as real

Human written summaries → Real or not

Discriminator

REINFORCE algorithm to deal with the discrete issue

Seq2seq → word sequence

Readable

Summary?

Seq2seq → document
## Experimental results

### English Gigaword (Document title as summary)

<table>
<thead>
<tr>
<th></th>
<th>ROUGE-1</th>
<th>ROUGE-2</th>
<th>ROUGE-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised</td>
<td>33.2</td>
<td>14.2</td>
<td>30.5</td>
</tr>
<tr>
<td>Trivial</td>
<td>21.9</td>
<td>7.7</td>
<td>20.5</td>
</tr>
<tr>
<td>Unsupervised (matched data)</td>
<td>28.1</td>
<td>10.0</td>
<td>25.4</td>
</tr>
<tr>
<td>Unsupervised (no matched data)</td>
<td>27.2</td>
<td>9.1</td>
<td>24.1</td>
</tr>
</tbody>
</table>

- Matched data: using the title of English Gigaword to train Discriminator
- No matched data: using the title of CNN/Diary Mail to train Discriminator
Semi-supervised Learning

Using matched data

Approaches to deal with the discrete issue. 3.8M pairs are used.
Unsupervised Abstractive Summarization

• **Document**: 澳大利亞今天與13個國家簽署了反興奮劑雙邊協議，旨在加強體育競賽之外的藥品檢查並共享研究成果 ......

• **Summary**: 
  • Human: 澳大利亞與13國簽署反興奮劑協議
  • Unsupervised: 澳大利亞加強體育競賽之外的藥品檢查

• **Document**: 中華民國奧林匹克委員會今天接到一九九二年冬季奧運會邀請函，由於主席張豐緒目前正在中南美洲進行友好訪問，因此尚未決定是否派隊赴賽 ......

• **Summary**: 
  • Human: 一九九二年冬季奧運會函邀我參加
  • Unsupervised: 奧委會接獲冬季奧運會邀請函
Unsupervised Abstractive Summarization

• **Document**: 據此間媒體27日報道，印度尼西亞蘇門答臘島的兩個省近日來連降暴雨，洪水泛濫導致塌方，到26日為止至少已有60人喪生，100多人失蹤……

• **Summary**:
  • **Human**: 印尼水災造成60人死亡
  • **Unsupervised**: 印尼門洪水泛濫導致塌雨

• **Document**: 安徽省合肥市最近為領導幹部下基層做了新規定：一律輕車簡從，不準搞迎來送往、不準搞層層陪同……

• **Summary**:
  • **Human**: 合肥規定領導幹部下基層活動從簡
  • **Unsupervised**: 合肥領導幹部下基層做搞迎來送往規定：一律簡
Outline

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

Part II: Applications to Natural Language Processing

Part III: Applications to Speech Processing
Unsupervised Conditional Generation

**Image Style Transfer**

Not Paired

Vincent van Gogh’s paintings

Not Paired

photos

**Speech Style Transfer**

Not Paired

Speaker A

Not Paired

Speaker B

This is **unsupervised voice conversion**.
Voice Conversion
Voice Conversion

• The same sentence has different impact when it is said by different people.

Do you want to study a PhD?

Go away!

新垣結衣 (Aragaki Yui)

Do you want to study a PhD?

Student

Student

❤❤❤❤❤
In the past

How are you?  How are you?
Good morning  Good morning

With GAN

天氣真好  How are you?
再見囉  Good morning

Speakers A and B are talking about completely different things.
Cycle GAN

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

\[
D_{Y} \quad \text{scalar: belongs to domain Y or not} \quad D_{X} \quad \text{scalar: belongs to domain X or not}
\]

\[
G_{Y \rightarrow X} \quad \text{as close as possible} \quad G_{X \rightarrow Y}
\]
Cycle GAN for Voice Conversion

$spectrogram \xrightarrow{G_{X \rightarrow Y}}$ spectrogram $\xrightarrow{G_{Y \rightarrow X}}$ spectrogram

scalar: belongs to domain X or not $\xrightarrow{D_X} \xleftarrow{D_Y}$ scalar: belongs to domain Y or not

$\xrightarrow{G_{Y \rightarrow X}}$ spectrogram $\xrightarrow{G_{X \rightarrow Y}}$ spectrogram

X: Speaker A, Y: Speaker B

Projection to Common Space
**Projection to Common Space**

- All the speakers share the same encoder.
- The model can deal with the speakers never seen during training.
Projection to Common Space

Use a vector (one-hot) to represent speaker identity. All the speakers also share the same decoder.

The encoder fools the discriminator. Which speakers?

We hope that encoder can extract the phonetic information while removing the speaker information.
**Projection to Common Space**

**Training**

A is reading the sentence of B

A is reading the sentence of B

**Testing**

A is reading the sentence of B

A is reading the sentence of B
Encoder

How are you?

Different colors:
different words

Discriminator

Which speakers?

Does it contain phonetic information?

Different colors:
different speakers
“Audio” Word to Vector
**Issues**

**Training**
- A is reading the sentence of B
- How are you?
- The Same Speakers

**Testing**
- A is reading the sentence of B
- Hello
- Different Speakers

Discriminator
- Which speakers?
- reconstructed

A

B

Hello

Low Quality

Encoder

Decoder
2nd Stage Training

Extra Criterion for Training

Encoder → Decoder

Different Speakers

Cheat discriminator

Help speaker classifier

real or generated?

which speaker?

Discriminator

Speaker Classifier

No learning target???
Experimental Results

- Subjective evaluations (20 speakers in VCTK)

- “Two stages” is better
- “One stage” is better
- Indistinguishable

“Projection” is better
“Cycle GAN” is better
Indistinguishable

[Chou et al., INTERSPEECH, 2018]
Demo

Source Speaker: B

Target Speaker: A

Source to Target: Hello

A is reading the sentence of B

Thanks Ju-chieh Chou for providing the results.
https://jjery2243542.github.io/voice_conversion_demo/
Thanks Ju-chieh Chou for providing the results.
https://jjery2243542.github.io/voice_conversion_demo/
Unsupervised Conditional Generation

This is unsupervised speech recognition.
Supervised Speech Recognition

(I believe you have seen similar figures before.)

- Supervised learning needs lots of annotated speech.
- However, most of the languages are low resourced.
Speech Recognition in the Future

Learning human language with very little supervision
Unsupervised Speech Recognition

- Machine learns to recognize speech from unparallel speech and text.

This idea was too crazy to be realized in the past. However, it becomes possible with GAN recently.

[Liu, et al., INTERSPEECH, 2018]
[Chen, et al., arXiv, 2018]
Acoustic Token Discovery

Acoustic tokens can be discovered from audio collection without text annotation.

Acoustic tokens: chunks of acoustically similar audio segments with token IDs

[Zhang & Glass, ASRU 09]
[Huijbregts, ICASSP 11]
[Chan & Lee, Interspeech 11]
Acoustic Token Discovery

Acoustic tokens can be discovered from audio collection without text annotation.

Acoustic tokens: chunks of acoustically similar audio segments with token IDs

[Zhang & Glass, ASRU 09]
[Huijbregts, ICASSP 11]
[Chan & Lee, Interspeech 11]
Acoustic Token Discovery

Phonetic-level acoustic tokens are obtained by segmental sequence-to-sequence autoencoder.

[Wang, et al., ICASSP, 2018]
Unsupervised Speech Recognition

Phone-level Acoustic Pattern Discovery

Phoneme sequences from Text

[Chen, et al., arXiv, 2018]
[Liu, et al., INTERSPEECH, 2018]
<table>
<thead>
<tr>
<th>Approaches</th>
<th>Matched (all 4000)</th>
<th>Nonmatched (3000/1000)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FER</td>
<td>PER</td>
</tr>
<tr>
<td>(I) Supervised (labeled)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(a) RNN Transducer [23]</td>
<td></td>
<td>17.7</td>
</tr>
<tr>
<td>(b) standard HMMs</td>
<td></td>
<td>21.5</td>
</tr>
<tr>
<td>(c) Phoneme classifier</td>
<td>27.0</td>
<td>28.9</td>
</tr>
<tr>
<td>(II) Unsupervised (with oracle boundaries)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(d) Relationship mapping GAN [22]</td>
<td>40.5</td>
<td>40.2</td>
</tr>
<tr>
<td>(e) Segmental Empirical-ODM [23]</td>
<td>33.3</td>
<td>32.5</td>
</tr>
<tr>
<td>(f) Proposed: GAN</td>
<td>27.6</td>
<td>28.5</td>
</tr>
<tr>
<td>(III) Completely unsupervised (no label at all)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(g) Segmental Empirical-ODM [23]</td>
<td></td>
<td>36.5</td>
</tr>
<tr>
<td>Proposed iteration 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(h) GAN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(i) GAN/HMM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(j) GAN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proposed iteration 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(k) GAN/HMM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proposed iteration 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(l) GAN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(m) GAN/HMM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The progress of supervised learning

Unsupervised learning today (2019) is as good as supervised learning 30 years ago.

Concluding Remarks

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

Part II: Applications to Natural Language Processing

Part III: Applications to Speech Processing
To Learn More ...

You can learn more from the YouTube Channel

https://www.youtube.com/playlist?list=PLJV_el3uVTsMd2G9ZjcpJn1YfnM9wVOBf
(in Mandarin)