PR12와 함께 이해하는

GANs

Jaejun Yoo
Ph.D. Candidate @KAIST
PR12

16th Apr, 2017

* Generative Adversarial Nets Ian Goodfellow et al. 2014를 바탕으로 작성한 리뷰

안녕하세요 저는

유재준

KAIST Korea Advanced Institute of Science and Technology BiSPL Bio Imaging Signal Processing Lab.

- Ph.D. Candidate
- Medical Image Reconstruction, Topological Data Analysis, EEG
- http://jaejunyoo.blogspot.com/

Generative Adversarial Network

Generative Adversarial Network

PREREQUISITES

Generative Models

“FACE IMAGES”

PREREQUISITES

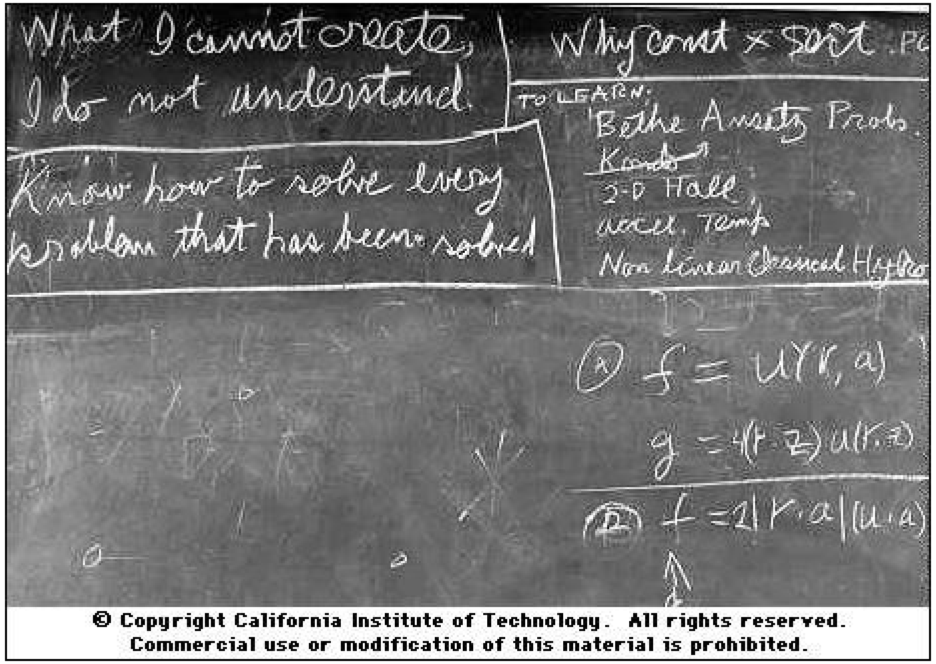
Generative Models

Generated Images by Neural Network

* Figure adopted from ***BEGAN*** paper released at 31. Mar. 2017
David Berthelot et al. Google (link)

PREREQUISITES

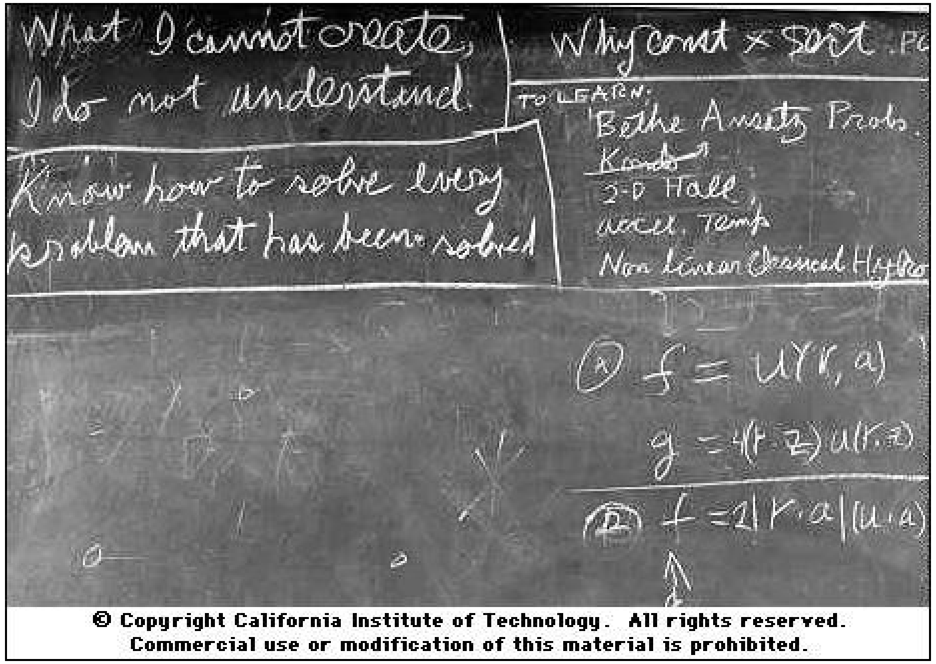
Generative Models



"What I cannot create, I do not understand"

PREREQUISITES

Generative Models

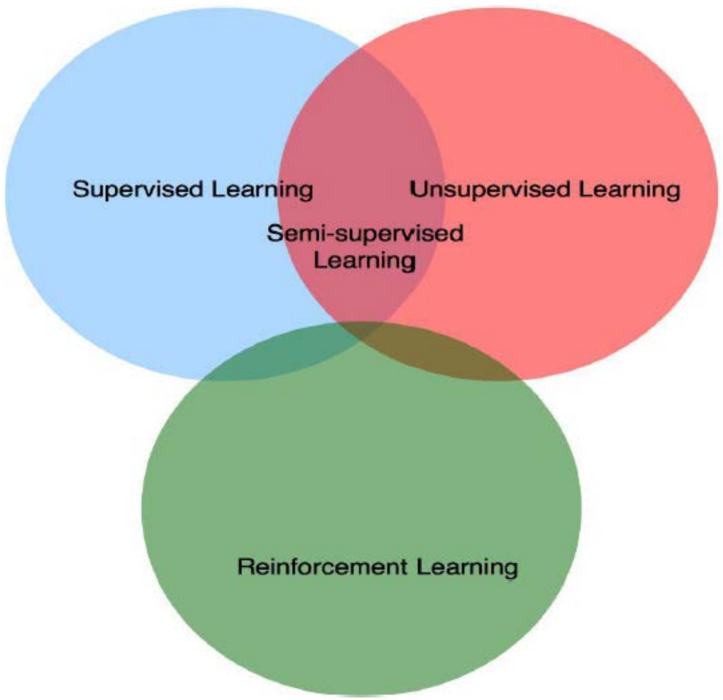


"What I cannot create, I do not understand"

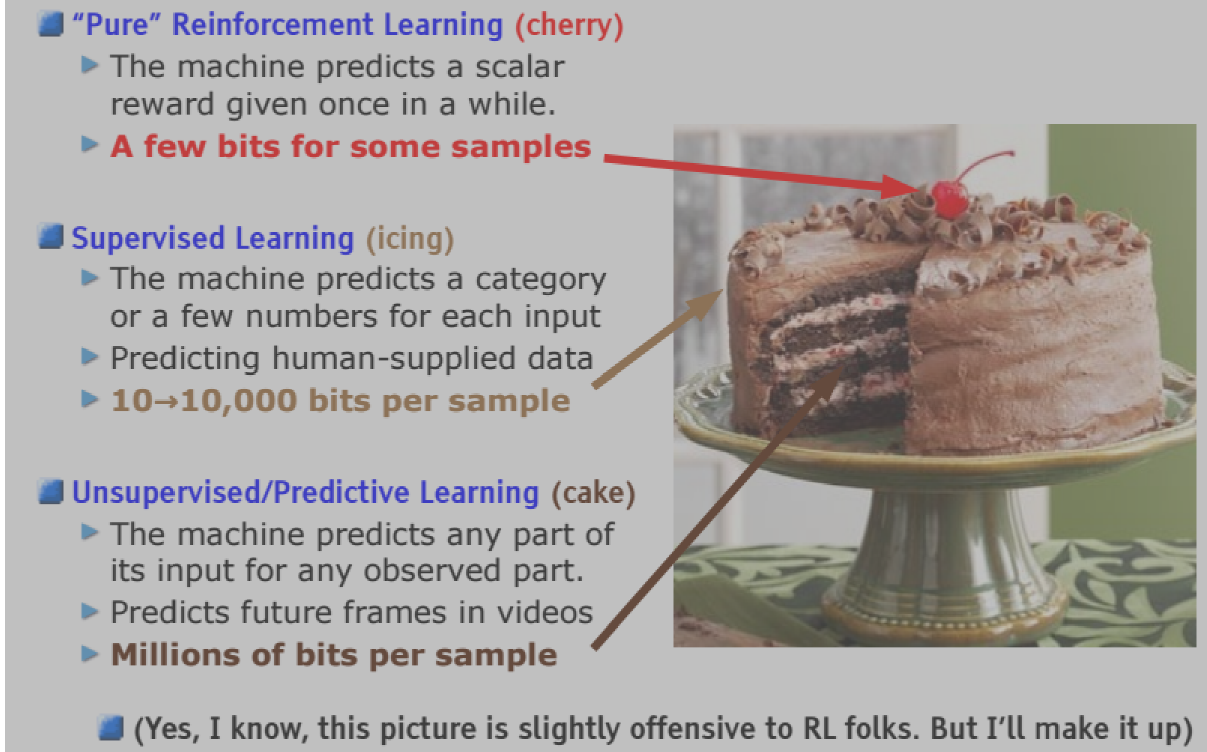
If the network can learn how to draw cat and dog separately,
it must be able to classify them, i.e. feature learning follows naturally.

PREREQUISITES

Taxonomy of Machine Learning



From **David silver**, Reinforcement learning (UCL course on RL, 2015)



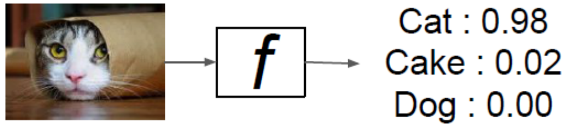
From **Yann Lecun**, (NIPS 2016)

PREREQUISITES

Introduction

Supervised Learning

- More flexible solution
  - Get probability of the label for given data instead of label itself



$$y = f(x)$$

Slide adopted from **Namju Kim**, Kakao brain (SlideShare, AI Forum, 2017)

PREREQUISITES

Introduction

Supervised Learning

- Mathematical notation of **classifying** (greedy policy)
  - y : label, x : data, z : latent, θ^*: fixed optimal parameter

Optimal label prediction

$$y^* = \arg\max_y P(Y \mid X; \theta^*)$$

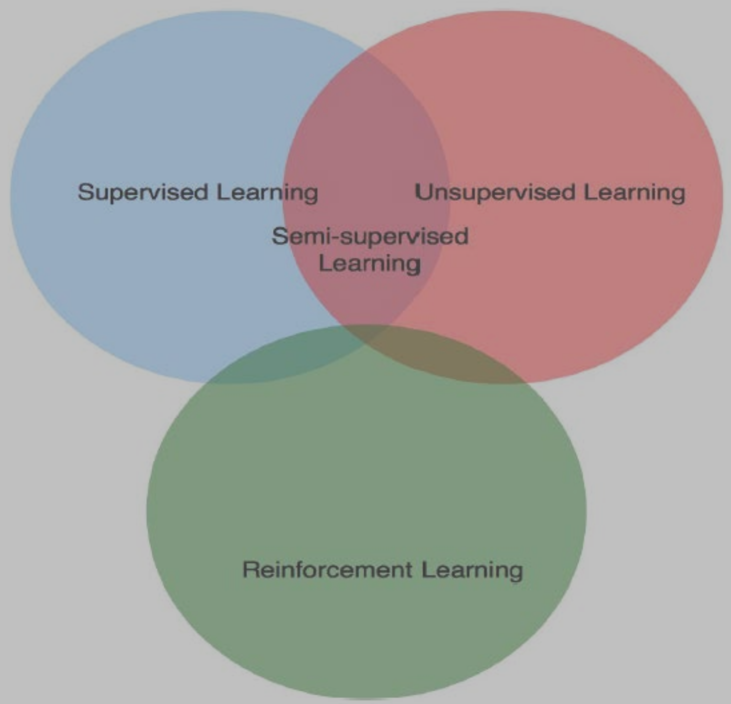
get y when P is maximum　probability　given　parameterized by

kakaobrain

Slide adopted from **Namju Kim**, Kakao brain (SlideShare, AI Forum, 2017)

PREREQUISITES

Taxonomy of Machine Learning



From **David silver**, Reinforcement learning (UCL course on RL, 2015)

- **"Pure" Reinforcement Learning (cherry)**
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- **Supervised Learning (icing)**
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- **Unsupervised/Predictive Learning (cake)**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

- **(Yes, I know, this picture is slightly offensive to RL folks. But I'll make it up)**

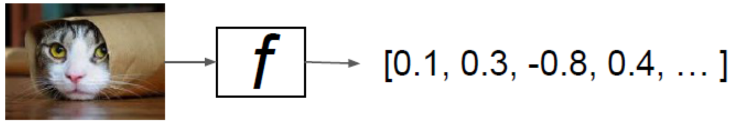
From **Yann Lecun**, (NIPS 2016)

PREREQUISITES

Introduction

Unsupervised Learning

- Find deterministic function ***f*** : $z = f(x)$, x : data, z : **latent**



kakaobrain

Slide adopted from **Namju Kim**, Kakao brain (SlideShare, AI Forum, 2017)

PREREQUISITES

Introduction

Unsupervised Learning

- More challenging than supervised learning :
  - No label or curriculum → self learning
- Some NN solutions :
  - Boltzmann machine
  - Auto-encoder or Variational Inference
  - Generative Adversarial Network

kakaobrain

Slide adopted from **Namju Kim**, Kakao brain (SlideShare, AI Forum, 2017)

PREREQUISITES

Introduction

Unsupervised Learning

- More challenging than supervised learning :
  - No label or curriculum → self learning
- Some NN solutions :
  - Boltzmann machine
  - Auto-encoder or Variational Inference
  - Generative Adversarial Network

kakaobrain

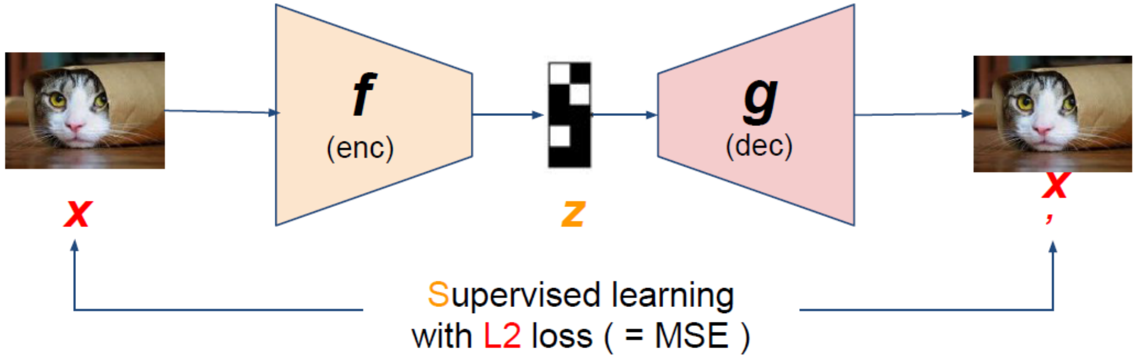
Slide adopted from **Namju Kim**, Kakao brain (SlideShare, AI Forum, 2017)

PREREQUISITES

Autoencoders

Stacked autoencoder - SAE

- Use data itself as label → Convert UL into reconstruction SL
- $z = f(x)$, $x = g(z) \rightarrow x = g(f(x))$
- https://github.com/buriburisuri/sugartensor/blob/master/sugartensor/example/mnist_sae.py



kakaobrain

Slide adopted from **Namju Kim**, Kakao brain (SlideShare, AI Forum, 2017)

PREREQUISITES

Autoencoders

Variational autoencoder - VAE

- Kingma et al, “Auto-Encoding Variational Bayes”, 2013.

- Generative Model + Stacked Autoencoder
  - Based on **Variational approximation**

Variational approximations Variational methods define a lower bound

$$\mathcal{L}(\boldsymbol{x};\boldsymbol{\theta}) \leq \log p_{\text{model}}(\boldsymbol{x};\boldsymbol{\theta}). \quad (7)$$

kakaobrain

Slide adopted from **Namju Kim**, Kakao brain (SlideShare, AI Forum, 2017)

PREREQUISITES

Autoencoders

Variational autoencoder - VAE

- Kingma et al, “Auto-Encoding Variational Bayes”, 2013.

- Generative Model + Stacked Autoencoder
  - Based on **Variational approximation**

Variational approximations Variational methods define a lower bound

$$\widetilde{\mathcal{L}}^B(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = -D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z})) + \frac{1}{L}\sum_{l=1}^{L}(\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}^{(i,l)})) \quad (7)$$

$$\text{where} \quad \mathbf{z}^{(i,l)} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^{(i,l)}, \mathbf{x}^{(i)}) \quad \text{and} \quad \boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon})$$

kaka

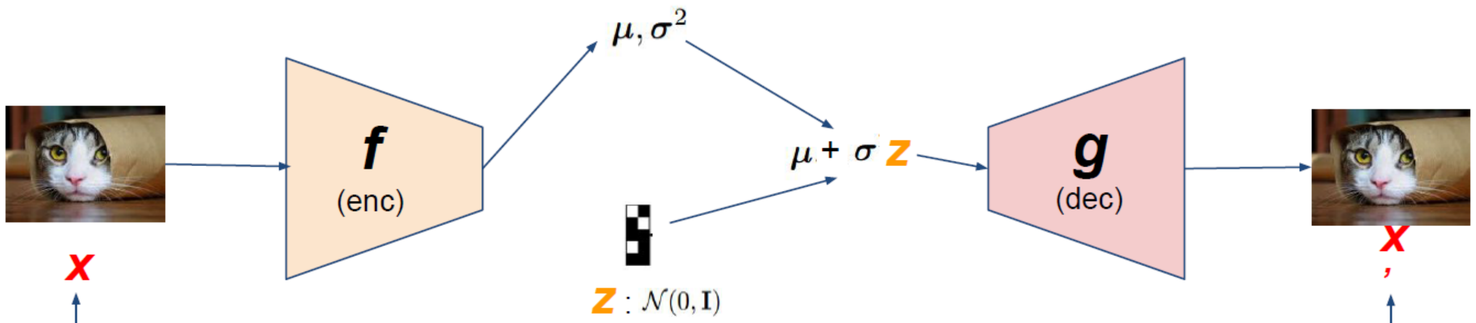
Slide adopted from **Namju Kim**, Kakao brain (SlideShare, AI Forum, 2017)

PREREQUISITES

Autoencoders

Variational autoencoder - VAE

- Training
  - https://github.com/buriburisuri/sugartensor/blob/master/sugartensor/example/mnist_vae.py



$$\widetilde{\mathcal{L}}^B(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = -D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z})) + \frac{1}{L}\sum_{l=1}^{L}(\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}^{(i,l)}))$$

$$\text{where} \quad \mathbf{z}^{(i,l)} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^{(i,l)}, \mathbf{x}^{(i)}) \quad \text{and} \quad \boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon})$$

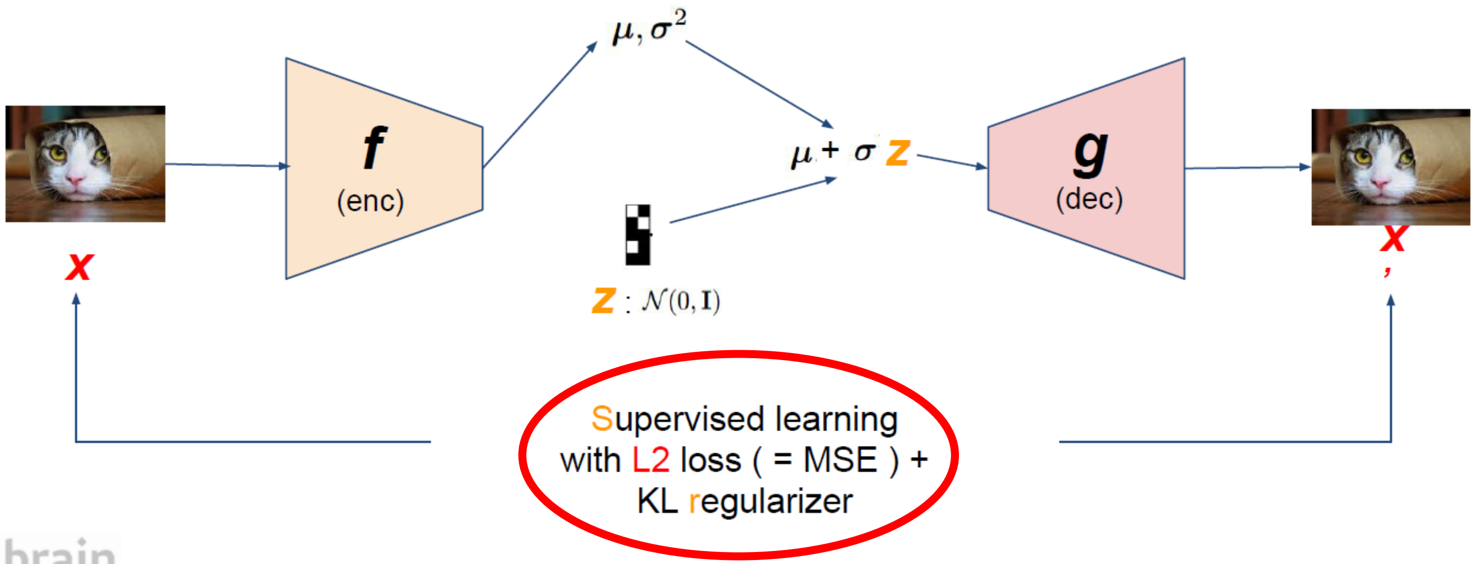
Slide adopted from **Namju Kim**, Kakao brain (SlideShare, AI Forum, 2017)

PREREQUISITES

Autoencoders

Variational autoencoder - VAE

- Training
  - https://github.com/buriburisuri/sugartensor/blob/master/sugartensor/example/mnist_vae.py



Slide adopted from **Namju Kim**, Kakao brain (SlideShare, AI Forum, 2017)

PREREQUISITES

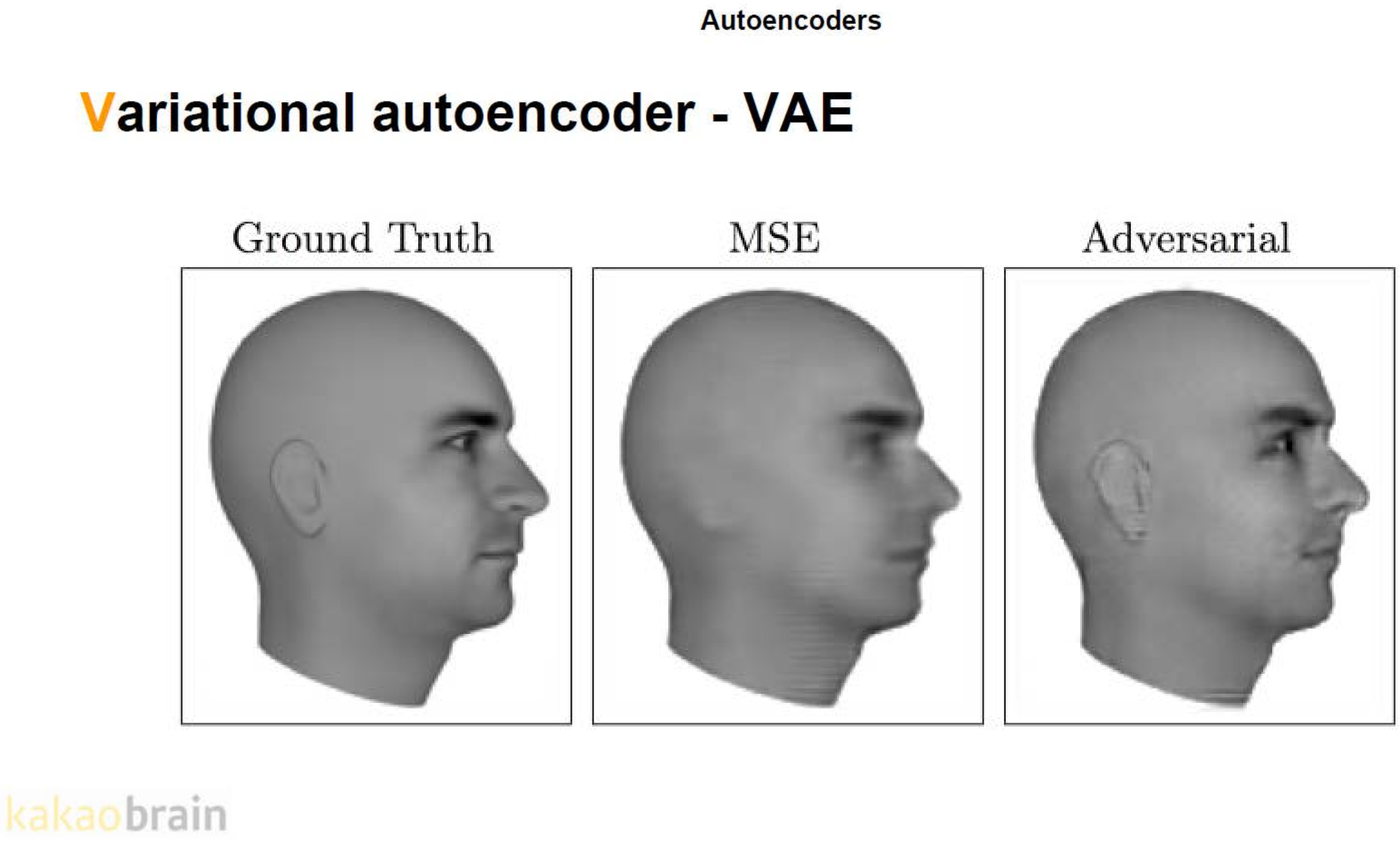
Autoencoders

Variational autoencoder - VAE

- Results

Slide adopted from **Namju Kim**, Kakao brain (SlideShare, AI Forum, 2017)

PREREQUISITES



Slide adopted from **Namju Kim**, Kakao brain (SlideShare, AI Forum, 2017)

* Figure adopted from NIPS 2016 Tutorial: GAN paper, Ian Goodfellow 2016

Generative Adversarial Network

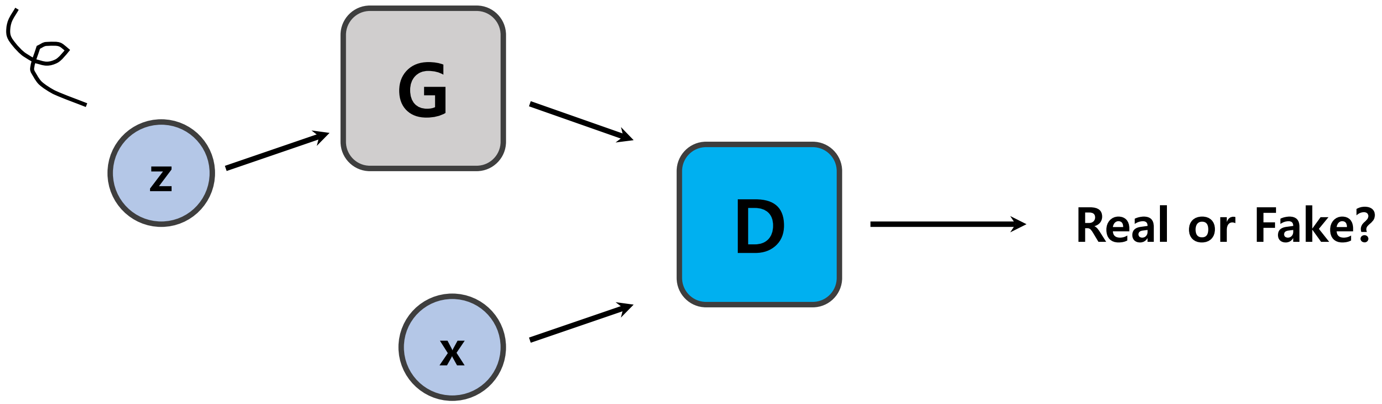
Generative Adversarial Network

SCHEMATIC OVERVIEW

Diagram of Standard GAN

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[logD(x)] + \mathbb{E}_{z\sim p_z(z)}[log(1-D(G(z)))]$$

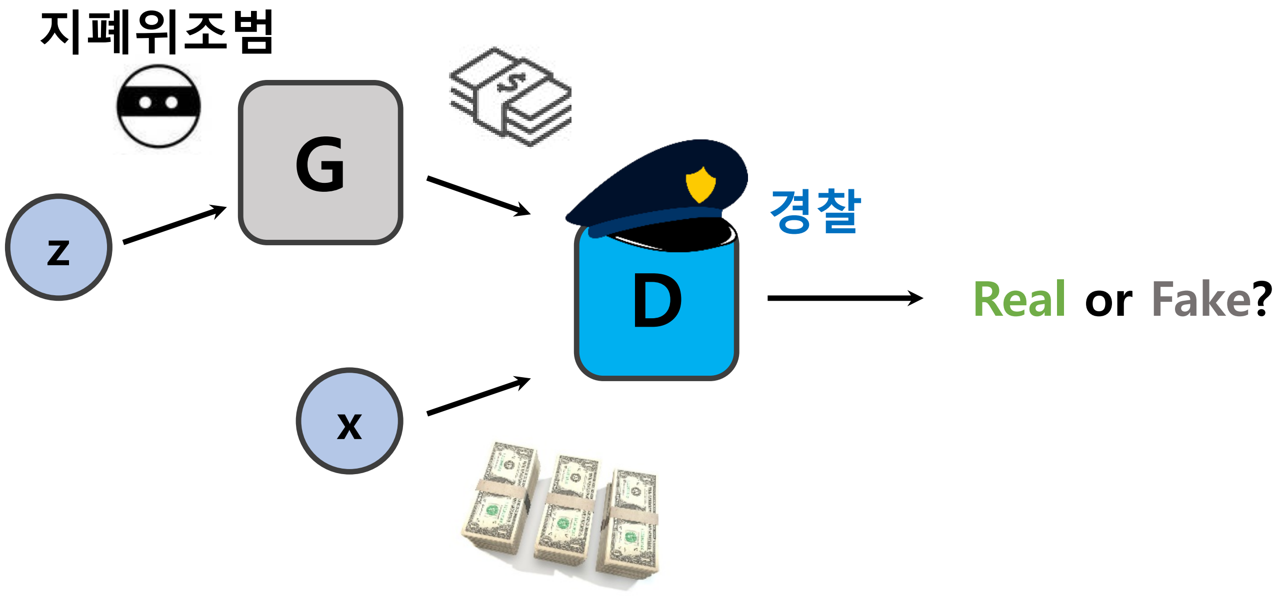
Gaussian noise as an input for G



SCHEMATIC OVERVIEW

Diagram of Standard GAN

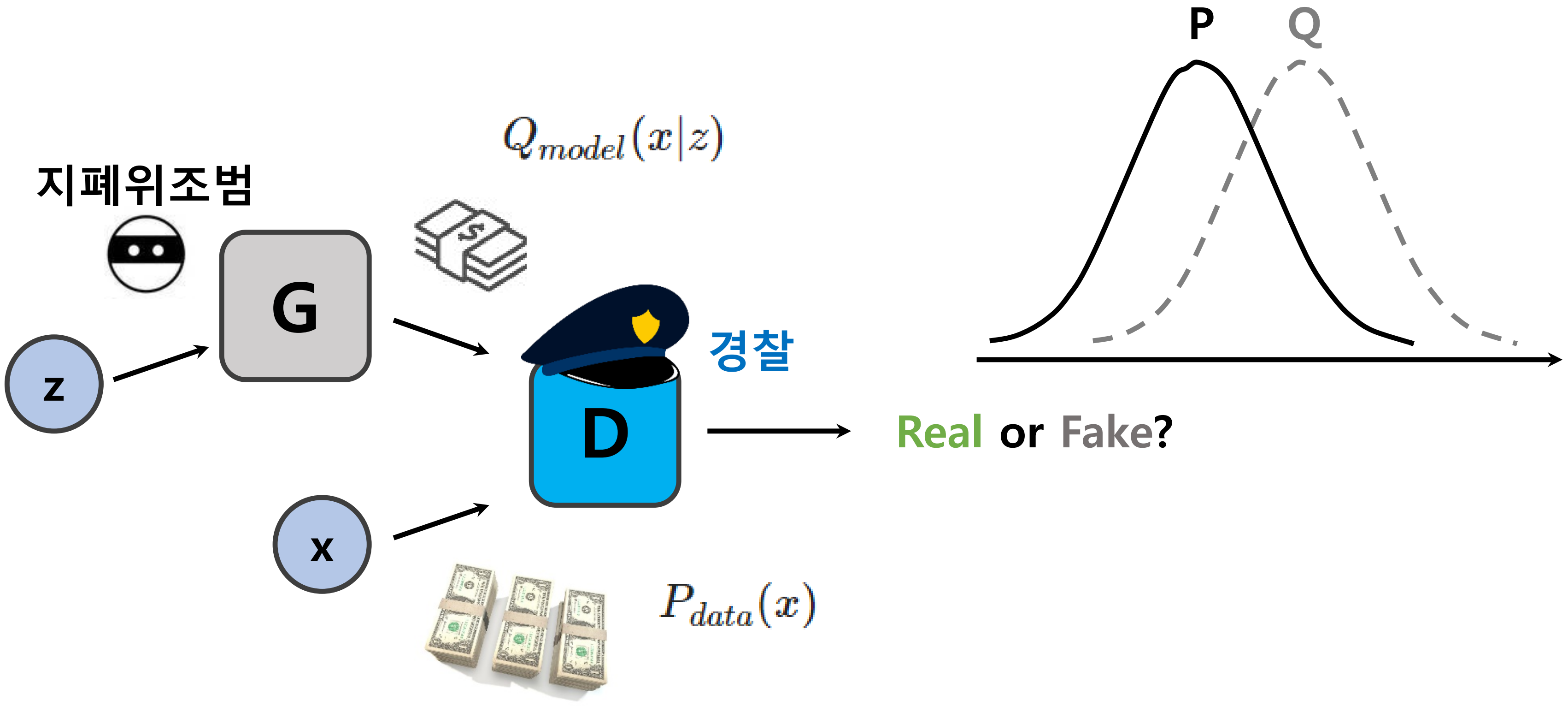
$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[logD(x)] + \mathbb{E}_{z\sim p_z(z)}[log(1-D(G(z)))]$$



SCHEMATIC OVERVIEW

Diagram of Standard GAN

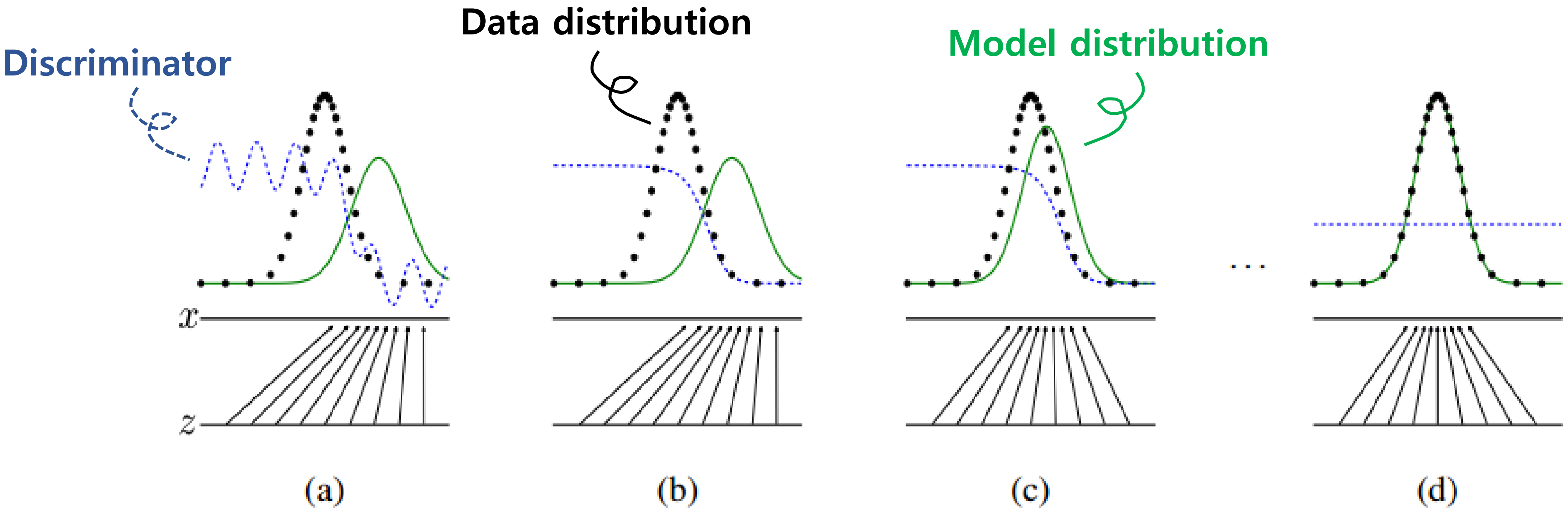
$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[logD(x)] + \mathbb{E}_{z\sim p_z(z)}[log(1-D(G(z)))]$$



SCHEMATIC OVERVIEW

Diagram of Standard GAN

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[logD(x)] + \mathbb{E}_{z\sim p_z(z)}[log(1-D(G(z)))]$$



* Figure adopted from Generative Adversarial Nets, Ian Goodfellow et al. 2014

THEORETICAL RESULTS

Minimax problem of GAN

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}\,(x)}[logD(x)] + \mathbb{E}_{z\sim p_z(z)}[log(1-D(G(z)))]$$

TWO STEP APPROACH

Show that…

1. The minimax problem of GAN has a global optimum at $p_g = p_{data}$
2. The proposed algorithm can find that global optimum

THEORETICAL RESULTS

Proposition 1.

For G fixed, the optimal discriminator D is

$$D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}.$$

$$\begin{aligned}
C(G) &= \max_D V(G, D) \\
&= \mathbb{E}_{x \sim p_{data}}\left[log D_G^*(x)\right] + \mathbb{E}_{z \sim p_z}\left[log(1 - D_G^*(G(z)))\right] \\
&= \mathbb{E}_{x \sim p_{data}}\left[log D_G^*(x)\right] + \mathbb{E}_{x \sim p_g}\left[log(1 - D_G^*(x))\right] \\
&= \mathbb{E}_{x \sim p_{data}}\left[log \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}\right] + \mathbb{E}_{x \sim p_g}\left[log \frac{p_g(x)}{p_{data}(x) + p_g(x)}\right]
\end{aligned}$$

THEORETICAL RESULTS

Proposition 1.

For G fixed, the optimal discriminator D is

$$D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}.$$

Proof. The training criterion for the discriminator D, given any generator G, is to maximize the quantity $V(G, D)$

$$\begin{aligned} V(G,D) &= \int_x p_{data}(x) log(D(x))dx + \int_z p_z(z) log(1 - D(G(z)))dz \\ &= \int_x p_{data}(x) log(D(x)) + p_g(x) log(1 - D(x))dx \end{aligned}$$

For any $(a, b) \in \mathbb{R}^2 \setminus \{0, 0\}$, the function $y \to a log(y) + b log(1 - y)$ achieves its maximum in $[0, 1]$ at $\frac{a}{a+b}$. The discriminator does not need to be defined outside of $Supp(p_{data}) \cup Supp(p_g)$, concluding the proof. ■

THEORETICAL RESULTS

Main Theorem

The global minimum of the virtual training criterion $C(G)$ is achieved if and only if $p_g = p_{data}$. At that point, $C(G)$ achieves the value $-log(4)$.

For $p_g = p_{data}$, $D^*_G(x) = \frac{1}{2}$ and

$$C(G) = \mathbb{E}_{x \sim p_{data}}[-log(2)] + \mathbb{E}_{x \sim p_g}[-log(2)] = -log(4).$$

To show that this is the best possible value of $C(G)$:

$$\begin{aligned} C(G) &= -log(4) + KL\left(p_{data} || \frac{p_{data} + p_g}{2}\right) + KL\left(p_g || \frac{p_{data} + p_g}{2}\right) \\ &= -log(4) + 2 \cdot JSD(p_{data} || p_g). \end{aligned}$$

Here, JSD is always positive value and equal to 0 only if two distributions match. Therefore, $C^* = -log(4)$ is the global minimum of $C(G)$ where the only solution is $p_g = p_{data}$.

THEORETICAL RESULTS

Convergence of the proposed algorithm

If G and D have enough capacity, and at each step of Algorithm 1, the discriminator is allowed to reach its optimum given G, and p_g is updated so as to improve the criterion

$$\mathbb{E}_{x \sim p_{data}} \left[log D_G^*(x) \right] + \mathbb{E}_{x \sim p_g} \left[log(1 - D_G^*(x)) \right]$$

then p_g converges to p_{data}.

Proof. Consider $V(G, D) = U(p_g, D)$ as a function of p_g as done in the above criterion. Note that $U(p_g, D)$ is convex in p_g. The subderivatives of a supremum of convex functions include the derivative of the function at the point where the maximum is attained. This is equivalent to computing a gradient descent update for p_g at the optimal D given the corresponding G, $\sup_D U(p_g, D)$ is convex in p_g with a unique global optima as proven in **Thm 1**, therefore with sufficiently small updates of p_g, p_g converges to p_x, concluding the proof. ■

THEORETICAL RESULTS

Convergence of the proposed algorithm

If G and D have enough capacity, and at each step of Algorithm 1, the discriminator is allowed to reach its optimum given G, and p_g is updated so as to improve the criterion

$$\mathbb{E}_{x \sim p_{data}}\left[log D_G^*(x)\right] + \mathbb{E}_{x \sim p_g}\left[log(1 - D_G^*(x))\right]$$

then p_g converges to p_{data}.

Proof. Consider $V(G, D) = U(p_g, D)$ as a function of p_g as done in the above criterion.

"The subderivatives of a supremum of convex functions include the derivative of the function at the point where the maximum is attained."

equivalent to computing a gradient descent update for p_g at the optimal D given the

If $f(p_g) = \sup_{D \in \mathcal{D}} f_D(p_g)$ and $f_D(p_g)$ is convex in p_g every D, then $\partial f_{D^*}(p_g) \in \partial f$ if $D^* = arg \sup_{D \in \mathcal{D}} f_D(p_g)$.

RESULTS

What can GAN do?

* Figure adopted from DCGAN, Alec Radford et al. 2016 (link)

RESULTS

What can GAN do?

Vector arithmetic
(e.g. word2vec)

$$KING\ (\text{왕}) - MAN\ (\text{남자}) + WOMAN\ (\text{여자})$$

RESULTS

What can GAN do?

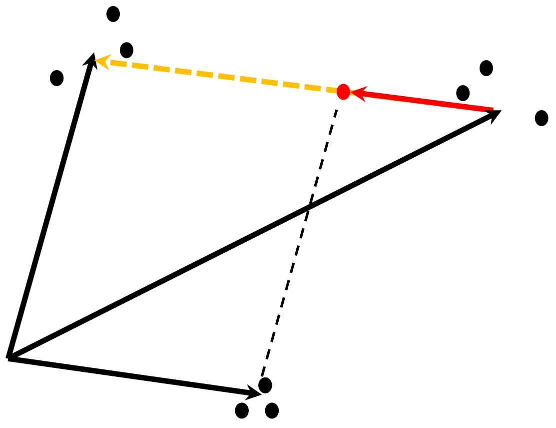
Vector arithmetic
(e.g. word2vec)

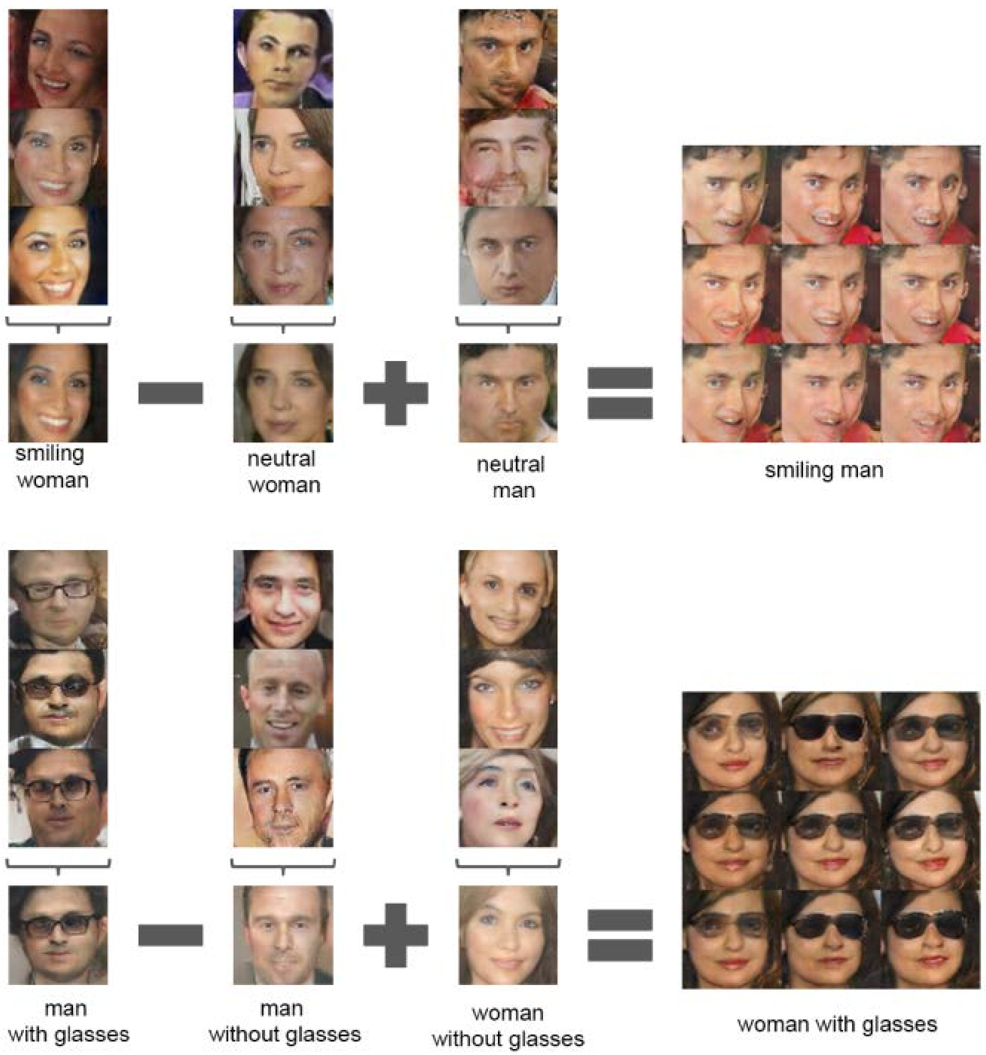
$$QUEEN \text{ (여왕)}$$

RESULTS

What can GAN do?

Vector arithmetic
(e.g. word2vec)





* Figure adopted from DCGAN, Alec Radford et al. 2016 (link)

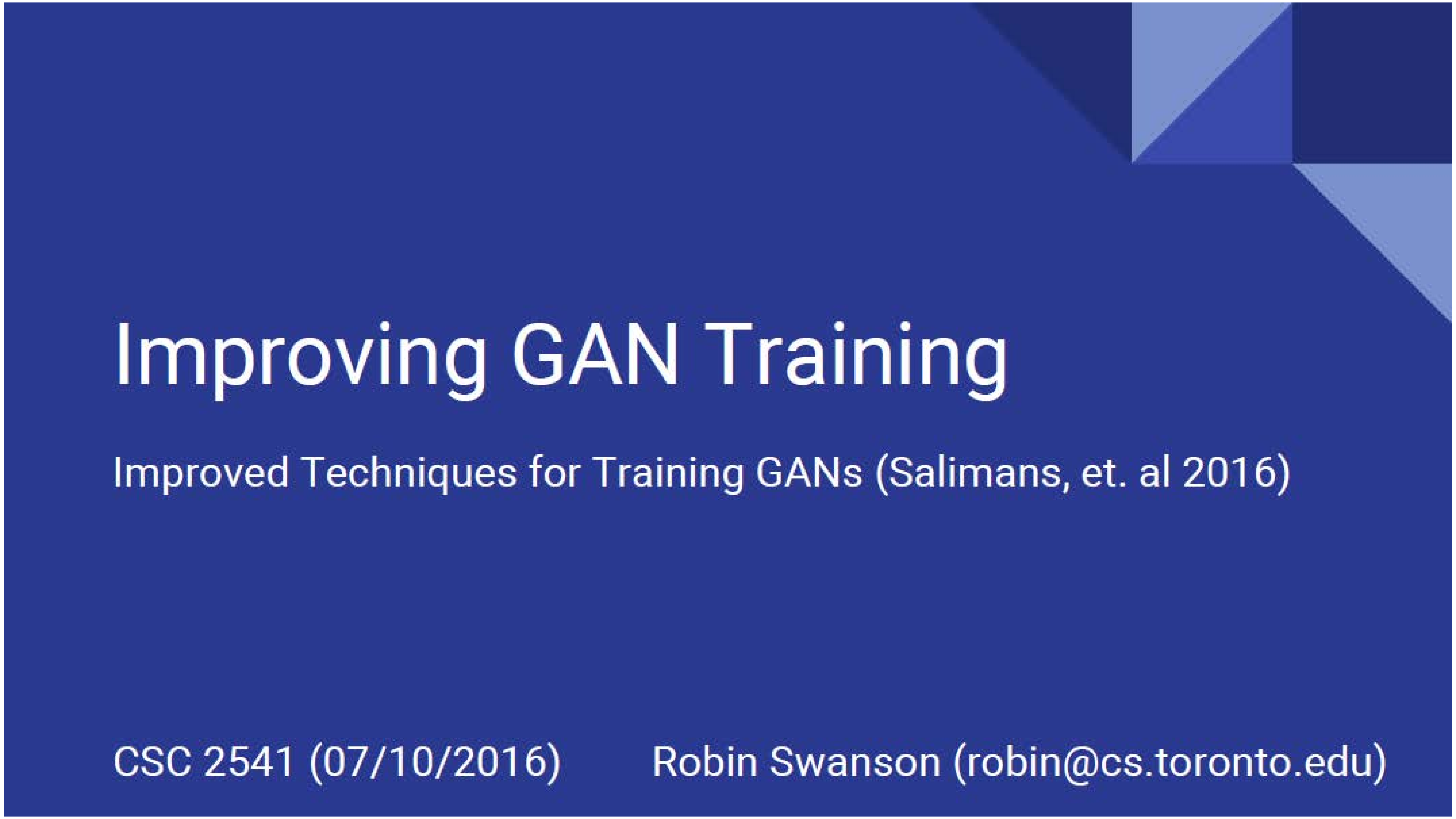
RESULTS

"We want to get a disentangled representation space EXPLICITLY."

Neural network understanding "Rotation"

* Figure adopted from DCGAN, Alec Radford et al. 2016 (link)

DIFFICULTIES



DIFFICULTIES

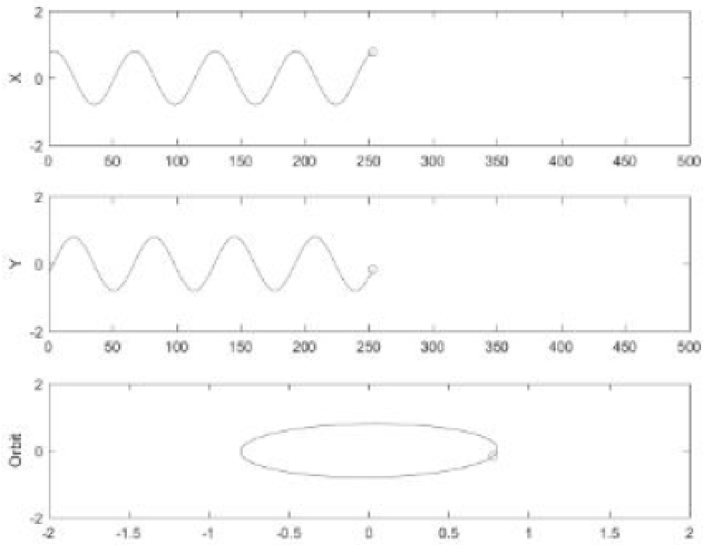
Training GANs is Difficult

- General Case is hard to solve
  - Cost functions are non-convex
  - Parameters are continuous
  - Extreme Dimensionality
- Gradient descent can't solve everything
  - Reducing cost of generator could increase cost of discriminator
  - And vice-versa

DIFFICULTIES CONVERGENCE OF THE MODEL

Simple Example

- Player 1 minimizes f(x) = xy
- Player 2 minimizes f(y) = -xy
- Gradient descent enters a stable orbit
- Never reaches x = y = 0

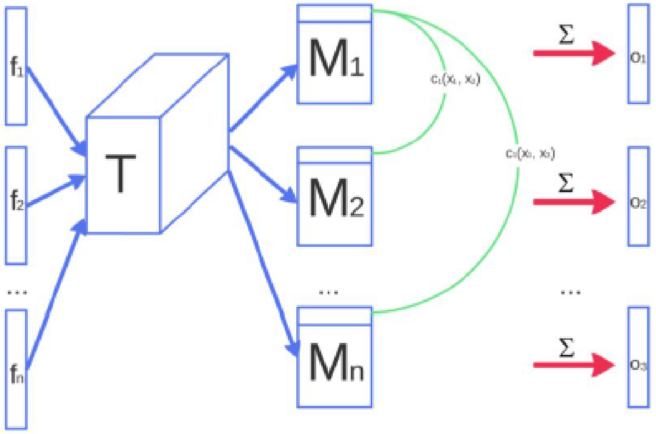


(Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. 2016. MIT Press)

DIFFICULTIES CONVERGENCE OF THE MODEL

Minibatch Discrimination

- Discriminator looks at generated examples independently
- Can't discern generator collapse
- Solution: Use other examples as side information
- KL divergence does not change
- JS favours high entropy

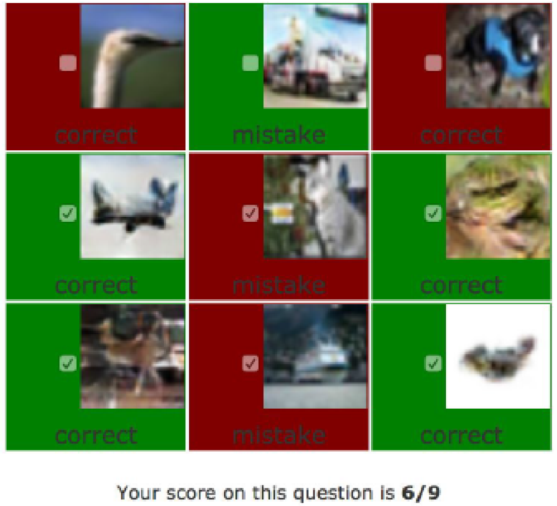


(Ferenc Huszár - http://www.inference.vc/understanding-minibatch-discrimination-in-gans/)

DIFFICULTIES HOW TO EVALUATE THE QUALITY?

Ask Somebody

- Solution: Amazon Mechanical Turk
- Problem:
  - “TASK IS HARD.”
  - Humans are slow, and unreliable, and …
- Annotators learn from mistakes



(http://infinite-chamber-35121.herokuapp.com/cifar-minibatch/)

DIFFICULTIES HOW TO EVALUATE THE QUALITY?

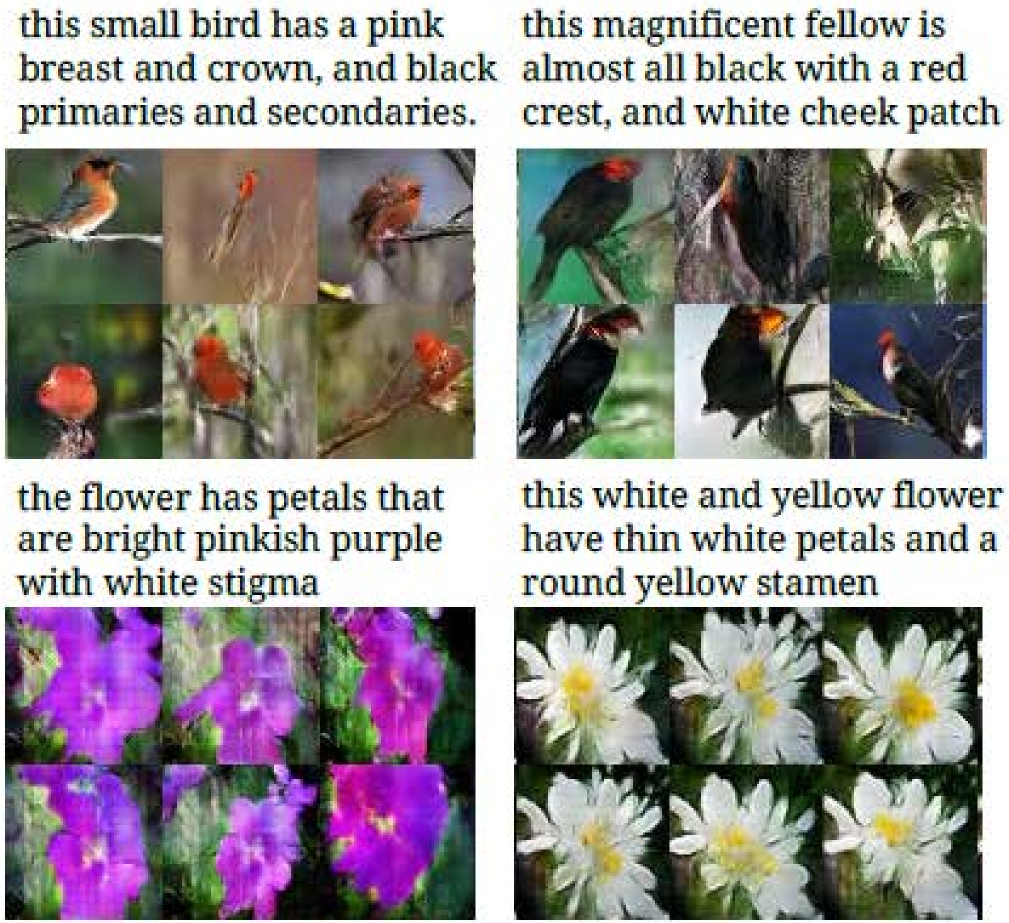
Inception Score

- Run output through Inception Model
- Images with meaningful objects should have a label distribution (p(y|x)) with low entropy
- Set of output images should be varied
- Proposed score:

$$\exp(\mathbb{E}_{\boldsymbol{x}} \mathrm{KL}(p(y|\boldsymbol{x}) || p(y)))$$

- Requires large data sets (>50,000 images)

DIFFICULTIES MODE COLLAPSE (SAMPLE DIVERSITY)

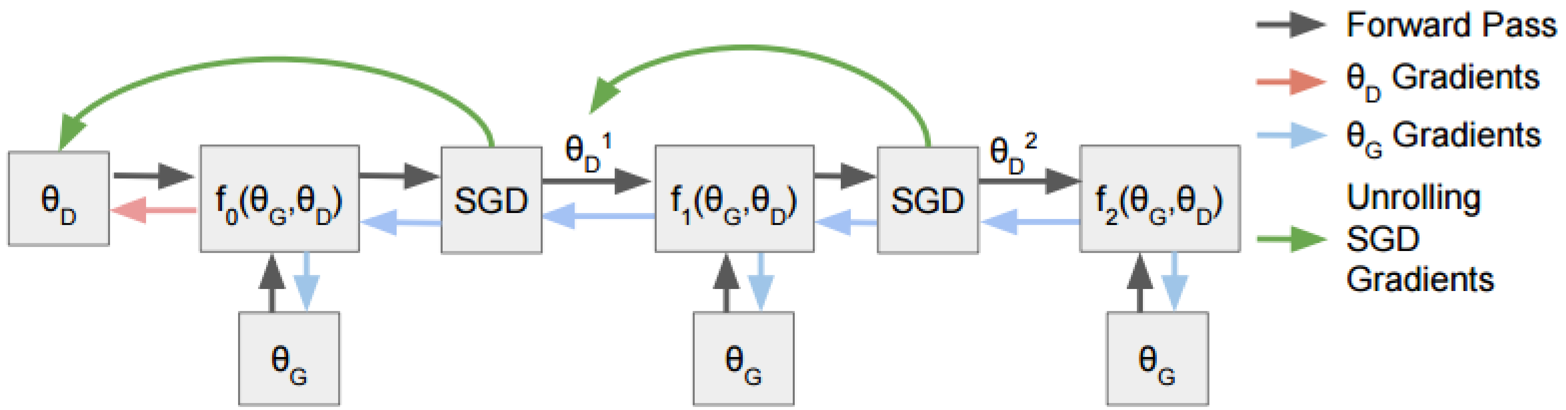


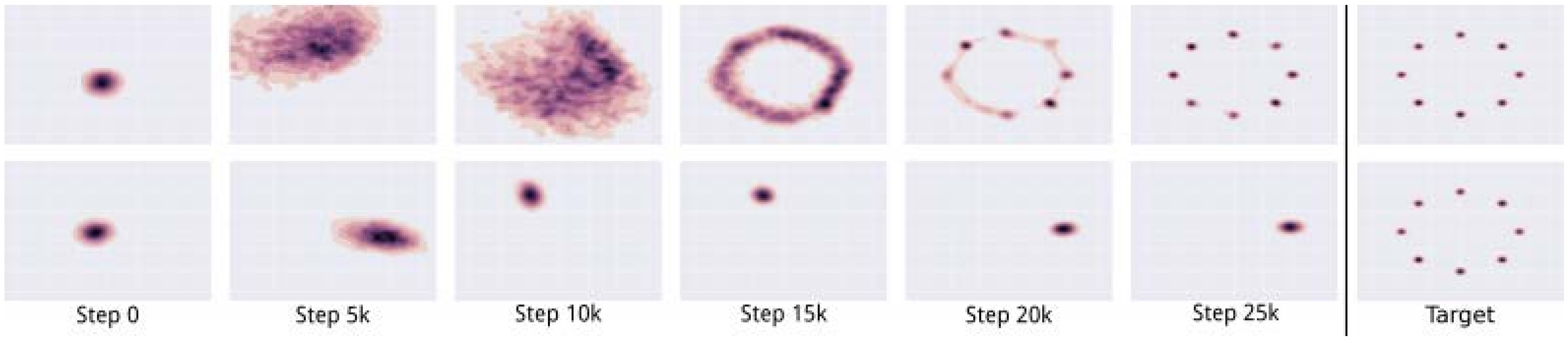
(Reed et al 2016)

(Reed et al, submitted to ICLR 2017)

* Slide adopted from NIPS 2016 Tutorial, Ian Goodfellow

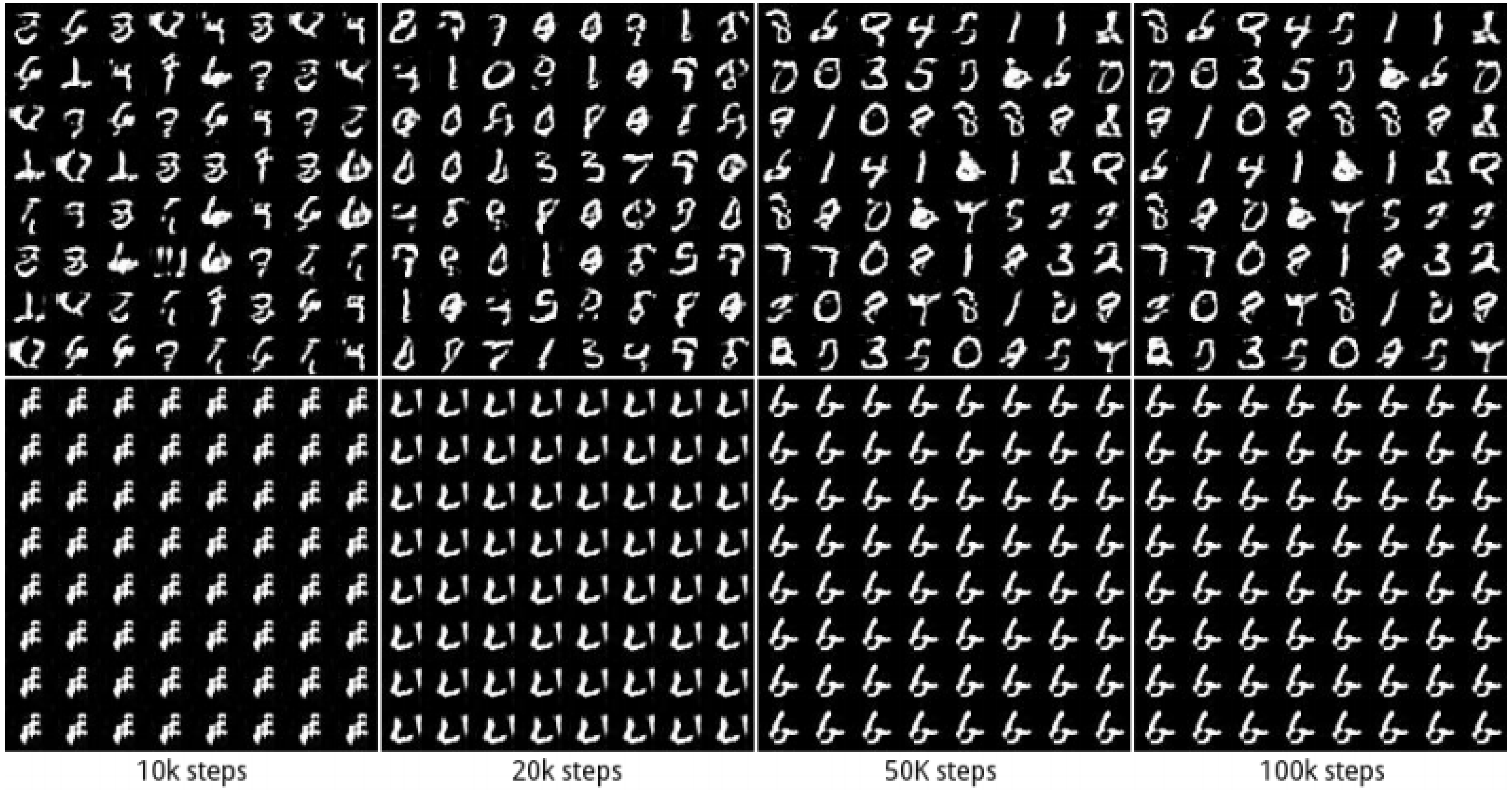
RELATED WORKS





* Unrolled GAN Luke Metz et al. 2016

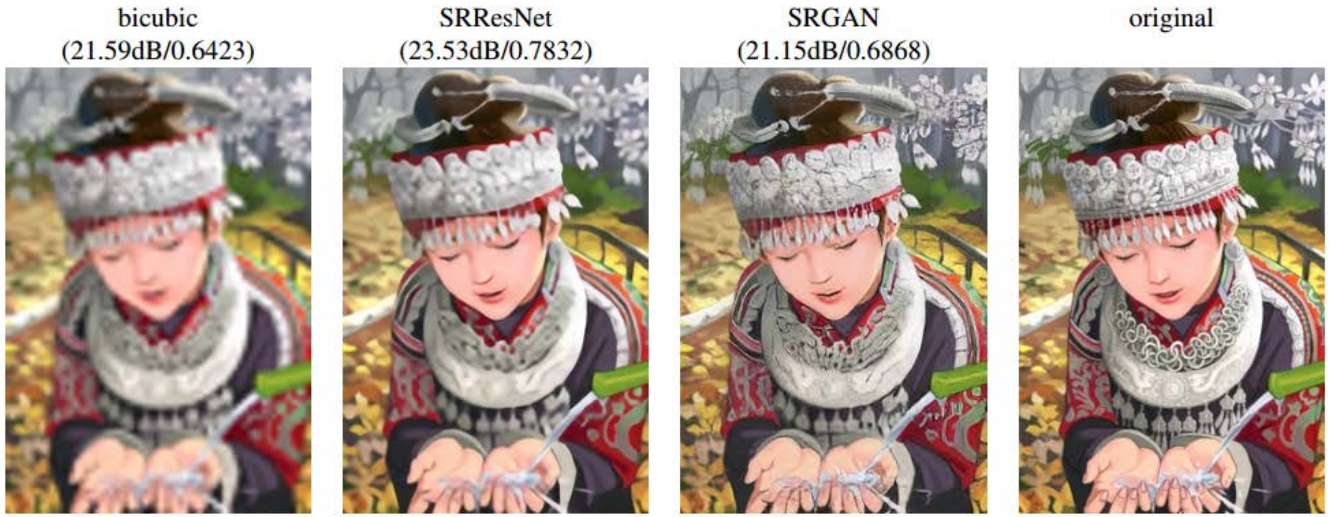
RELATED WORKS



* Unrolled GAN Luke Metz et al. 2016

RELATED WORKS

Super-resolution



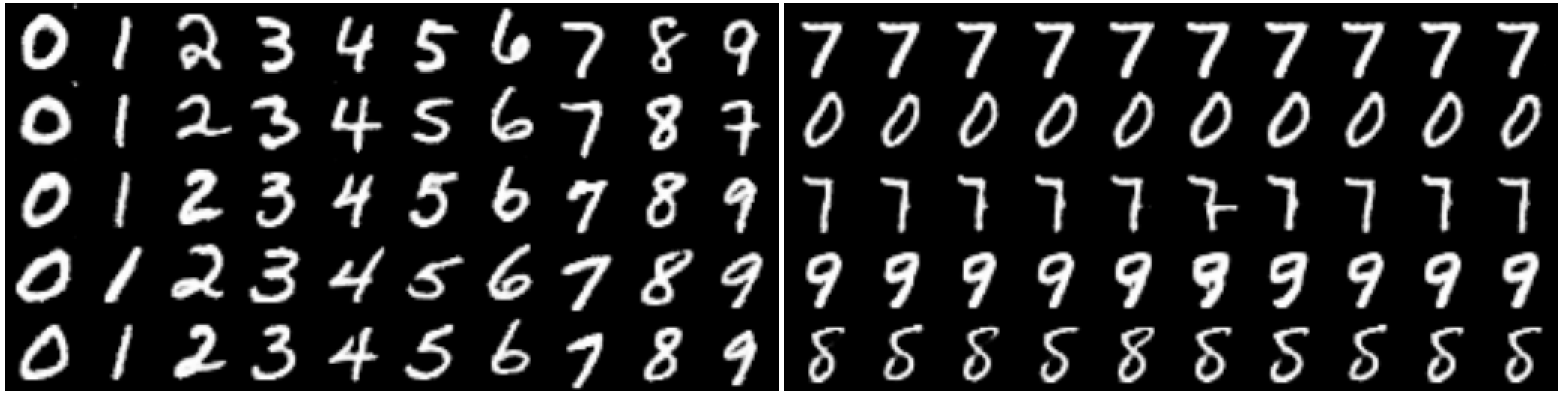
* SRGAN Christian Ledwig et al. 2017

Img2Img Translation

* CycleGAN Jun-Yan Zhu et al. 2017

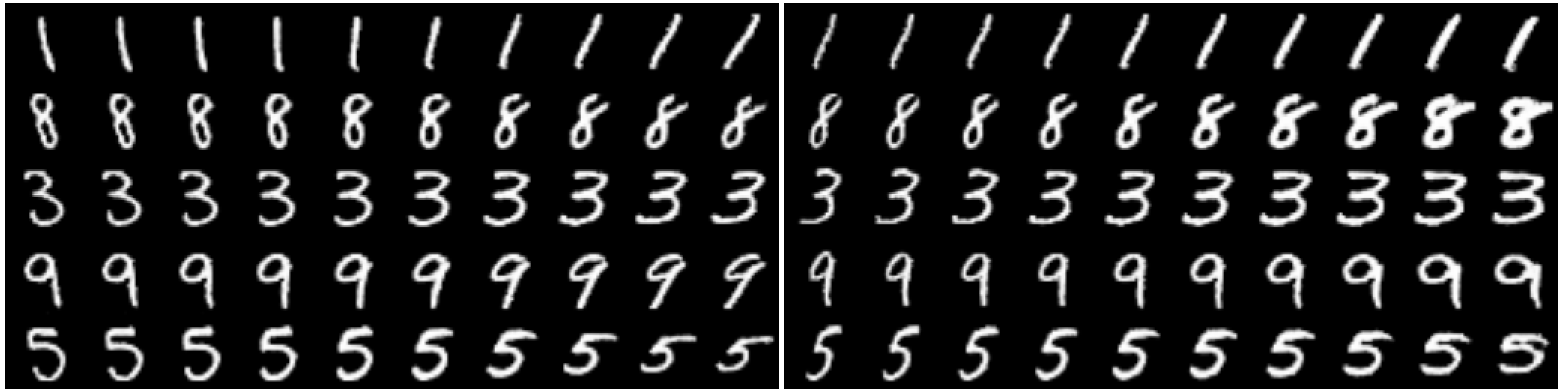
RELATED WORKS

Find a CODE



(a) Varying c_1 on InfoGAN (Digit type)

(b) Varying c_1 on regular GAN (No clear meaning)



(c) Varying c_2 from -2 to 2 on InfoGAN (Rotation)

(d) Varying c_3 from -2 to 2 on InfoGAN (Width)

* infoGAN Xi Chen et al. 2016

RELATED WORKS

Find a CODE

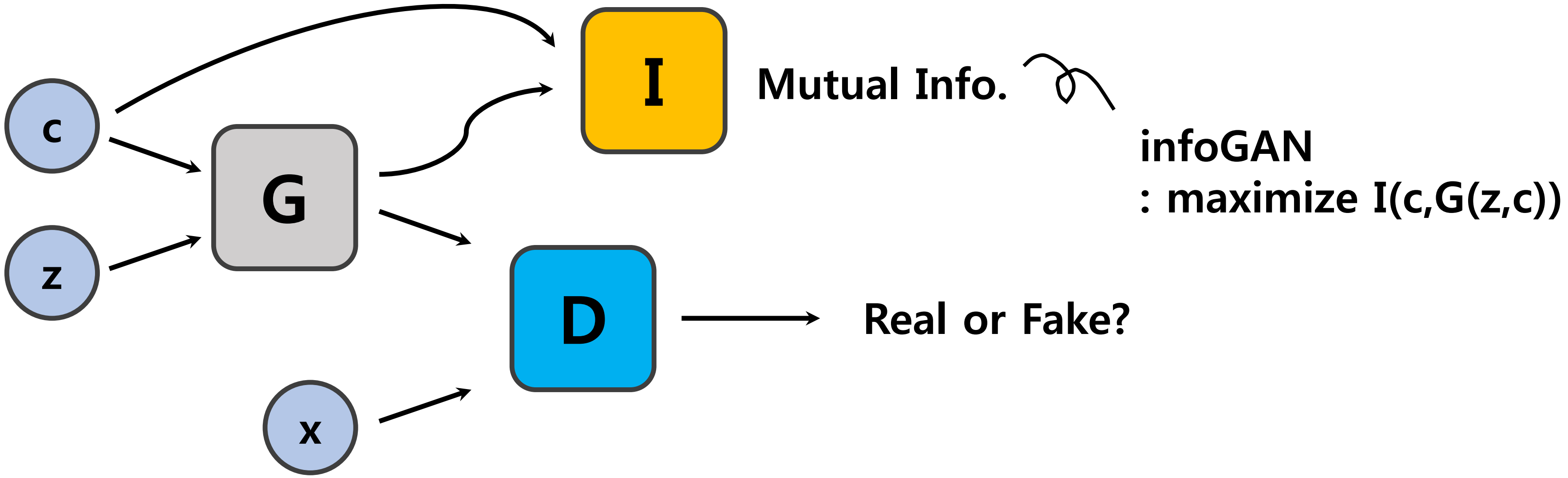
(a) Rotation (b) Width

* infoGAN Xi Chen et al. 2016

RELATED WORKS

Diagram of infoGAN

Impose an extra constraint to learn disentangled feature space



"The information in the latent code c should not be lost in the generation process."

THANK YOU ☺

jaejun.yoo@kaist.ac.kr