Generative Adversarial Networks

(NeurIPS-2014)

0.

SEOULTECH BrAIn Lab. Brain and Artificial Intelligence Lab.

Ian Goodfellow

DeepMind
Verified email at deepmind.com - Homepage
Deep Learning

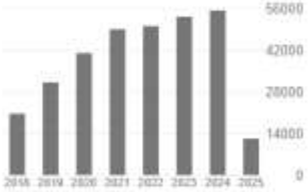
FOLLOW

GET MY OWN PROFILE

TITLE	CITED BY	YEAR
Generative adversarial networks I Goodfellow, J Pouget-Abadie, M Mirza, B Xu, D Warde-Farley, S Ozair, ... Advances in neural information processing systems 27	93619 *	2014
Deep learning I Goodfellow, Y Bengio, A Courville, Y Bengio MIT press 1 (2)	77348	2016
Explaining and Harnessing Adversarial Examples I Goodfellow, J Shlens, C Szegedy ICLR	24697	2014
Intriguing properties of neural networks C Szegedy, W Zaremba, I Sutskever, J Bruna, D Erhan, I Goodfellow, ... arXiv preprint arXiv:1312.6199	18955	2013
Tensorflow: Large-scale machine learning on heterogeneous distributed systems M Abadi, A Agarwal, P Barham, E Brevdo, Z Chen, C Citro, GS Corrado, ... arXiv preprint arXiv:1603.04467	12761	2016
Improved techniques for training gans T Salimans, I Goodfellow, W Zaremba, V Cheung, A Radford, X Chen Advances in neural information processing systems 29	11804	2016
Deep learning with differential privacy M Abadi, A Chu, I Goodfellow, HB McMahan, I Mironov, K Talwar, L Zhang Proceedings of the 2016 ACM SIGSAC conference on computer and communications ...	7656	2016
Adversarial examples in the physical world A Kurakin, IJ Goodfellow, S Bengio Artificial intelligence safety and security, 99-112	7243	2018
Self-attention generative adversarial networks H Zhang, I Goodfellow, D Metaxas, A Odena International conference on machine learning, 7354-7363	5178	2019
Practical black-box attacks against machine learning N Papernot, P McDaniel, I Goodfellow, S Jha, ZB Celik, A Swami Proceedings of the 2017 ACM on Asia conference on computer and ...	4542	2017
Mixmatch: A holistic approach to semi-supervised learning D Berthelot, N Carlini, I Goodfellow, N Papernot, A Oliver, CA Raffel Advances in neural information processing systems 32	4029	2019
Adversarial machine learning at scale A Kurakin, I Goodfellow, S Bengio arXiv preprint arXiv:1611.01236	3955	2016
Adversarial autoencoders A Makhzani, J Shlens, N Jaitly, I Goodfellow, B Frey arXiv preprint arXiv:1511.05644	3545	2015
Ensemble adversarial training: Attacks and defenses F Tramèr, A Kurakin, N Papernot, I Goodfellow, D Boneh, P McDaniel arXiv preprint arXiv:1705.07204	3501	2017

Cited by VIEW ALL

	All	Since 2020
Citations	330385	261670
h-index	96	90
i10-index	169	161



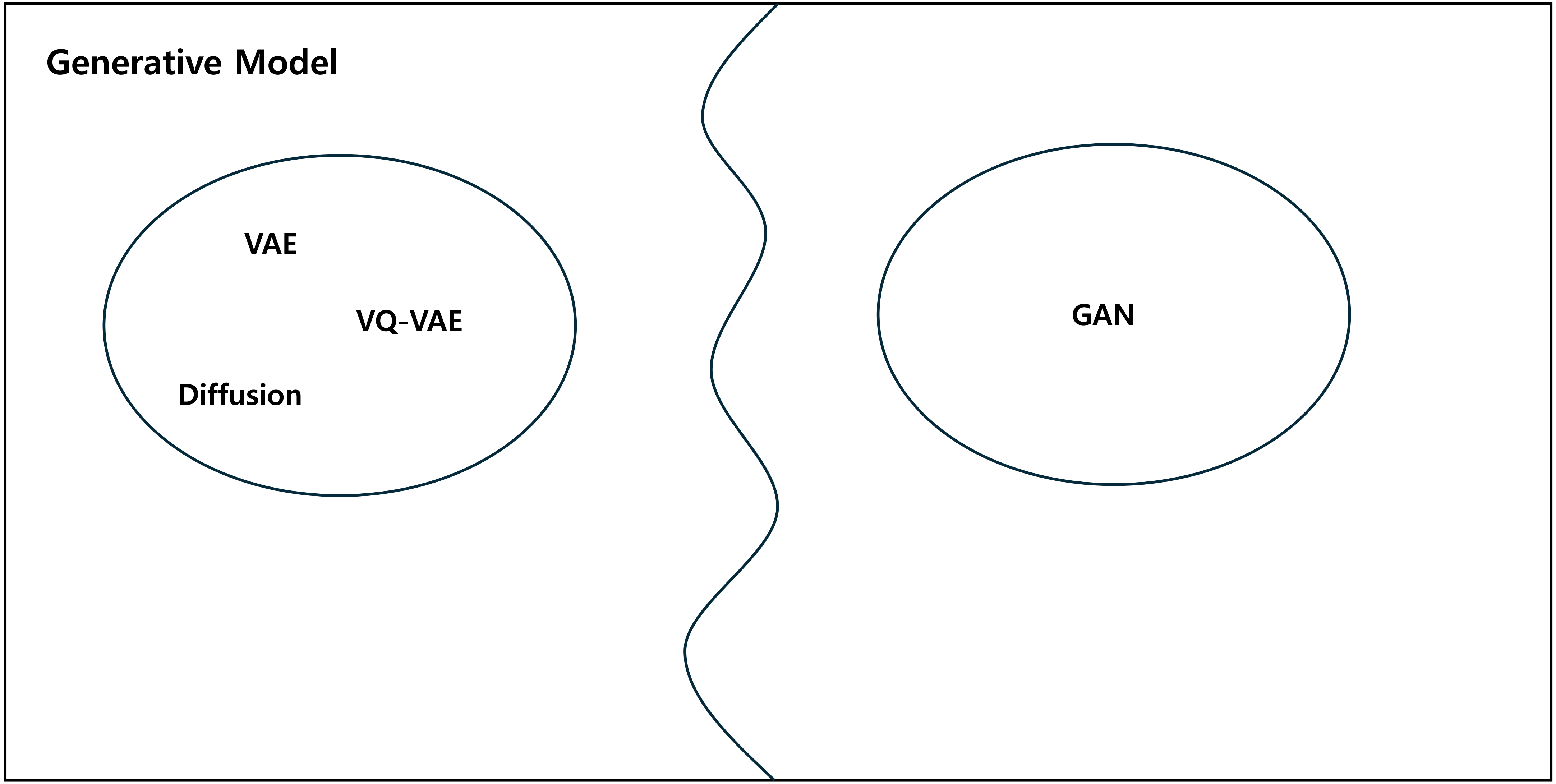
Public access VIEW ALL

1 article — 4 articles
not available — available
Based on funding mandates

Co-authors VIEW ALL

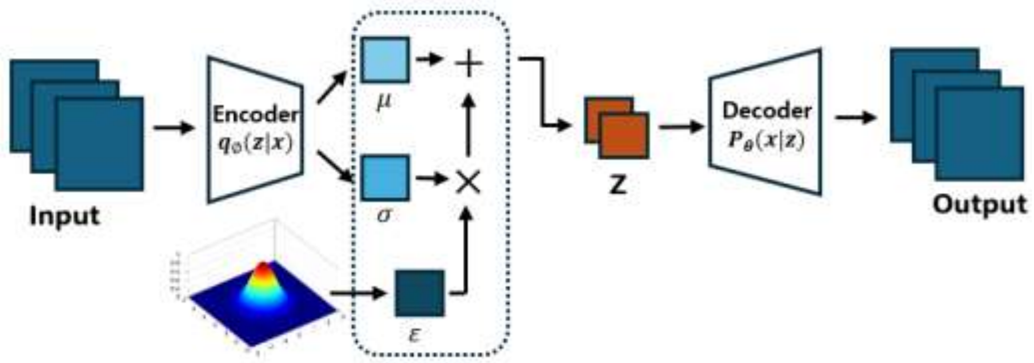
- Yoshua Bengio — Professor of computer science, ...
- Aaron Courville — Professor, DIRO, Université de ...
- David Warde-Farley — Staff Research Scientist at Goog...
- Mehdi Mirza — DeepMind
- Bing Xu — Distinguished Engineer, NVIDIA
- Jean Pouget-Abadie — Google Research
- Nicolas Papernot — University of Toronto and Vector ...
- Alexey Kurakin — Research Software Engineer, Go...
- Wojciech Zaremba — Co-Founder of OpenAI

1. GAN



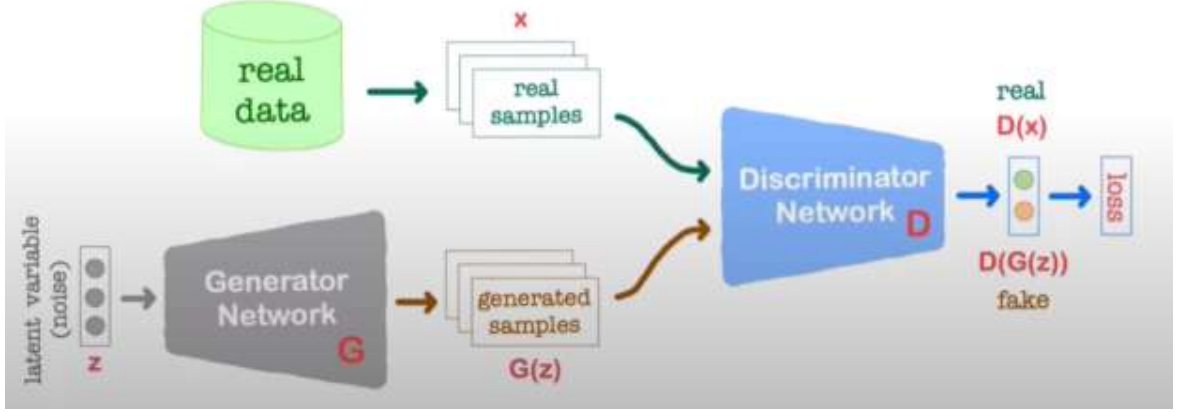
1. GAN

VAE



- Encoder와 Decoder의 구조
- Train data의 likelihood를 최대화(MLE) 하는 방향으로 학습이 진행된다.
- Train data와 유사하게 생성하도록 학습

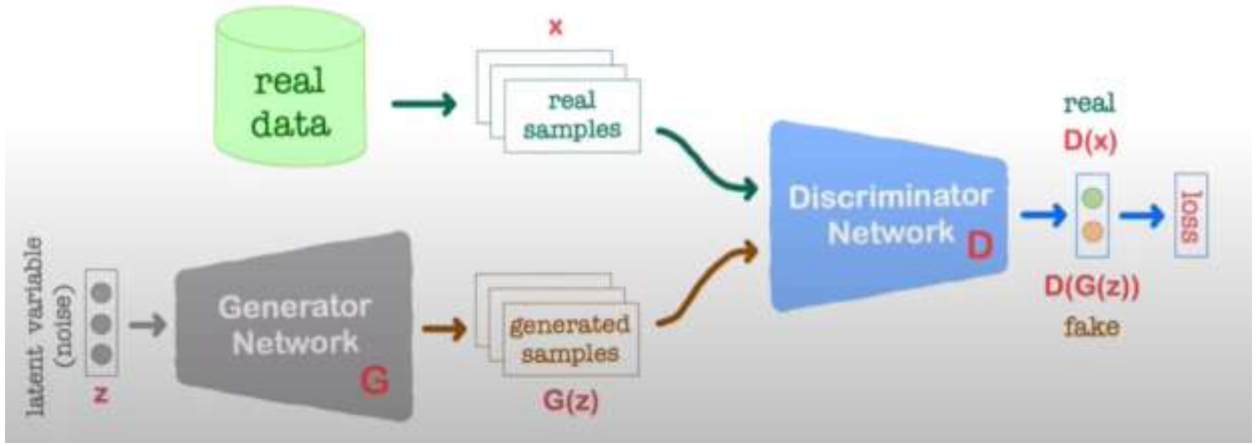
GAN



- Generator와 Discriminator 구조
- 두 Network가 적대적으로 학습
- Min/Max 게임을 이용해서 학습이 진행
- 분포 매칭 관점에서 학습이 진행

1. GAN

GAN



- Discriminator : Generator 가 생성한 데이터의 True/False 를 판단
- Generator : 임의의 noise로부터 Discriminator를 속이는 데이터를 생성
- G & D 는 순차적으로 학습 (Discriminator -> Generator)
- G & D 의 학습은 한쪽이 학습되면 한쪽은 Frezze (gradient가 전파되지 않는다)
- Inference 단계에서는 Generator만 사용된다.

SEOULTECH BrAIn Lab. Brain and Artificial Intelligence Lab.

Objective Function

Objective Function ?

➔ GAN 은 일반적인 Network와 다르게 최적화 문제에서 Min / Max game을 최적화 하는게 목적(objective)

➔ 일반적으로 Loss function은 단일 모델을 최적화 하기위해 사용되는 함수

$$Objective\ Function = \min_G \max_D L(D,G) = \underbrace{\overset{1)}{E}_{x \sim \underbrace{P_{data}(x)}_{\text{Real-data}}}}_{\text{Batch 샘플}}[\log(D)] + \overset{2)}{E}_{z \sim \underbrace{P_z(z)}_{\text{Latent variable}}}[\log(1 - D(G(z)))]$$

Normal / Uniform

Optimize objective function

➔ Discriminator는 1) 과 2) 가 전부 커지도록 학습을 진행한다 (real data -> 1 , fake data ->0)

➔ Generator 는 1)은 사용을 하지 않고 2)가 점점 작아지도록 학습을 진행한다.

SEOULTECH BrAIn Lab. Brain and Artificial Intelligence Lab.

Optimize objective function

$$Objective\ Function = \min_G \max_D L(D,G) = E_{x\sim P_{data}(x)}[\log(D)] + E_{z\sim P_z(z)}[\log(1-D(G(z))]$$

1) Fiexed generator G, maximal discriminator D

$$L(D,G) = \int_x p_{data}(x)\log(D(x))\,dx + \int_z p_z(z)\log\left(1-D(G(z))\right)dz = \int_x p_{data}(x)\log(D(x)) + p_g(x)\log(D(x))\,dx$$

$$\rightarrow \boldsymbol{a = p_{data}(x), b = p_{g(x)}, \log(D(x)) = y}$$

$\boldsymbol{a\log(y) + b\log(1-y)}$의 **maximum 구하기**

$$\rightarrow \frac{\boldsymbol{a}}{\boldsymbol{y}} + \frac{\boldsymbol{-b}}{\boldsymbol{1-y}} = \boldsymbol{0},\quad \boldsymbol{y} = \frac{\boldsymbol{b}}{\boldsymbol{a+b}}$$

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

2. Objective Function

Optimize objective function

$$Objective\ Function = \min_G \max_D L(D,G) = E_{x\sim P_{data}(x)}[\log(D)] + E_{z\sim P_z(z)}[\log(1-D(G(z))]$$

2) Fiexed discriminator D, Minimizing generator G

$$L(D^*,G) = E_{x\sim p_{data}}[\log D_G^*(x)] + E_{x\sim p_g}[\log(1-D_g^*(x))]$$

$$= E_{x\sim p_{data}}\left[log\frac{p_{data}(x)}{P_{data}(x)+p_g(x)}\right] + E_{x\sim p_g}[\log\left(1-\frac{p_{data}}{p_{data}+p_g(x)}\right)]$$

$$= E_{x\sim p_{data}}\left[log\frac{p_{data}(x)}{P_{data}(x)+p_g(x)}\right] + E_{x\sim p_g}[\log\frac{p_g(x)}{p_{data}(x)+p_g(x)}]$$

$$= D_{KL}(p_{data}\ ||\frac{p_{data}+p_g}{2}) + D_{KL}(p_g||\frac{p_{data}+p_g}{2}) - 2log2$$

$$= D_{KL}(p_{data}\ ||\frac{p_{data}+p_g}{2}) + D_{KL}(p_g||\frac{p_{data}+p_g}{2}) - 2log2$$

$$\boxed{= 2 * D_{JS}(p_{data}, p_g) - \log 4}$$

**

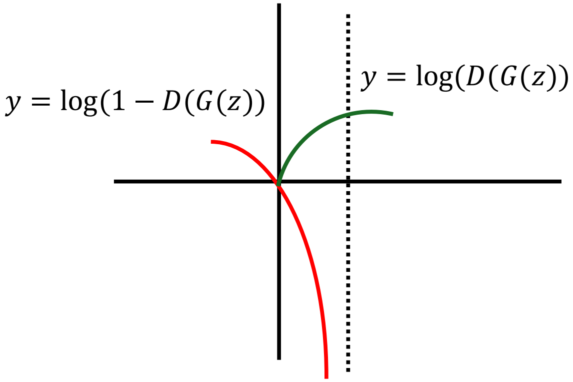
$$D_{KL}(P||Q) = E_{x\sim p}\left[\log\frac{p(x)}{q(x)}\right]$$

$$D_{JS}(P,Q) = \frac{1}{2}D_{KL}(P\ ||\frac{P+Q}{2}) + \frac{1}{2}D_{KL}(Q\ ||\frac{P+Q}{2})$$

2. Objective Function

Objective Function flip trick

$$Objective\ Function = \min_{G} \max_{D} L(D,G) = E_{x \sim P_{data}(x)}[\log(D)] + E_{z \sim P_z(z)}[\log(1 - D(G(z)))]$$



Gradient 기울기를 이용한 학습 트릭

$$\min_{G} L(G) = E_{z \sim p_z}[\log(1 - D(G(z))]$$

$$\max_{g} L(G) = E_{z \sim p_z}[\log(D(G(z))]$$

학습 초반 안정성/속도를 위하여 위 식으로 대체한다

2-1. Algorithm

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

  for k steps **do**

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
  - Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

  end for

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Update the generator by descending its stochastic gradient:

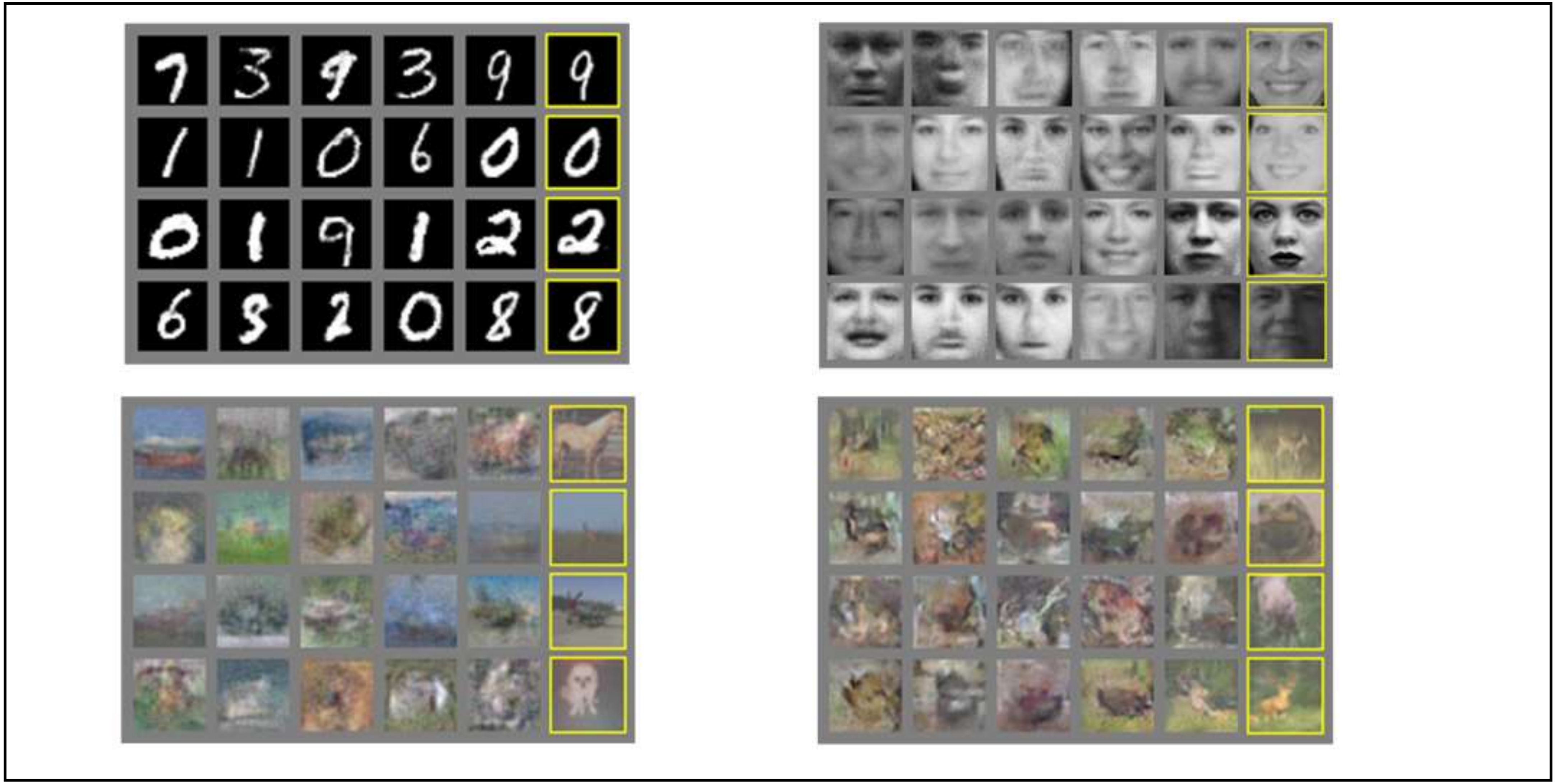
$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

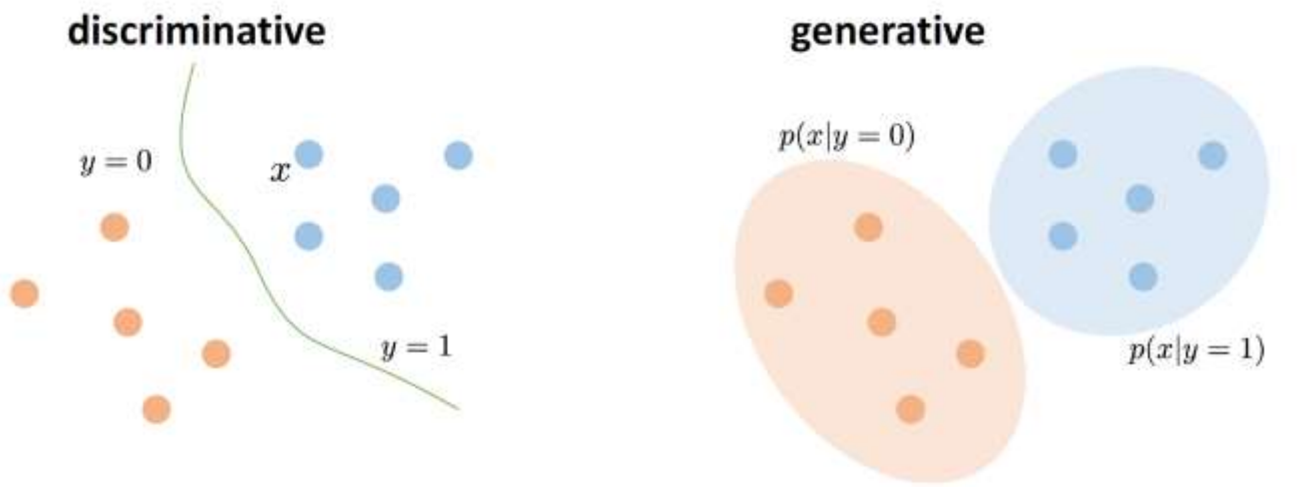
The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

1. **Noise z sampling (minibatch)**
2. **Data x sampling(minibatch)**
3. **Discriminator updata**
4. **Nosie z sampling (minibatch)**
5. **Generator updata**

4. Result



5. Drawback



- 데이터를 1:1로 직접 비교 X
- Fake / real 분포를 학습
- Mode collapse 발생