



[2014] GAN: Generative Adversarial Nets



`ctrl+alt+t` 를 누르면 한 번에 여닫기를 할 수 있습니다.

개인적으로 정리하고 있던 자료와 합치다 보니, 다소 내용이 많은 점 참고 부탁 드립니다.
그리고 화이트 모드로 보는 것을 권장드립니다.

1. 생성 모델을 처음 접함에 있어 필요한 확률, 통계적인 개념을 우선적으로 작성하였습니다.
2. 이후 논문에 대해 자세하게 살펴봅니다.
3. 추가적으로 평가 지표에 대해서 다룹니다.
→ 이미지 생성 모델의 평가 지표에 대해 다른 논문인데, 의견이 흥미로워서 읽어보시는 것을 추천 드립니다.



< Contents >

1. Probabilistic Perspective

들어가며

Basic Statistics

Maximum

Likelihood

Estimation(MLE)

Appendix:

Maximum A

Posterior(MAP)

Kullback-Leibler

Divergence(KLD)

Jenson-

Shannon

Divergence(JSD)

Information &

Entropy

Appendix: MSE

with Probabilic

Perspective —

Regression

Wrap-up

2. Paper

1. Introduction

2. Related Work

3. Adverarial nets

4. Theoretical

Results

5. Experiments

6. Adantages and
disadvantages

7. Conclusion

and future work

3. 참고자료

4. GAN Evaluation

Metrics

5. 추가 사항

1. Probabilistic Perspective

▼ 들어가며

▼ our objective is

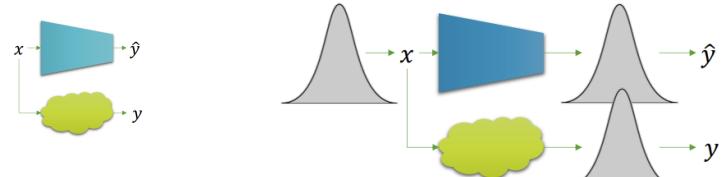
- 가상의 함수를 모사하여, 원하는 출력 값을 반환하는 신경망의 파라미터를 찾자
- 그래서 우리는 DNN을 얘기할 때,
 - Gradient Descent
 - Back-propagation
 - Feature Vector
 - etc
- 이제는 사고의 확장이 필요하다

▼ 이 세상은 확률에 기반한다

- 우리의 새로운 목표: 확률 분포를 학습하는 것

▼ Before(Func) vs After(Probability)

- | | |
|---------------------------------|--|
| • Before: 함수를 배우
자 | • After: 확률 분포 함수를 배우자 |
| ◦ Deterministic
target 값을 예측 | <ul style="list-style-type: none">◦ 수학적으로 좀 더 설명 가능함◦ 불확실성까지 학습 |



▼ Summary

- Neural Networks는 확률 분포 함수를 모델링 할 수 있음
 - 이를 통해 가상의 확률 분포 함수 $P(y|x)$ 를 근사(approximation)할 것
- 대부분의 최신 기술들은 이 관점에 기반을 두고 만들어짐
- DNN을 확률 분포로 보았을 때, 가능한 이론들에 대해 앞으로 전개 예정
 - Likelihood

- Maximum Likelihood Estimation(MLE)
- Maximum A Posterior (MAP) Function
- Cross Entropy & KL-Divergence & JSD

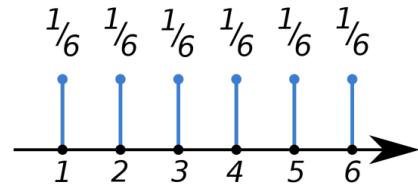
▼ Basic Statistics

▼ Random Variable & Probability Distribution

- 어떤 변수 x 가 x 라는 값을 가질 확률
 - $P(x = x)$
- 확률 분포(함수)
 - 입력: 확률 변수 x
 - 출력: x 가 각 값에 해당될 때에 대한 확률

▼ Discrete Probability Distribution

- 확률 값의 총합은 1
 - $\sum_x P(x = x) = 1$, where $0 \leq P(x = x) \leq 1, \forall x \in \mathcal{X}$
- Probability Mass Function (확률 질량 함수, PMF)



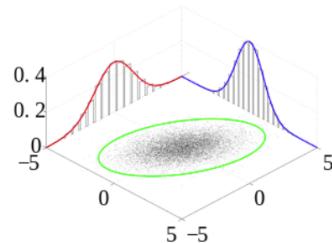
▼ Continuous Probability Distribution

- Probability Density Function (확률 밀도함수, PDF)
 - 면적의 합이 1
 - 함수 값이 1보다 클 수 있다
 - $\int p(x)dx = 1$, where $p(x) \geq 0, \forall x \in \mathbb{R}$
- 연속 확률 변수의 경우, 어떤 샘플이 주어졌을 때, 확률 값을 알 수 없다.

▼ Joint Probability

- 결합 분포

$$P(x, y)$$



▼ Conditional Probability

- 조건부 확률

$$P(y | x) = \frac{P(x,y)}{P(x)}$$
- 좀 더 친해져야 할 형태

$$\begin{aligned} \circ P(x, y) &= P(y | x)P(x) \\ &= P(x | y)P(y) \end{aligned}$$

▼ Bayes Theorem

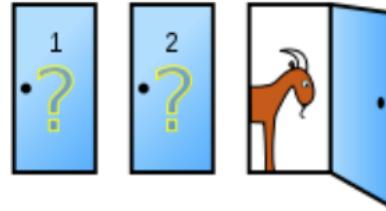
- 데이터 D 가 주어졌을 때, 가설 h 의 확률

$$\circ P(h | D) = \frac{P(D|h)P(h)}{P(D)}$$

▼ Function? or Value?

- 확률 값
 - $P(x) \rightarrow \text{Value}$
- 확률 분포 함수
 - $P(x) \rightarrow \text{Function}$
- 그럼 아래의 것들은?
 - $P(y|x) \rightarrow \text{Value}$
 - $P(Y = y|x = x)$
 - $P(y|x) \rightarrow \text{Function}$
 - $P(Y|x = x)$
 - $P(y|x) = f(x) \rightarrow \text{Function}$
 - $P(Y = y|x)$

▼ Monti-Hall Problem



- Random Variables
 - A: a door index what I selected at the first time
 - B: a door index what host selected. Host will not open the answer.
 - C: a door index of the answer.

$$P(C = 0 | A = 0, B = 1) \quad P(C = 2 | A = 0, B = 1)$$

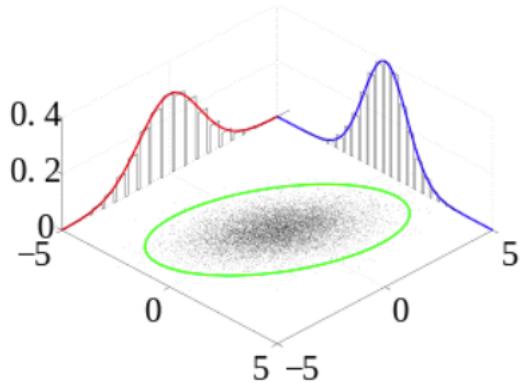
$$\begin{aligned} P(C = 2 | A = 0, B = 1) &= \frac{P(A = 0, B = 1, C = 2)}{P(A = 0, B = 1)} \\ &= \frac{P(B = 1 | A = 0, C = 2)P(A = 0, C = 2)}{P(A = 0, B = 1)} \\ &= \frac{P(B = 1 | A = 0, C = 2)P(A = 0)P(C = 2)}{P(B = 1 | A = 0)P(A = 0)} \\ &= \frac{\frac{1}{2} \times \frac{1}{3}}{\frac{1}{2}} = \frac{2}{3}, \end{aligned}$$

where $P(B = 1 | A = 0) = \frac{1}{2}$, $P(C = 2) = \frac{1}{3}$, and $P(B = 1 | A = 0, C = 2) = 1$.

$$\begin{aligned}
P(C = 0 \mid A = 0, B = 1) &= \frac{P(A = 0, B = 1, C = 0)}{P(A = 0, B = 1)} \\
&= \frac{P(B = 1 \mid A = 0, C = 0)P(A = 0, C = 0)}{P(A = 0, B = 1)} \\
&= \frac{P(B = 1 \mid A = 0, C = 0)P(A = 0)P(C = 0)}{P(B = 1 \mid A = 0)P(A = 0)} \\
&= \frac{\frac{1}{2} \times \frac{1}{3}}{\frac{1}{2}} = \frac{1}{3}, \\
\text{where } P(B = 1 \mid A = 0, C = 0) &= \frac{1}{2}
\end{aligned}$$

▼ Marginal Distribution

- 결합 분포에서 한 변수를 적분한 형태



$$\begin{aligned}
P(x) &= \int P(x, z) dz \\
&= \int P(x \mid z) P(z) dz \\
&= \int P(z \mid x) P(x) dz = P(x) \int P(z \mid x) dz
\end{aligned}$$

▼ Expectation and Sampling

- 앞으로 자주 보게 될 수식
 - $\mathbb{E}_{x \sim P(x)}[f(x)]$
- 이것을 전개해 보면,
 - $\mathbb{E}_{x \sim P(x)}[f(x)] = \sum_{x \in \mathcal{X}} P(x) \cdot f(x)$

▼ Example: Rolling a Dice

- 주사위의 기대 값은?

$$\begin{aligned}
\mathbb{E}_{x \sim P(x)}[f(x)] &= \sum_{x \in \{1, 2, 3, 4, 5, 6\}} P(x = x) \cdot f(x) \\
&= \frac{1}{6} \times (f(1) + f(2) + f(3) + f(4) + f(5) + f(6)) \\
&= \frac{1}{6} \times (1 + 2 + 3 + 4 + 5 + 6) = 3.5, \text{ where } f(x) = x.
\end{aligned}$$

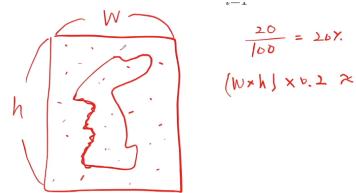
▼ Also, we can do this

- 해석해보자

$$\begin{aligned}
 P(x) &= \int P(x, z) dz \\
 &= \int P(x | z) P(z) dz \\
 &= \mathbb{E}_{z \sim P(z)} [P(x | z)]
 \end{aligned}$$

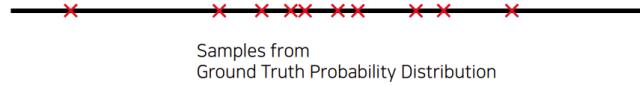
▼ Monte-Carlo

- 확률 분포로부터 샘플링을 통해 f 의 가중 평균을 구해보자.
- $\mathbb{E}_{x \sim P(x)} [f(x)] \approx \frac{1}{n} \sum_{i=1}^n f(x_i)$, where $x_i \sim P(x)$

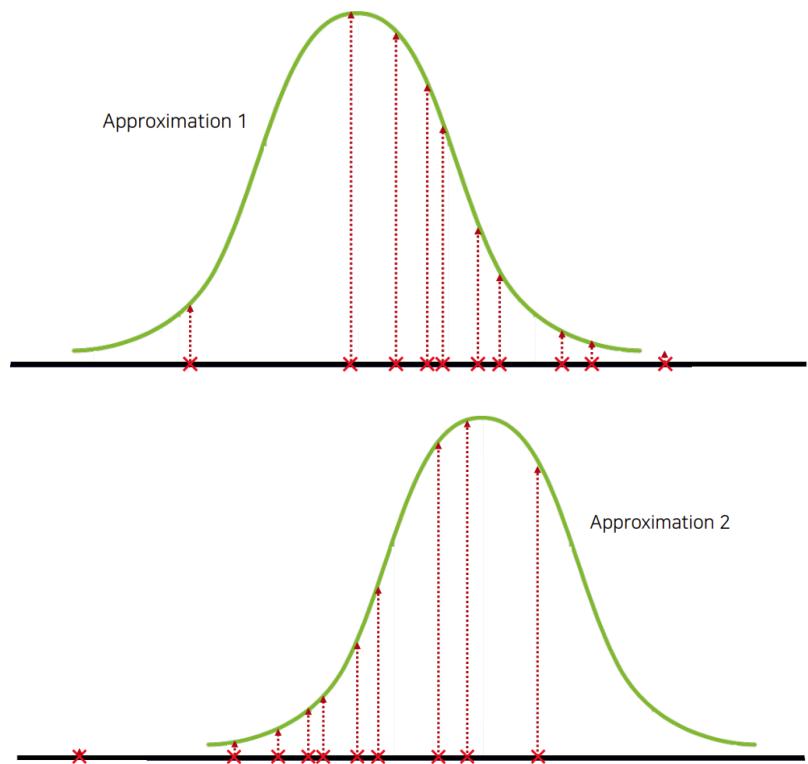


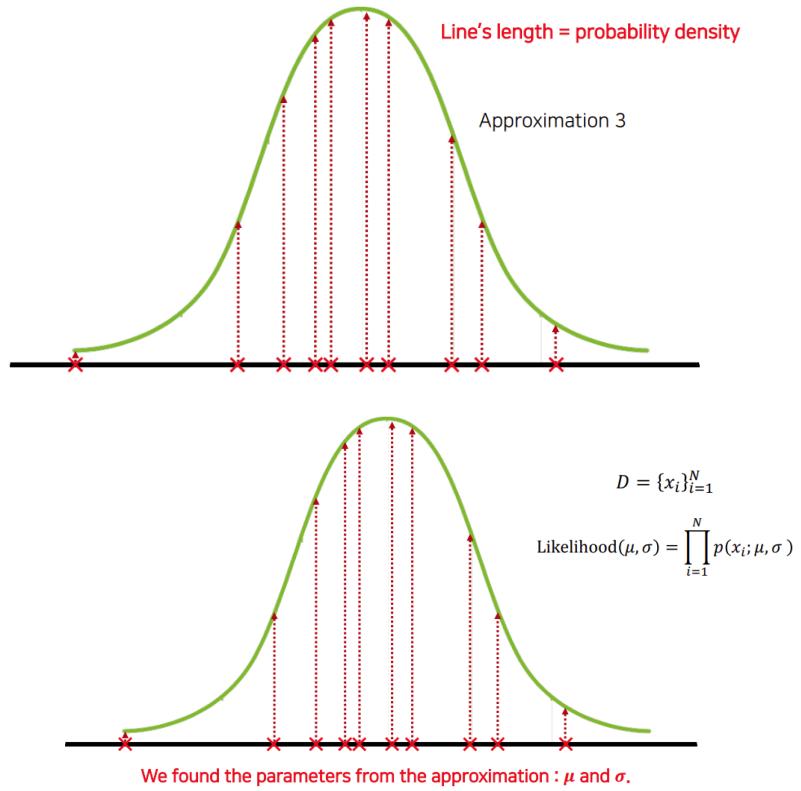
▼ Maximum Likelihood Estimation(MLE)

▼ Gaussian Example



▼ Likelihood Function





▼ Example without DeepLearning

- 입력으로 주어진 확률 분포(파라미터)가 데이터를 얼마나 잘 설명하는지 나타내는 점수(Likelihood)를 출력하는 함수
 - 입력: 확률 분포를 표현하는 파라미터
 - 출력: 데이터를 설명하는 정도
- 데이터를 잘 설명하는지 알 수 있는 방법
 - 데이터가 해당 확률 분포에서 높은 확률 값을 가질 것

▼ Example by Simple Solution

- 내기를 좋아하는 기현이는 어느 날 주사위 게임에서 돈을 많이 잃었습니다. 사기 당한 것을 깨달은 기현이는 복수를 위해 주사위의 트릭을 알고 싶습니다. 다행히 기현이는 기억력이 좋아서 주사위의 나온 숫자를 모두 기억하고 있습니다.
- 20번 던졌을 때, 나온 숫자
 $\mathcal{D} = \{5, 6, 4, 6, 5, 2, 6, 1, 5, 3, 1, 6, 4, 2, 5, 6, 2, 1, 4, 5\}$

▼ Example by Likelihood Estimation

- 해당 숫자가 나온 횟수와 전체 횟수를 알면, 확률 값을 추측할 수 있다.

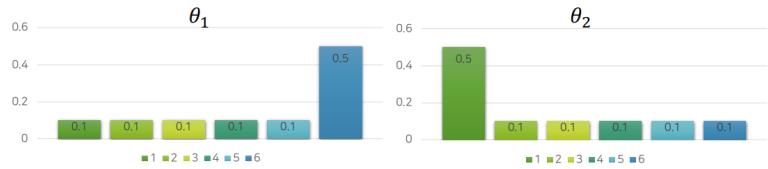


$$\theta =$$

$$\{w_1 = 0.15, w_2 = 0.15, w_3 = 0.05, w_4 = 0.15, w_5 = 0.25, w_6 = 0.25\}$$

- 이에 비춰 주사위 30이 나올 확률은?

- $P_\theta(x = 3) = 0.05$
- 목표: 주사위의 확률 분포를 알고 싶다.
- 임의의 확률 분포 생성



$$\mathcal{L}(\theta_1) = \prod_{i=1}^{n=20} P_{\theta_1}(x = x_i) = .1^3 \times .1^3 \times .1^1 \times .1^3 \times .1^5 \times .5^5 = 3.125e -$$

$$\mathcal{L}(\theta_2) = \prod_{i=1}^{n=20} P_{\theta_2}(x = x_i) = .5^3 \times .1^3 \times .1^1 \times .1^3 \times .1^5 \times .1^5 = 1.25e -$$

- 임의의 확률 분포 생성 Again



$$\mathcal{L}(\theta_1) = \prod_{i=1}^{n=20} P_{\theta_1}(x = x_i) = .1^3 \times .1^3 \times .1^1 \times .1^3 \times .1^5 \times .5^5 = 3.125e -$$

$$\mathcal{L}(\theta_3) = \prod_{i=1}^{n=20} P_{\theta_3}(x = x_i) = .15^3 \times .15^3 \times .05^1 \times .15^3 \times .25^5 \times .25^5 = 1$$

▼ Log Likelihood

- 앞선 예제에서 볼 수 있듯이, Likelihood는 확률 값의 곱으로 표현되며
- Underflow의 가능성 → 매우 작은 값이 발생
- 따라서 Log를 취하여 곱셈을 덧셈으로 바꾸고, Log Likelihood로 문제를 해결
- 덧셈이 곱셈보다 연산도 빠름

$$\prod_{i=1}^n P_\theta(x = x_i) \rightarrow \sum_{i=1}^n \log P_\theta(x = x_i)$$

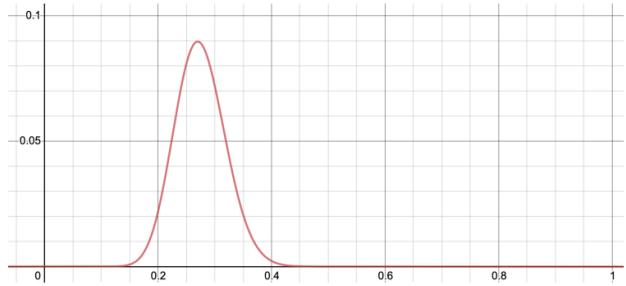
▼ MLE via Gradient Ascent

- 랜덤 생성 대신, Gradient Ascent를 통해,
- Likelihood 값을 최대 만드는 파라미터(θ)를 찾자.
- $\theta \leftarrow \theta + \alpha \cdot \frac{\partial \mathcal{L}(\theta)}{\partial \theta}$

▼ Example of MLE via Gradient Ascent

- $n = 100$ 번 던졌을 때, $k = 27$ 번 앞면(True)이 나오는 동전이 있다.
- 이 동전의 확률 분포(파라미터 θ)를 추정하자
- Binomial Distribution

$$\mathcal{L}(\theta) = \frac{n!}{k!(n-k)!} \times \theta^k \times (1-\theta)^{n-k}$$



$$x = \theta$$

$$y = \text{likelihood}$$

- 미분하여 변곡점을 찾아 최대값을 찾는다.
- 미분이 불가능하다면, Gradient Ascent를 통해 찾는다.

▼ Summary

- 우리는 가상의 확률 분포를 모사하는 확률 분포의 파라미터(θ)를 찾고 싶다.
- 목표 확률 분포로부터 데이터를 수집한 후, 데이터를 잘 설명하는 파라미터를 찾자.
 - Likelihood라는 값을 통해 얼마나 잘 설명하는지 알 수 있다.
 - Likelihood function은 θ 를 입력으로 받아, 데이터들의 θ 에 대한 확률 값의 곱을 출력
- Likelihood를 최대화하는 파라미터를 찾으면, 주어진 데이터를 가장 잘 설명한다.
 - Gradient Ascent를 통해서 찾자.

DNN with MLE

▼ Before we start,

- 모두 같은 표현
 - $P_\theta(x) = P(x; \theta) = P(x | \theta)$
마지막 θ 의 경우, random variable
 - $P_\theta(y | x) = P(y | x; \theta) = P(y | x, \theta)$
 y 라는 확률 분포를 따르는 x 가 주어졌을 때, y 의 확률 값

▼ Again, our objective is

- 데이터를 넣었을 때 출력을 반환하는 가상의 함수를 모사하는 것
- 확률 분포로부터 샘플링하여 데이터를 넣었을 때,
확률 분포를 반환하는 가상의 함수를 모사하는 것
 - 출력 분포에서 샘플링하면 원하는 출력 값을 얻을 수 있다.
- Example: 손 글씨가 주어졌을 때, 글씨의 클래스의 확률 분포
 $P(c | x)$, where $x \sim P(x)$

▼ Review: Maximum Likelihood Estimation

- 우리는 가상의 확률 분포를 모사하는 확률 분포의 파라미터(θ)를 찾고 싶다.
- 목표 확률 분포로부터 데이터를 수집한 후, 데이터를 잘 설명하는 파라미터를 찾자.
 - Likelihood라는 값을 통해 얼마나 잘 설명하는지 알 수 있다.
 - Likelihood function은 θ 를 입력으로 받아, 데이터들의 θ 에 대한 확률 값의 곱을 출력
- Likelihood를 최대화하는 파라미터를 찾으면, 주어진 데이터를 가장 잘 설명한다.
 - Gradient Ascent를 통해서 찾자.

$$\circ \theta \leftarrow \theta + \alpha \cdot \frac{\partial \mathcal{L}(\theta)}{\partial \theta}$$

▼ Parameters for Probability Distribution

- Bernoulli Distribution

$$\circ \theta = \{p\}$$

- Gaussian Distribution

$$\circ \theta = \{\mu, p\}$$

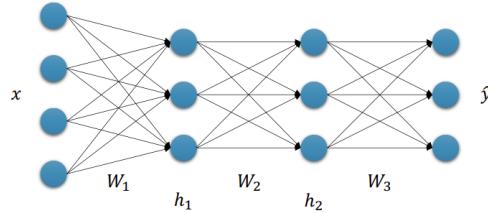
- Parameters for Probability Distribution

$$\theta = \{W_1, b_1, W_2, b_2, \dots, W_\ell, b_\ell\}$$

◦ Layer들의 weight가 파라미터이다.

◦ 가우시안 분포의 μ, σ 와 다를게 없다.

◦ 마찬가지로 Gradient Ascent를 통해 Likelihood를 최대로 하는 파라미터(θ)를 찾을 수 있다!



▼ Negative Log Likelihood(NLL)

- 하지만 대부분의 딥러닝 프레임워크들은 Gradient Descent만 지원

$$\theta \leftarrow \theta - \alpha \cdot \frac{\partial \mathcal{L}(\theta)}{\partial \theta}$$

- 따라서 Maximization 문제에서 minimization 문제로 접근

▼ DNN with MLE

- 분포 $P(x)$ 로부터 샘플링한 데이터 x 가 주어졌을 때, 파라미터 θ 를 갖는 DNN은 조건부 확률 분포를 나타낸다.

$$P(y | x; \theta), \text{ where } x \sim P(x).$$

- 이때, 우리는 Gradient Descent를 통해 NLL을 최소화하는 θ 를 찾을 수 있다.

$$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmin}} \sum_{i=1}^N -\log P(y_i | x_i; \theta)$$

▼ Summary

- MLE를 통해 수집한 데이터셋을 잘 설명하는 확률 분포의 파라미터를 찾을 수 있음
- Neural Networks 또한 확률 분포 합수이므로, MLE를 통해 파라미터를 찾을 것
 - 최대화 대신 최소화를 위해 NLL을 Gradient Descent
- Gradient Descent를 수행하기 위해선, 파라미터에 대한 미분이 필요함
 - 이를 효율적으로 수행하기 위해 back-propagation을 활용

MLE Equations

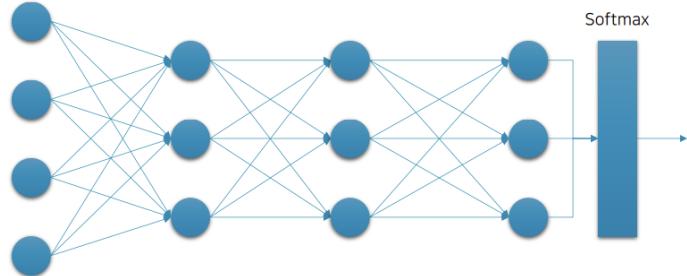
▼ MLE Equations

--

$$\begin{aligned}
D &= \{(x_i, y_i)\}_{i=1}^N \\
\hat{\theta} &= \underset{\theta \in \Theta}{\operatorname{argmax}} \sum_{i=1}^N \log P(y_i | x_i; \theta) \\
&= \underset{\theta \in \Theta}{\operatorname{argmin}} - \sum_{i=1}^N \log P(y_i | x_i; \theta) \\
\theta &\leftarrow \theta - \alpha \nabla_{\theta} \mathcal{L}(\theta)
\end{aligned}$$

▼ Connection to DNN

- We can consider **softmax** result to probability distribution.



$$\begin{aligned}
D &= \{(x_i, y_i)\}_{i=1}^N \\
\hat{\theta} &= \underset{\theta \in \Theta}{\operatorname{argmin}} - \sum_{i=1}^N \log P(y_i | x_i; \theta)
\end{aligned}$$

→ By implement →

$$\begin{aligned}
\hat{y}_i &= f_{\theta}(x_i) \rightarrow \text{softmax result} \\
-\sum_{i=1}^N \log P(y_i | x_i; \theta) &= -\sum_{i=1}^N y_i^T \cdot \log \hat{y}_i
\end{aligned}$$

▼ MLE(NLL) and Cross Entropy Loss

- Minimizing NLL is equal to Minimizing Cross Entropy.
- $-\frac{1}{N}$ 만 다르지만, 미분시 상수라 어차피 사라지게 된다.

$$\begin{aligned}
\text{CE}(y_{1:N}, \hat{y}_1 : N) &= -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^d y_{i,j} \log \hat{y}_{i,j} \\
&= -\frac{1}{N} \sum_{i=1}^N y_i^T \cdot \log \hat{y}_i,
\end{aligned}$$

where $y_{1:N} \in \mathbb{R}^{N \times d}$, $\hat{y}_{1:N} \in \mathbb{R}^{N \times d}$.

▼ Appendix: Maximum A Posterior(MAP)

▼ Bayes Theorem

$$P(h|D) = \frac{\underbrace{P(D|h)P(h)}_{\text{Posterior}}}{\underbrace{P(D)}_{\text{Evidence}}}$$

Likelihood Prior
 Evidence

h : hypothesis
 D : Data

▼ MAP Estimation Example

- 절도 사건의 범인은 발자국을 남겼습니다.
 - 신발 사이즈 240 일 때, 범인은 남자일까? 여자일까?
 $P(y | x = 240)$
 - 지인 중에 신발 사이즈가 240 이었던 사람들을 떠올려보자..
 - 여자 중에 많을까? 남자 중에 많을까?
 $P(x = 240 | y)$
 - 그런데 범행 장소가 군부대라면?
 $P(y = \text{male}) > P(y = \text{female})$
-
- Likelihood는 여자일 가능성이 높지만, Prior를 고려하였을 때, 범인은 남자일 가능성이 높다.

$$P(y | x = 240) = \frac{P(x=240|y)P(y)}{P(x=240)}$$

$$\frac{P(x=240|y=\text{male})P(y=\text{male})}{P(x=240)} > \frac{P(x=240|y=\text{female})P(y=\text{female})}{P(x=240)}$$

▼ MAP Estimation

- Find \hat{h} , which maximizes posterior.

$$\begin{aligned} \hat{h} &= \operatorname{argmax}_{h \in \mathcal{H}} P(h | D) \\ &= \operatorname{argmax}_{h \in \mathcal{H}} \frac{P(D | h)P(h)}{P(D)} \\ &= \operatorname{argmax}_{h \in \mathcal{H}} P(D | h)P(h) \end{aligned}$$

◦ $P(D|h) \rightarrow \text{Likelihood}$

$P(D)$ 는 어차피 $P(h)$ 에 대해 다루기 때문에 상수 취급되어 사라짐.

▼ Bayesian vs Frequentist

- | | |
|--|--|
| <ul style="list-style-type: none"> • Bayesian 관점 ◦ 파라미터 또한 random variable이 며, prior 분포를 따를 것. ◦ 미래의 uncertainty 까지 고려 ◦ Prior에 대한 가정 필요 | <ul style="list-style-type: none"> • Frequentist 관점 ◦ 파라미터는 최적화의 대상 ◦ 현재까지의 정보를 바탕으로 추정 ◦ Overfitting에 취약함 |
|--|--|

$$\begin{aligned} \hat{\theta} &= \operatorname{argmax}_{\theta \in \Theta} P(\theta | D) \\ &= \operatorname{argmax}_{\theta \in \Theta} \frac{P(D | \theta)P(\theta)}{P(D)} \\ &= \operatorname{argmax}_{\theta \in \Theta} P(D | \theta)P(\theta) \end{aligned}$$

→ 확률 분포가 존재하게 된다.

▼ Summary

- MAP를 통해 우리는 posterior를 최대화하는 가정을 찾을 수 있음

- 마찬가지로 주어진 데이터셋에 대한 posterior를 최대화하는 파라미터(θ)를 찾을 수 있음
- Bayesian 관점에서는 prior에 대한 가정을 통해, 앞으로의 uncertainty까지 고려
 - 이를 통해 overfitting 등의 문제도 해결할 수 있음
 - Bayesian Deep Learning에 대한 다양한 시도들도 이어지고 있음

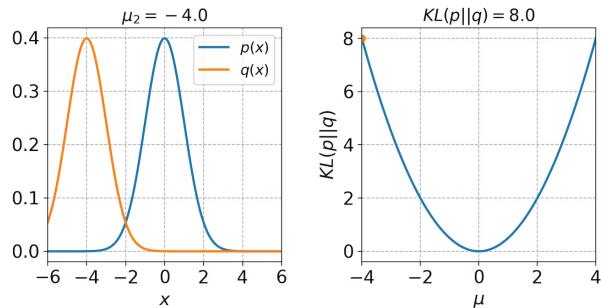
▼ Kullback-Leibler Divergence(KLD)

▼ Kullback-Leibler Divergence

- Measuer dissimilarity of two probability distributions, p and q
- Since KLD is asymmetric(비대칭), **it is not a distance**

$$\begin{aligned} \text{KL}(p\|q) &= -\mathbb{E}_{x \sim p(x)} \left[\log \frac{q(x)}{p(x)} \right] \\ &= - \int p(x) \log \frac{q(x)}{p(x)} dx \end{aligned}$$

- p 관점에서 다른을 측정
- p 에서 현재 샘플링하기 때문 ($x \sim p(x)$)



- 비슷할수록 작은 값 반환
- 같은 분포라면 0을 반환

▼ DNN Optimization using KL-Divergence

$$\begin{aligned} \mathcal{L}(\theta) &= -\mathbb{E}_{x \sim p(x)} \left[\mathbb{E}_{y \sim p(y|x)} \left[\log \frac{p_\theta(y|x)}{p(y|x)} \right] \right] \\ &= KL(P(y|x)||P_\theta(y|x)) \\ \rightarrow \text{By Monte-Carlo} \rightarrow KL(P||P_\theta) &= 0 \rightarrow \\ D &= \{(x_i, y_i)\}_{i=1}^N \\ \mathcal{L}(\theta) &\approx -\frac{1}{N \cdot k} \sum_{i=1}^N \sum_{j=1}^k \log \frac{p_\theta(y_{i,j} | x_i)}{p(y_{i,j} | x_i)} \\ &\approx -\frac{1}{N} \sum_{i=1}^N \log \frac{p_\theta(y_i | x_i)}{p(y_i | x_i)}, \text{ if } k = 1. \end{aligned}$$

- y 를 k 번 Sampling

$$\begin{aligned} \mathcal{L}(\theta) &= -\frac{1}{N} \sum_{i=1}^N \log \frac{p_\theta(y_i | x_i)}{p(y_i | x_i)} \\ \hat{\theta} &= \underset{\theta \in \Theta}{\operatorname{argmin}} \mathcal{L}(\theta) \\ \theta &\leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\theta) \end{aligned}$$

▼ Jenson-Shannon Divergence(JSD)

- 해당 블로그에 KLD와 JSD의 개념에 대해 잘 정리되어 있습니다.

KL divergence와 JSd의 개념 (feat. cross entropy)

1. KLD / JSd 얼마전 GAN 논문을 읽는데 KLD, JSd에 관한 내용이 나왔다. 그냥 단순히 두 확률분포 간의 distance를 나타내는 divergence라고 생각했는데, 사실은 이게 아니라 더 심오한 내용이

<https://ddongwon.tistory.com/118>

$$\begin{aligned}
 D_{\text{KL}}(P||Q) &= H(p, q) - H \\
 &= - \sum_i p_i \log_2 q_i + \sum_i p_i \log_2 p_i \\
 &= \sum_i P(i) \log \frac{P(i)}{Q(i)}
 \end{aligned}$$

▼ Information & Entropy

▼ Information

- 본디, 통신이나 압축을 위해 주로 다루어지던 분야
- Representation Learning에 관해서 다루다 보니 자연스럽게 연결됨
- 불확실성을 나타내는 값

$$I(x) = -\log P(x)$$

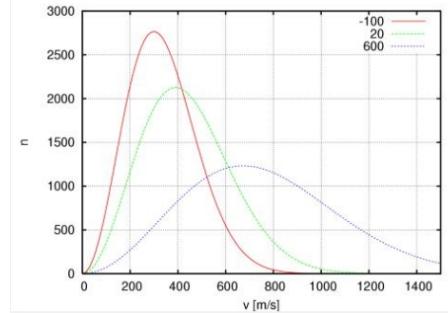
▼ Examples

1. 올 여름 대한민국의 평균 여름 기온은 26도입니다.
 2. 올 여름 대한민국의 평균 여름 기온은 8도입니다.
-
1. 내일 아침 해는 동쪽 하늘에서 뜹니다.
 2. 내일 아침 해는 서쪽 하늘에서 뜹니다.
- 확률이 낮을수록 많은 정보량을 담고 있다.
- 예를 들어, 해가 동쪽에서 뜬다면 그것에는 많은 의미가 담겨 있음.

▼ Entropy

- 정보량의 기대값(평균)
- 분포의 평균적인 불확실성을 나타내는 값
 - 분포의 형태를 예측해볼 수 있음

$$H(P) = -\mathbb{E}_{x \sim P(x)}[\log P(x)]$$



▼ Cross Entropy

- 분포 P 의 관점에서 본 분포 Q 의 정보량의 평균
- 두 분포가 비슷할 수록 작은 값을 가진다.

$$H(P, Q) = -\mathbb{E}_{x \sim P(x)}[\log Q(x)]$$

▼ DNN Optimization using Cross Entropy

- Classification 문제에서 Cross Entropy Loss를 사용하여 최소화

$$\begin{aligned}
\text{CE}(y_{1:N}, \hat{y}_{1:N}) &= -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^k y_{i,j} \log \hat{y}_{i,j} \\
&= -\frac{1}{N} \sum_{i=1}^N i = 1^N y_i^T \cdot \log \hat{y}_i, \\
\text{where } y_{1:N} &\in \mathbb{R}^{N \times d}, \hat{y}_{1:N} \in \mathbb{R}^{N \times d}, \frac{1}{N} \sum_{i=1}^N \log P(y_i | x_i; \theta), \text{ if } k = 1. \\
&= -\frac{1}{N} \sum_{i=1}^N y_i^T \cdot \log \hat{y}_i
\end{aligned}$$

▼ KL-Divergence and Cross Entropy

- KL-Divergence와 Cross Entropy를 θ 로 미분하면 같다.

$$\begin{aligned}
\text{KL}(p \| p_\theta) &= -\mathbb{E}_{x \sim p(x)} \left[\log \frac{p_\theta(x)}{p(x)} \right] \\
&= - \int p(x) \log \frac{p_\theta(x)}{p(x)} dx \\
&= - \int p(x) \log p_\theta(x) dx + \int p(x) \log p(x) dx \\
&= H(p, p_\theta) - H(p) \\
&\rightarrow \nabla_\theta \text{KL}(p \| p_\theta) = \nabla_\theta H(p, p_\theta) - \nabla_\theta H(p)
\end{aligned}$$

- CE – Entropy

▼ Summary

- Objective:
 - 확률 분포 $P(x)$ 로부터 수집한 데이터셋 D 를 통해, 확률 분포 함수 $P(y|x)$ 를 근사하고 싶다.
 - 확률 분포 함수 신경망 $P_\theta(y|x)$ 를 통해 이를 수행하자.
- KL-Divergence(또는 Cross Entropy)가 최소가 되도록 gradient descent 수행

▼ Appendix: MSE with Probabilistic Perspective — Regression

▼ MSE with Probabilistic Perspective

- Gaussian PDF
- MLE with Gradient Descent

$$\begin{aligned}
p(x; \mu, \sigma) &= \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \{(x_i, y_i)\}_{i=1}^N \\
\theta &\stackrel{\text{argmax}}{\equiv} \sum_{\theta \in \Theta} i = 1^N \log p(y_i | x_i; \theta) \\
\log p(x; \mu, \sigma) &= -\log \sigma \sqrt{2\pi} - \frac{1}{2} \left(\frac{x-\mu}{\sigma} \right)^2 \\
-\log p(x; \mu, \sigma) &= \log \sigma \sqrt{2\pi} + \frac{1}{2} \left(\frac{x-\mu}{\sigma} \right)^2 \\
&\quad \underset{\theta}{\leftarrow} \theta - \alpha \nabla_\theta \mathcal{L}(\theta)
\end{aligned}$$

▼ Get gradient of NLL

$$\begin{aligned}
-\log p(y_i | x_i; \phi, \psi) &= \log \sigma_\psi(x_i) \sqrt{2\pi} + \frac{1}{2} \left(\frac{y_i - \mu_\phi(x_i)}{\sigma_\psi(x_i)} \right)^2, \text{ where } \theta = \{\cdot\} \\
-\nabla_\phi \log p(y_i | x_i; \phi, \psi) &= \nabla_\phi \log \sigma_\psi(x_i) \sqrt{2\pi} + \nabla_\phi \frac{1}{2} \left(\frac{y_i - \mu_\phi(x_i)}{\sigma_\psi(x_i)} \right)^2 \\
&= \frac{1}{2 \cdot \sigma_\psi(x_i)^2} \nabla_\phi (y_i - \mu_\phi(x_i))^2 \\
&= \alpha \cdot \nabla_\phi (y_i - \mu_\phi(x_i))^2, \text{ where } \alpha = \frac{1}{2 \cdot \sigma_\psi(x_i)^2}.
\end{aligned}$$

$$\hat{y}_i = (y_i - \mu_\phi(x_i))^2$$

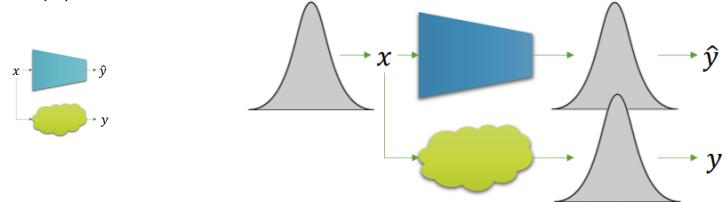
▼ Wrap-up

▼ Before this chapter,

- Our objective is:
 - 우리의 세계(머릿속)에 존재하는 가상의 함수를 모사하자.
- 주어진 입력(x)에 대해서 원하는 출력(y)을 반환하도록,
손실함수를 최소화하는 파라미터(θ)를 찾자.
- Gradient Descent를 수행하기 위해 back-propagation을 수행하자.

▼ Before(func) vs After(Probability)

- | | |
|--|---|
| <ul style="list-style-type: none"> • Before: 함수를
배우자 ◦ Deterministic
target값을
예측 | <ul style="list-style-type: none"> • After: 확률 분포 함수를 배우자 ◦ 수학적으로 좀 더 설명 가능함 ◦ 불확실성까지 학습 |
|--|---|



▼ After

- Our objective is:
 - 우리의 세계(머릿속)에 존재하는 가상의 확률 분포 함수를 모사하자.
- 확률 분포 $P(x)$ 에서 수집한 입력 데이터 x 에 대해서
원하는 조건부 확률 분포 $P(y|x)$ 또는 샘플링한 출력 데이터 y 를 반환하도록,
손실함수를 최소화하는 확률 분포 함수의 파라미터(θ)를 찾자.
 - Maximum Likelihood Estimation
- Gradient descent를 잘 수행하기 위해 back-propagation을 수행하자.

2. Paper

▼ 1. Introduction

- ▼ 현재 딥러닝은 다양한 분야에서 성공을 이루고 있습니다.
- 주로, Back-propagation & Dropout 개념, relu등의 개념이 포함됩니다.
- relu등과 같은 piecewise linear units를 통해 더 많은 층을 쌓을 수 있게 되었음
 - 그냥 비선형 함수(활성화 함수)를 통해, 층을 깊게 쌓음을 의미합니다.
- ▼ 하지만 기존 Deep Generative Models은 2가지 어려움을 직면한다.

- 당시를 기준으로 생각해보면 Deep Generative Models은 VAE 계열의 모델을 의미하는 것 같습니다.

1. Difficulty of approximating many intractable probabilistic computations

- a. MLE나 관련 전략

2. Difficulty of leveraging the benefits of piecewise linear units in the generative context

▼ GAN에서는 새로운 procedure(Adversarial Nets)를 통해 이러한 어려움을 회피 가능함을 제시합니다.

- Adversarial Nets

- 생성자(Generator)과 판별자(Discriminator)을 적대적 관계를 이루며 이미지를 생성합니다.

- 판별자(Discriminator)은 진위 판별을 학습합니다.(경찰)

- 생성자(Generator)은 기존 데이터의 분포를 학습합니다.(위조범)

- 즉, 생성자와 판별자는 게임처럼 싸우며 학습하는 컨셉입니다.

- Adversarial Nets 방식은 기존의 Markov chains이나 근사 추론을 하지 않아도 됩니다.

- ()

- 즉, 모든 모델은 multi-layer perceptron 기반으로 학습됩니다.

- =

▼ 2. Related Work

▼ Deep Learning in Unsupervised Learning

- GAN 이전에도 딥러닝을 사용한 비지도 학습 방법이 있었습니다.

- 복잡한 데이터의 특징을 잡아내기 위해 다양한 비지도 학습 기법들이 사용되었습니다.

▼ Generative Models

- GAN 이전에도 생성 모델은 주요 연구 주제였습니다.

- 예를 들어, Restricted Boltzmann Machines (RBM)이나 Deep Belief Networks (DBN)와 같은 방법들이 있었습니다.

▼ Markov Chains

- 일부 생성 모델은 Markov Chains를 사용하여 데이터를 생성하였습니다.

- 하지만, 이러한 방법은 종종 느리고 비효율적이었습니다.

▼ Variational Methods

- Variational Autoencoders (VAE)와 같은 방법들은 데이터의 확률 분포를 근사하는 방식으로 생성 모델을 학습시켰습니다.

▼ Other Models

- 다른 여러 가지 모델들이 생성 작업에 사용되었으며, 이들 중 일부는 GAN의 등장과 함께 주요 연구 주제가 되었습니다.

▼ 3. Adversarial nets

▼ 모델 구조: GAN은 두 개의 신경망, 생성자 (G)와 판별자 (D)로 구성됩니다.

- 생성자 (G): 랜덤 노이즈 (z)를 입력으로 받아 가짜 데이터를 생성합니다.
- 판별자 (D): 실제 데이터나 생성자가 만든 가짜 데이터를 입력으로 받아, 해당 데이터가 진짜인지(1) 아니면 가짜인지(0) 판별합니다.

▼ 학습 목표: GAN의 학습은 두 네트워크의 경쟁으로 진행됩니다.

- 생성자(G): 판별자를 속이도록 학습하여, $(D(G(z)))$ 의 값이 1에 가깝게 만듭니다.

- 판별자 (**D**): 진짜 데이터는 1로, 가짜 데이터는 0으로 분류하도록 학습합니다.

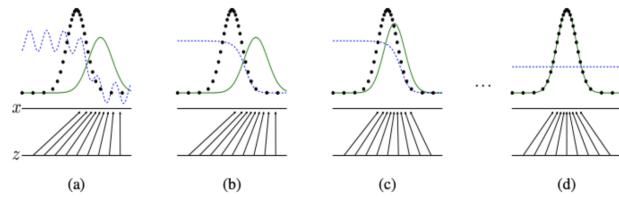
▼ 비용 함수: GAN의 학습 목표는 다음의 최소-최대 문제를 해결하는 것입니다.

- $[\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]]$
- 여기서 (\mathbb{E})는 기대 값을 의미하며, (p_{data})는 실제 데이터의 분포, (p_z)는 랜덤 노이즈의 분포를 나타냅니다.
- 위 수식에서는 Generator의 Density 정보를 미리 설정하지 않습니다.
 - 이는 $G(z)$ 에서 샘플링 된 이미지에서 implicit하게 Likelihood를 최대화함을 의미합니다.

▼ 학습 방법: GAN의 학습은 다음의 두 단계를 번갈아 가며 진행됩니다.

1. 판별자 학습: 실제 데이터와 가짜 데이터를 사용하여 판별자를 학습시킵니다.

2. 생성자 학습: 판별자가 가짜 데이터를 진짜로 판별하도록 생성자를 학습시킵니다.



- 파란선: D

- 초록선: G

- 점선: $\mathbb{E}_{x \sim p_{\text{data}}(x)}$

▼ (a) → (d)로 감에 따라 시간이 지나면서 생성모델 (G)가 원본 데이터의 분포를 학습합니다.

- 생성자는 기존 데이터의 분포인 점선에 유사해짐을 알 수 있습니다.
- 판별자는 0.5에 근사됨을 알 수 있습니다.
- $\mathbb{E}_{z \sim p_z(z)}$ 의 경우, uniform하거나 Gaussian 등 선택가능하며, 일반적으로 Gaussian 분포를 위 그림처럼 사용한다고 합니다.

▼ 4. Theoretical Results

▼ Algorithm1

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 $\{z^{(1)}, \dots, z^{(m)}\}$ from noise prior $p_g(z)$.

- Sample minibatch of m examples $\{x^{(1)}, \dots, x^{(m)}\}$ from data generating distribution $p_{\text{data}}(x)$.

- Update the discriminator by ascending its stochastic gradient:

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

end for

- Sample minibatch of m noise samples $\{z^{(1)}, \dots, z^{(m)}\}$ from noise prior $p_g(z)$.

- Update the generator by descending its stochastic gradient:

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

end for

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

▼ 실제로 코드를 작성할 경우, 위의 그림과 같은 구조로 학습합니다.

- G와 D는 동시에 학습되는 것이 아닙니다.

- 위 알고리즘 상에서는 Discriminator를 우선 학습하고,
- Generator를 학습합니다.
- 이후에는, 에 대해 수식적으로 설명하고,
- 를 입증합니다.

▼ Goal of Formulation

$$P_g \rightarrow P_{\text{data}}, D(G(z)) \rightarrow 1/2$$

▼ 4-1. Global Optimality of $p_g = p_{\text{data}}$

▼ proposition1

- Proposition: $D_G^*(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)}$

• Proof: For G fixed,

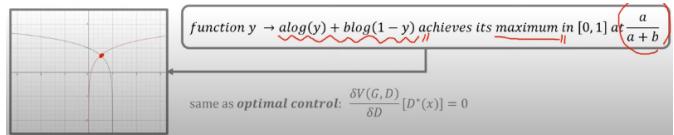
$$\circ V(G, D) = E_{x \sim p_{\text{data}}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

▼ 참고

$$\begin{aligned} E[X] &= \int_{-\infty}^{\infty} x f(x) dx \\ &= \int_x p_{\text{data}}(x) \log(D(x)) dx + \int_z p_z(z) \log(1 - D(g(z))) dz \\ &= \int_x p_{\text{data}}(x) \log(D(x)) + p_g(x) \log(1 - D(x)) dx \end{aligned}$$

▼ 참고

function $y \rightarrow a \log(y) + b \log(1 - y)$ achieves its maximum in $[0, 1]$ at $\frac{a}{a+b}$
same as optimal control: $\frac{\delta V(G, D)}{\delta D} [D^*(x)] = 0$



<https://velog.io/@lee9843/GAN-Generative-Adversarial-Nets-논문-리뷰>

▼ Theorem1

- Proposition: Global optimum point is $p_g = p_{\text{data}}$

• Proof:

$$\circ C(G) = \max DV(G, D) = E_{x \sim p_{\text{data}}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

▼ 참고

$$DG^*(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)}$$

$$\begin{aligned} &= E_{x \sim p_{\text{data}}(x)} \left[\log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)} \right] + \\ &Ex \sim p_g(x) \left[\log \frac{p_g(x)}{p_{\text{data}}(x) + p_g(x)} \right] \end{aligned}$$

$$\begin{aligned} &= E_{x \sim p_{\text{data}}(x)} \left[\log \frac{2 * p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)} \right] + \\ &Ex \sim p_g(x) \left[\log \frac{2 * p_g(x)}{p_{\text{data}}(x) + p_g(x)} \right] - \log(4) \end{aligned}$$

▼ 참고

$$\bullet E_{x \sim p_{\text{data}}(x)} \left[\log \frac{2 * p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)} \right] + \\ Ex \sim p_g(x) \left[\log \frac{2 * p_g(x)}{p_{\text{data}}(x) + p_g(x)} \right]$$

$$\begin{aligned}
& \circ \text{ removed when } \mathbf{p}_g = \mathbf{p}_{\text{data}} \\
& \bullet KL(p_{\text{data}} \| p_g) = \int_{-\infty}^{\infty} p_{\text{data}}(x) \log \left(\frac{p_{\text{data}}(x)}{p_g(x)} \right) dx \\
& \circ \\
& = KL \left(p_{\text{data}} \| \frac{p_{\text{data}}(x) + p_g(x)}{2} \right) + KL \left(p_g \| \frac{p_{\text{data}}(x) + p_g(x)}{2} \right) - \log(4)
\end{aligned}$$

▼ 참고

$$JSD(p \| q) = \frac{1}{2}KL(p \| \frac{p+q}{2}) + \frac{1}{2}KL(q \| \frac{p+q}{2})$$

$$= 2 * JSD(p_{\text{data}} \| p_g) - \log(4)$$

$= -\log(4)$ 에서 Global Optimum을 가질 수 있음이 증명되었다.

▼ 4-2. Convergence of Algorithm 1

▼ Proposition2

▼ 제안2

- 다음으로는 G 와 D 의 capacity가 충분하다면, 기반으로 각 스텝에서, 생성자와 판별자는 최적값을 구할 수 있다고 제안 한다.
- $\mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} [\log D_G^*(\mathbf{x})] + \mathbb{E}_{\mathbf{x} \sim p_g} [\log (1 - D_G^*(\mathbf{x}))]$
- then P_g converges to P_{data}

• Proof:

- Let Consider,

- $V(G, D) = U(p_g, D)$

- $U(p_g, D)$ 는 p_g 에 있는 Convex한 부분이다.

- 즉, 오차역전파를 기반으로 수렴 가능하다.

- p_g 로 미분하면 p_{data} 항은 상수취급 받으므로 p_g 만 고려하면 된다.

- G 입장에서 적은 업데이트로도 충분히 수렴 가능하다.

- → 여기 수식에 있어요!

- 다르게 표현해보자면,

- $f(x) = \sup_{\alpha \in A} f_\alpha(x)$ & $f_\alpha(x)$ 가 모든 α 에 대해서 모두 convex일 때,
 $\beta = \arg \sup_{\alpha \in A} f_\alpha(x)$ 인 조건에서 $\partial f_\beta(x) \in \partial f$ 로 표현 가능하다.

• In practice,

- adversarial nets은 $G(Z; \theta_g)$ 분포 내에서 제한적으로 p_g 분포를 나타냅니다.
- 또한 p_g 보다는 θ_g 를 최적화하는 것을 목표로 합니다.
- 다층 퍼셉트론을 사용하여 $G(Z; \theta_g)$ 를 정의하는 것은 파라미터로 공간에 표현하는 것입니다.
- 그러나 실제로 MLP의 우수한 성능은 이론적으로 보장되지 않았지만, 사용하기에 합리적인 모델이라고 합니다.
 - 이 당시에, 딥러닝 기반에서는 임계점이 여러 개가 존재하고 이를 해결할 방법이 없었습니다.

▼ 5. Experiments

▼ 실험 결과 관련 사진

- 여러 데이터 셋에서 log-likelihood 기준 다른 모델 대비 뛰어나다고 합니다.

Model	MNIST	TFD
DBN [3]	138 ± 2	1909 ± 66
Stacked CAE [3]	121 ± 1.6	2110 ± 50
Deep GSN [6]	214 ± 1.1	1890 ± 29
Adversarial nets	225 ± 2	2057 ± 26

Table 1: Parzen window-based log-likelihood estimates. The reported numbers on MNIST are the mean log-likelihood of samples on test set, with the standard error of the mean computed across examples. On TFD, we computed the standard error across folds of the dataset, with a different σ chosen using the validation set of each fold. On TFD, σ was cross validated on each fold and mean log-likelihood on each fold were computed. For MNIST we compare against other models of the real-valued (rather than binary) version of dataset.

- 아래 사진은 실제 생성된 사진 중 랜덤하게 뽑은 사진이라고 합니다.
 - 왼쪽 마킹된 사진들을 보면 실제에 존재하는 것과 같은 이미지가 생성됨을 확인할 수 있습니다.

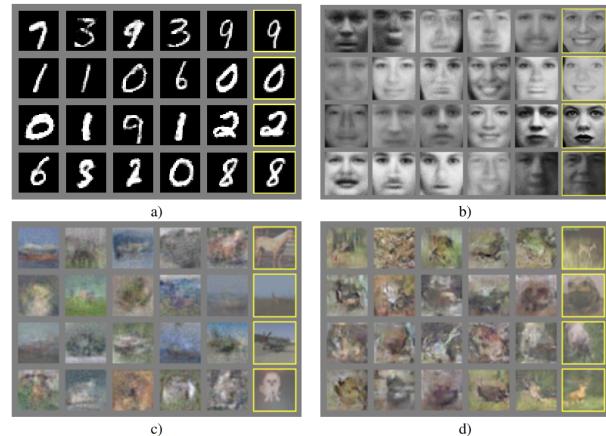


Figure 2: Visualization of samples from the model. Rightmost column shows the nearest training example of the neighboring sample, in order to demonstrate that the model has not memorized the training set. Samples are fair random draws, not cherry-picked. Unlike most other visualizations of deep generative models, these images show actual samples from the model distributions, not conditional means given samples of hidden units. Moreover, these samples are uncorrelated because the sampling process does not depend on Markov chain mixing. a) MNIST b) TFD c) CIFAR-10 (fully connected model) d) CIFAR-10 (convolutional discriminator and “deconvolutional” generator)

- Linear하게 interpolating함에 따라 $1 \rightarrow 5$ 로 변하는 과정 $7 \rightarrow 9 \rightarrow 1$ 로 변하는 것이, 합리적이게? 바꿔고 있음을 보여줍니다.



Figure 3: Digits obtained by linearly interpolating between coordinates in z space of the full model.

- 다른 기존 모델들과의 비교 정리 내용입니다.

	Deep directed graphical models	Deep undirected graphical models	Generative autoencoders	Adversarial models
Training	Inference needed during training.	Inference needed during training, MCMC needed to approximate partition function gradient.	Enforced tradeoff between mixing and power of reconstruction generation	Synchronizing the discriminator with the generator. Helvetica.
Inference	Learned approximate inference	Variational inference	MCMC-based inference	Learned approximate inference
Sampling	No difficulties	Requires Markov chain	Requires Markov chain	No difficulties
Evaluating $p(x)$	Intractable, may be approximated with AIS	Intractable, may be approximated with AIS	Not explicitly represented, may be approximated with Parzen density estimation	Not explicitly represented, may be approximated with Parzen density estimation
Model design	Nearly all models incur extreme difficulty	Careful design needed to ensure multiple properties	Any differentiable function is theoretically permitted	Any differentiable function is theoretically permitted

Table 2: Challenges in generative modeling: a summary of the difficulties encountered by different approaches to deep generative modeling for each of the major operations involving a model.

▼ 데이터셋

- 실험에서 사용된 주요 데이터셋은 MNIST, the Toronto Face Database (TFD), 그리고 CIFAR-10입니다.

▼ 기준 모델

- 비교를 위해 다양한 생성 모델들과 GAN이 사용되었습니다.

▼ 생성 결과

- 실험에서는 GAN이 다른 생성 모델들에 비해 더 나은 결과를 보였습니다.
- 특히, GAN으로 생성된 이미지는 더 선명하고, 높은 품질로 평가되었습니다.

▼ 데이터 간의 차이

- GAN은 다른 모델들에 비해 특히 훈련 데이터셋에 없는 새로운 종류의 이미지도 잘 생성하는 것으로 관찰되었습니다.

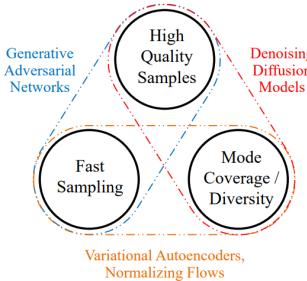
▼ 모델의 특성

- GAN의 생성자와 판별자는 균형있게 학습되었으며, 생성된 이미지의 품질은 학습 과정에서 지속적으로 향상되었습니다.

▼ 연산 시간

- GAN은 Markov Chain Monte Carlo (MCMC) 방식에 기반한 다른 생성 모델들보다 더 빠르게 결과를 생성할 수 있었습니다.

▼ 6. Advantages and disadvantages



- 해당 그림을 참고하시면, 장단점을 이해하기 더 편하실 것 같습니다,

▼ Advantages

1. **높은 품질의 생성:** GAN은 다른 생성 모델에 비해 더 높은 품질의 이미지나 데이터를 생성할 수 있습니다.
2. **모델링 없음:** GAN은 명시적으로 생성 과정을 모델링하지 않기 때문에, 복잡한 Markov Chains나 MCMC 샘플링 과정이 필요 없습니다.
3. **다양성:** GAN은 훈련 데이터셋에 없는 새로운 종류의 데이터도 잘 생성할 수 있습니다. 이는 GAN이 데이터의 다양성을 잘 잡아내기 때문입니다.

▼ Disadvantages

1. **학습의 어려움:** GAN의 학습은 생성자와 판별자 간의 경쟁적인 과정을 포함하기 때문에, 안정적으로 학습시키기 어려울 수 있습니다.
2. **모드 붕괴 (Mode Collapse):** 학습 과정에서 GAN은 특정 모드에만 집중하여 다양성을 잃을 수 있습니다. 이를 "모드 붕괴"라고 합니다.
3. **수렴 문제:** GAN의 수렴 특성은 아직 완전히 이해되지 않았습니다. 때로는 생성자와 판별자가 균형있게 학습되지 않을 수 있습니다.
4. **하이퍼파라미터 민감도:** GAN은 학습률, 네트워크 구조 등의 하이퍼파라미터에 민감하게 반응할 수 있습니다.

▼ 7. Conclusion and future work

▼ 결론 (Conclusion)

1. **새로운 접근법:** GAN은 기존의 생성 모델링 방법과는 다르게, 두 신경망 간의 경쟁을 기반으로 학습하는 새로운 방법을 제안하였습니다.

2. 성능: GAN은 다양한 데이터셋에서 높은 품질의 생성 결과를 보였으며, 특히 이미지 생성 작업에서 뛰어난 결과를 보였습니다.

▼ 향후 연구 방향 (Future Work):

1. 안정성: GAN의 학습은 때로는 불안정할 수 있으므로, 학습의 안정성을 높이는 방법에 대한 연구가 필요합니다.
2. 다양한 데이터셋 적용: GAN의 아이디어와 구조를 다양한 타입의 데이터셋에 적용하여 성능을 검증해볼 필요가 있습니다.
3. 복잡한 모델링: GAN을 확장하여 더 복잡한 데이터 생성 문제, 예를 들면 시퀀스 데이터나 구조화된 데이터에 적용하는 방법에 대한 연구가 제안됩니다.
4. 이론적 근거: GAN의 수렴 특성 및 학습 동적에 대한 더 깊은 이해를 위한 연구가 필요하다고 제안되었습니다.

3. 참고자료

- 확률 및 통계 자료 → 기존 개인 자료

▼ 원문

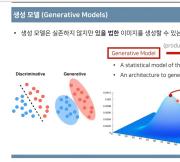
<https://arxiv.org/pdf/1406.2661.pdf>

- ▼ GAN 수식은 강의 자료와 아래의 나동빈님의 유트브 영상을 참고하여 깔끔하게 정리하였습니다.

GAN: Generative Adversarial Networks (꼼꼼한 딥러닝 논문 리뷰와 코드 실습)

생성 모델(Generative Model)은 실제로는 존재하지 않지만, 이를 위한 데이터를 만들어 내는 모델을 의미합니다. 오늘은 현대 딥러닝 기반 생성 모델에 큰 영향을 끼친 논문인 GAN(NIPS 2014)을 소개합니다. GAN은 최근까지 이미지 도메인에서의 많은 발전이 이루

 <https://www.youtube.com/watch?v=AVvIDmhHgC4>



- ▼ GAN 관련 Timeline이 잘 작성되어 있습니다.

- <https://github.com/dongb5/GAN-Timeline>

▼ 평가 지표 관련 참고자료

- 2018

Pros and Cons of GAN Evaluation Measures

Generative models, in particular generative adversarial networks (GANs), have received significant attention recently. A number of GAN variants have been proposed and have been utilized in

 <https://arxiv.org/abs/1802.03446>



- 2021

Pros and Cons of GAN Evaluation Measures: New Developments

This work is an update of a previous paper on the same topic published a few years ago. With the dramatic progress in generative modeling, a suite of new quantitative and qualitative techniques to...

 <https://arxiv.org/abs/2103.09396>



4. GAN Evaluation Metrics

- 예상 보다 평가 지표가 너무 다양했고, 각 평가 지표에 대한 비교 연구 논문이 있어 아래에 첨부합니다.

- 논문이 나올 시점에서는 IS와 같은 General한 평가 지표가 존재하지 않아, Parzen window-based log-likelihood estimates를 사용하였습니다.
- 아래 논문에서 언급되는 각 Metrics를 자세하게 살펴보지는 못했지만, GAN의 후속 연구들을 평가지표와 함께 참고할 수 있어 참고자료로 활용하면 좋을 것 같습니다.

▼ Popular한 평가 지표

- IS와 FID만 정리했습니다.

▼ IS → Quantitative Metrics

<https://arxiv.org/pdf/1606.03498.pdf>

- Inception Score는 생성 모델의 성능을 평가하기 위한 지표로, 생성된 이미지들의 품질과 다양성을 함께 고려합니다.
 - Quantitative만 고려한다.
- $IS(G) = \exp(\mathbb{E}_{x \sim p_g} [KL(p(y|x)||p(y))])$
 - x 는 생성 모델 G 에 의해 생성된 이미지입니다.
 - $p(y|x)$ 는 이미지 x 를 주어졌을 때의 조건부 클래스 확률 분포입니다. 이는 주로 Inception 모델을 사용하여 추정됩니다.
 - $p(y)$ 는 생성된 모든 이미지에 대한 평균 클래스 확률 분포입니다.
 - KL 는 Kullback-Leibler 발산을 나타냅니다.
- 간략하게 설명하면, Inception Score는 두 가지 주요 부분을 고려합니다:
 1. 각 이미지가 명확한 클래스에 속할 확률이 높아야 합니다. 이는 $p(y|x)$ 가 특정 클래스에 높은 확률을 가져야 함을 의미합니다.
 2. 다양한 클래스의 이미지가 생성되어야 합니다. 이는 $p(y)$ 가 다양한 클래스에 대해 균일하게 분포되어 있지 않아야 함을 의미합니다.
- 이 두 가지 요소를 잘 조화시키면 높은 Inception Score를 얻을 수 있습니다. IS가 높을수록 생성 모델의 성능이 더 좋다고 평가할 수 있습니다.

▼ FID → Quantitative Metrics

<https://arxiv.org/pdf/1706.08500.pdf>

- FID는 실제 데이터 분포와 생성된 데이터 분포 간의 거리를 계산하여 생성된 이미지의 품질과 다양성을 평가합니다.
- $FID(x, g) = \|\mu_x - \mu_g\|^2 + \text{Tr}(\Sigma_x + \Sigma_g - 2(\Sigma_x \Sigma_g)^{0.5})$
 - x 는 실제 데이터 분포에서 추출된 샘플들의 집합입니다.
 - g 는 생성 모델에서 생성된 샘플들의 집합입니다.
 - μ 와 Σ 는 각각 평균 및 공분산 행렬을 나타냅니다.
- 간략하게 설명하면, FID는 다음 두 가지 주요 부분을 고려합니다:
 1. 실제 데이터와 생성된 데이터의 특징 벡터들의 평균 간의 유clidean 거리입니다.
 2. 두 분포의 공분산 행렬 간의 차이를 측정하는 항입니다.
- 이 두 요소를 합산하여 실제 데이터와 생성된 데이터 간의 거리를 나타내는 하나의 스칼라 값으로 FID를 계산합니다.

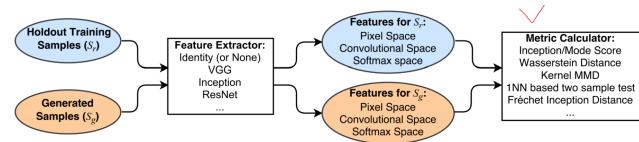
FID가 낮을수록 생성 모델의 성능이 더 좋다고 평가할 수 있습니다.

▼ [2018] Pros and Cons of GAN Evaluation Measures

Pros and Cons of GAN Evaluation Measures.pdf

- 2014년 GAN이 등장한 이후, 다양한 파생 모델들이 연구되어 왔습니다.
- GAN을 평가하는 지표는 크게 Qualitative&Quantitative Metrics로 분류됩니다.
- 해당 논문에서는 아래의 그림과 같이 5가지 주요 지표를 제시합니다.

▼ 대체로 Quantitative Metrics입니다.



▼ 해당 논문 저자는 Qualitative적인 측면도 강조합니다.

1. favor models that generate high fidelity samples (*i.e.* ability to distinguish generated samples from real ones; discriminability),
2. favor models that generate diverse samples (and thus is sensitive to overfitting, mode collapse and mode drop, and can undermine trivial models such as the memory GAN),
3. favor models with disentangled latent spaces as well as space continuity (*a.k.a* controllable sampling),
4. have well-defined bounds (lower, upper, and chance),
5. be sensitive to image distortions and transformations. GANs are often applied to image datasets where certain transformations to the input do not change semantic meanings. Thus, an ideal measure should be invariant to such transformations. For instance, score of a generator trained on CelebA face dataset should not change much if its generated faces are shifted by a few pixels or rotated by a small angle.
6. agree with human perceptual judgments and human rankings of models, and
7. have low sample and computational complexity.

▼ 이외에도 다양한 평가 지표에 대해서 수식과 함께 자세하게 정리되어 있습니다.

- 각 지표에 대한 Pros&Cons는 논문에 정리되어 있으니 참고하면 좋을 것 같습니다.

Measure	Description
1. Average Log-Likelihood [19, 20]	• For training of a generative model, it is the log-likelihood of the data using a density estimated from the generated data. • e.g., using KLD of the generated samples vs. the real samples.
2. Coverage Metric [21]	• Measures the fraction of generated samples that fall within a set of bins.
3. Latency [22]	• Measures the time taken to generate a sample.
4. Modelled Inception Score (m-IS) [23]	• Measures diversity within images sampled from a particular category, $m\text{-IS} = \frac{1}{N} \sum_{i=1}^N \text{IS}(f_i)$, where f_i is the function f that maps $\{x_i\}$ to $\{f_i(x_i)\}$. $\text{IS}(x_i) = \log P(f_i(x_i)) - \mathbb{E}_{x \sim p_\theta} [\log P(f(x))]$.
5. Mode Score [24]	• Measures the quality of generated samples based on the distribution of generated samples.
6. ARI Score [25]	• Measures the quality of generated samples based on the distribution of generated samples.
7. Perceptual Inception Distance (PID) [26]	• Measures the perceptual distance between generated samples and real samples. The metric is based on the Inception V3 network.
8. The Wasserstein Metric [27]	• Measures the perceptual distance between generated samples and real samples independently of the specific metric used to calculate the distance.
9. The Wassertein Critic [28]	• The critic (ϕ) in GAN is trained to predict high values at real samples and low values at generated samples.
10. Blinder Foward Test [29]	• Measures the effect of a direct (controllable) transformation by counting the frequency with which the transformation is successful.
11. Classification Performance [1, 10]	• Measures the classification performance of generated samples.
12. Classification Accuracy [30]	• Measures the accuracy of a classifier (trained on real samples) in classifying generated samples.
13. Classification Probability [31]	• Measures the probability of a classifier (trained on real samples) in classifying generated samples.
14. Novelty of Generated Samples [32]	• Measures the distribution of distances to the second neighbor of some query images (<i>i.e.</i> diversity).
15. Design Retention Performance [33]	• Measures the retention of design elements in generated images. The metric is based on the number of shared features between generated images and their original designs.
16. Entropy [34]	• Measures the entropy of generated samples.
17. Entropy-Wi Rate and Entropy-Wo Rate [35]	• Measures the entropy of generated samples.
18. Normalized Entropy Distribution [36]	• Compares a GAN's generated samples to the fact that the generated samples are close to real ones.
19. Adversarial Accuracy [37]	• Measures the accuracy of the discriminator according to the generated samples.
20. Generative Adversarial Critic [38]	• Measures the accuracy of the generator according to the generated samples.
21. Generative Adversarial Score [39]	• Measures the quality of generated samples based on the distribution of generated samples.
22. Image Quality Metrics [40, 50]	• Measures the quality of generated images by comparing them with real images.
23. Low-level Image Statistics [33, 51]	• Evaluates the low-level statistics of generated images to the level of several metrics, such as mean, variance, standard deviation, entropy, etc.
24. Mode Drop and Collapse [34, 52]	• Measures the quality of generated samples based on the distribution of generated samples.
25. Network Internals [1, 48, 53, 54, 55]	• Measures the quality of generated samples based on the internal representation and dynamics of models (<i>e.g.</i> space continuity).

Measure	Discriminability	Detecting Diverging	Discriminating Latent Space	With-Adversarial Attacks	Perceptual Judgments	Sensitivity to Distortion	Computational Efficiency
1. Average Log-Likelihood [18, 21]	low	low	-	[1, ∞] [0, ∞]	low	low	low
2. Coverage Metric [24]	high	low	-	[1, ∞]	high	low	low
3. Inception Score (IS) [3]	high	moderate	-	[1, ∞]	high	moderate	high
4. Modified Inception Score (m-IS) [34]	high	moderate	-	[1, ∞]	high	moderate	high
5. Mode Score [24]	high	moderate	-	[0, ∞]	high	moderate	high
6. AM Score [36]	high	moderate	-	[0, ∞]	high	moderate	high
7. Fréchet Inception Distance (FID) [37]	high	moderate	-	[0, ∞]	high	high	high
8. Minimum Mean Discrepancy (MMD) [38]	high	moderate	-	[0, ∞]	high	high	high
9. The Wasserstein Critic [28]	high	moderate	-	[0, ∞]	-	-	low
10. Birthday Paradox Test [27]	low	high	-	[1, ∞]	low	low	-
11. Chi-squared Goodness-of-fit Test (C2ST) [39]	low	moderate	-	[0, ∞]	high	moderate	high
12. Classification Performance [1, 15]	high	low	-	[0, 1]	low	-	-
13. Boundary Distortion [41]	low	low	-	[0, 1]	-	-	-
14. NRS [42]	low	moderate	-	[0, ∞]	low	moderate	high
15. Image Retrieval Performance [34]	moderate	low	-	-	low	-	-
16. Generative Adversarial Metric (GAM) [31]	high	low	-	*	-	-	moderate
17. Entropy-Wi Rate and SKT Rating [43]	high	low	-	[0, 1]	-	-	low
18. NRD [32]	high	low	-	[0, 1]	-	-	poor
19. Adversarial G-Divergence [44]	high	low	-	[0, 1], [0, ∞]	-	-	low
20. Goodwillie Error [45]	high	low	-	[0, 1]	-	-	low
21. Reconstruction Error [46]	moderate	low	-	[0, ∞]	-	moderate	moderate
22. Image Quality Metrics [40, 51]	low	moderate	-	*	high	high	-
23. Low-level Image Statistics [33]	low	moderate	-	*	low	low	high
24. Precision, Recall and F1 score [23]	low	high	✓	[0, 1]	-	-	-

Table 2: Meta measures of GAN quantitative evaluation scores. Notice that the rating are relative. “-” means unknown (hence warranting further research). “*” indicates that several bounds are provided. “✓” indicates that the metric is able to detect mode collapse, while some of the measures might be possible. It seems that most of the measure do not systematically evaluate disentanglement in the latent space.

▼ [후속 논문][2021] Pros and Cons of GAN Evaluation Measures: New Developments

Pros and Cons of GAN Evaluation Measures_New Development.pdf

- 2018년도 연구 이후, 기존 GAN을 바탕으로 다양한 연구들이 이뤄졌습니다.

- 이에 따라, Evaluation Metrics에 대해 추가적으로 연구가 이루어졌습니다.

▼ 추가 평가 지표

▼ Quantitative

▼ Specialized Variants of Frechet Inception Distance and Inception Score

- Spatial FID(sFID)
- Class-aware FID and Conditional FID
- Fast FID
- Memorization-informed FID(MiFID)
- Unbiased FID and IS
- Clean FID
- Frechet Video Distance(FVD)

▼ Methods based on Analysing Data Manifold

- Local Intrinsic Dimensionality(LID)
- Intrinsic Multi-scale Distance(IMD)
- Perceptual Path Length(PPL)
- Linear Separability in Latent Space
- Classification Accuracy Score(CAS)
- Non-parametric Test to Detect Data-Copying
- Measures that Probe Generalization in GANs

▼ New Ideas based on Precision and Recall

- Density and Coverage
- Alpha Precision and Recall
- Duality GAP Metric
- Spectral Methods
- Caption Score(Caps)

▼ Qualitative

- Human Eye Perceptual Evaluation (HYPE)
- Neuroscore
- Seeing What a GAN Can Not Generate
- Measuring GAN Steerability
- GAN Dissection
- A Universal Fake vs Real Detector

▼ 아래 표의 경우, 2021년을 기준으로 추가적으로 잘 정리되어 있으며 수식은 해당 논문을 참고해주시기 바랍니다.

	Quantitative /Analysis /Optimization	Qualitative	Overfitting /Memorization	Latent Space Disentanglement	Deepfake Detection
FID & IS Variants					
Spatial FID (sFID)	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		
Class-aware FID (CAFD)	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		
Conditional FID	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		
Fast FID	<input checked="" type="checkbox"/>				
Memorization-informed FID (MiFID)	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		
Unbiased FID and IS	<input checked="" type="checkbox"/>				
Clean FID	<input checked="" type="checkbox"/>				
Fre'chet Video Distance (FVD)	<input checked="" type="checkbox"/>				
Methods based on Self-supervised Learned Representations					
Methods based on Analysing Data Manifold					
Local Intrinsic Dimensionality (LID)	<input checked="" type="checkbox"/>				
Intrinsic Multi-scale Distance (IMD)	<input checked="" type="checkbox"/>				
Perceptual Path Length (PPL)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	
Linear Separability in Latent Space	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>	
Classification Accuracy Score (CAS)	<input checked="" type="checkbox"/>				
Non-Parametric Tests to Detect Data-Copying	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		
Measures that Probe Generalization					
New Ideas based on Precision and Recall (P&R)					
Density and Coverage	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		
Alpha Precision and Recall	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		
Duality GAP Metric	<input checked="" type="checkbox"/>				
Spectral Methods	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	
Caption Score (CapS)	<input checked="" type="checkbox"/>				
Human Eye Perceptual Evaluation (HYPE)	<input checked="" type="checkbox"/>				
Neuroscore		<input checked="" type="checkbox"/>			
GAN Steerability & Dissection		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	
A Universal Fake vs. Real Detector	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>	

Figure 24: A summary of evaluation measures covered in this work.



해당 논문의 의견(Quantitative뿐만 아니라 Qualitative도 고려해야 한다)

1. 기존에는 단순히 얼굴을 기준으로만 평가했으며, 다양한 케이스에 대한 평가 지표를 깊이 있게 다루지 않았다.
2. Quality and diversity에 대한 관점은 많지만, 차원(Generalization, fairness)부분에 대한 연구가 적다.
 - a. Generalization 관점에서는 생성 모델을 더 깊게 이해할 수 있게 된다.
 - b. Fairness 관점에서는 올바른 사회적 영향에 대해서 다룰 수 있으며, 모델 배포로 인해 발생하는 잠재적 위험 요소를 완화할 수 있습니다.
3. Downstream Task마다 중요한 평가 요소가 다르다.
 - a. 예를 들어, 이미지의 퀄리티가 중요한 반면 다른 상황에서는 Diversity가 확보되는 것이 중요하다.(두 관계는 Trade-off 관계입니다)
4. 좋은 평가 지표란 모델의 순위를 매길 수 있는 측면도 있지만, 도메인(e.g. 의료 이미지) 별로 적절한 평가 지표가 유의미한 지이다.
5. 생성 모델이 학습 데이터를 기억하는 정도는 여전히 명확하지 않습니다.
6. 궁극적으로, 딥 페이크와 같은 사회적 이슈를 다루기 위한 평가 지표를 고려해야 합니다.

5. 추가 사항

- 추가적으로 생성 모델은 **VAE**, **GAN**, **Diffusion** 모델로 크게 3가지 주제로 나뉘며, 아래의 그림을 통해 해당 3가지 장단점을 이해하실 수 있습니다.

▼ **VAE, GAN, Diffusion** 모델

- [2022] [Tackling the Generative Learning Trilemma with Denoising Diffusion GANs](#)

[Tackling the Generative Learning Trilemma with Denoising Diffusion GANs.pdf](#)

- 제가 아는 선에서 최근 이슈에 대해서 추가적으로 브리핑 드리자면,

▼ 올해 초 **Diffusion** 모델의 Sampling 속도를 보완한 논문도 추가적으로 나왔습니다.

- 사실상 현재 생성 모델들 중 가장 성능이 좋음을 알 수 있습니다.
- 생성 모델에 관심이 있으신 분들은 참고하시면 좋을 것 같습니다.
- [2015] [Deep Unsupervised Learning using Nonequilibrium Thermodynamics](#)
 - Diffusion의 시작 논문
- [2021] [Diffusion Models Beat GANs on Image Synthesis](#)

[Diffusion Models Beat GANs on Image Synthesis.pdf](#)

- [2023] [Consistency Models](#)

- 샘플링 속도를 보완한 논문

[Consistency Models.pdf](#)

<https://github.com/CompVis/stable-diffusion>

<https://github.com/Stability-AI/stablediffusion>

https://github.com/openai/consistency_models

- Diffusion에 관심이 많다면?(Discord에 업데이트 알림도 해줍니다!)
 - https://github.com/kwonminki/One-sentence_Diffusion_summary 개인적으로 강화

▼ **GAN** 도 후속 연구가 많으니, 토글을 참고하시면 좋을 것 같습니다!

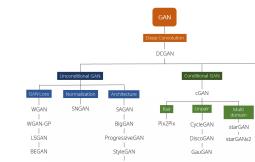
▼ GAN 관련 추천 링크

- <https://github.com/hollobit/All-About-the-GAN>
- <https://github.com/soumith/ganhacks>
- <https://github.com/nightrome/really-awesome-gan>

GitHub - eriklindernoren/PyTorch-GAN: PyTorch implementations of Generative Adversarial Networks. - GitHub - eriklindernoren/PyTorch-GAN: PyTorch implementations of Generative Adversarial Networks. - GitHub - eriklindernoren/PyTorch-GAN: PyTorch implementations of Generative Adversarial Networks.

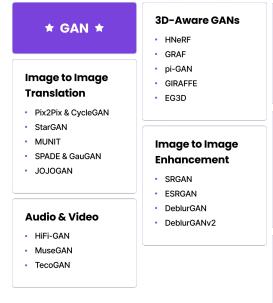
<https://github.com/eriklindernoren/PyTorch-GAN#gan>

- 기술적 분류



<https://github.com/dongb5/GAN-Timeline>

- 분야별 분류



- 결론: 이미지 생성 모델 중에서는 현재 Diffusion이 가장 좋다?