23.07.13 - 5주차

딥러닝 논문 요약 및 구현 스터디

발표자

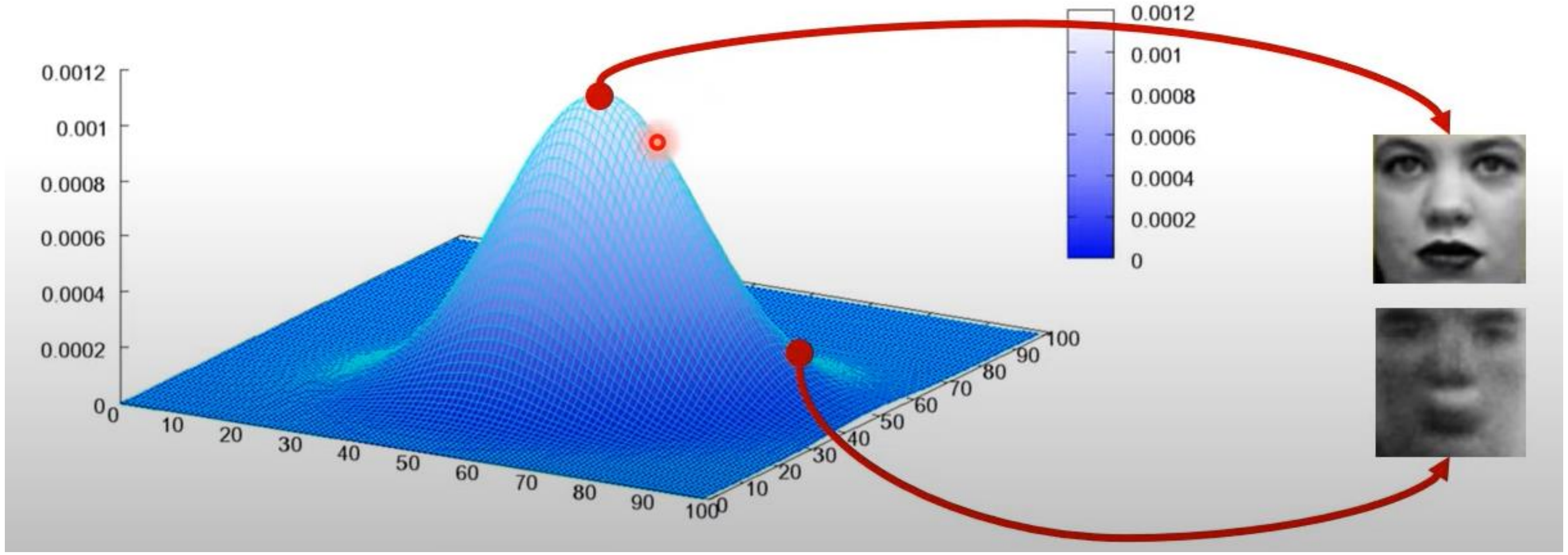
김정택 (연구원)

GAN:

Generative Adversarial Nets

Abstract

- **"존재하지 않는" "그럴싸한" 이미지**를 생성하는 모델
- A Statistical Model of **the Joint Probability Distribution**
- An Architecture to **generate new data instances**
- Adversarial Model (**Generative Model V.S. Determinative Model**)



1. Intro

- Deep Generative Model 의 근황
    - 최대 우도 추정과 같은 확률 근사 계산의 어려움, Leveraging the benefits of piecewise linear units 의 어려움으로 인해 생성 모델이 만들어지기 어려웠다.

- Adversarial Nets Framework
    - 생성 모델(G) 는 위조 통화를 생성하고, 판별 모델(D)는 위조 통화인지 여부를 판별합니다. MinMax 게임을 통해 두 모델은 위조 통화가 없을 때 까지 본인의 모델을 개선하게 됩니다.

2. Related

- RBMs (Restricted Boltzmann Machines)
  - 확률 변수의 모든 state 에 대한 전역 합산/적분으로 정규화
  - Markov chain Monte Carlo 방식을 사용하여 추정할 수 있으나 Mixing이 큰 문제가 된다.

- DBNs (Deep Belief Networks)
  - 하나의 undirected layers와 몇 개의 다른 directed layers를 포함한 하이브리드 모델
  - 계산에 대한 어려움이 있다.

- NCE (Noise-Constractive Estimation) or Score Matching
  - 로그우도를 근사하거나 대안하는 방식
  - 학습된 확률 밀도에 대해 정규화 상수를 포함해 분석적으로 지정해야하지만 이는 쉽지 않은 작업입니다.

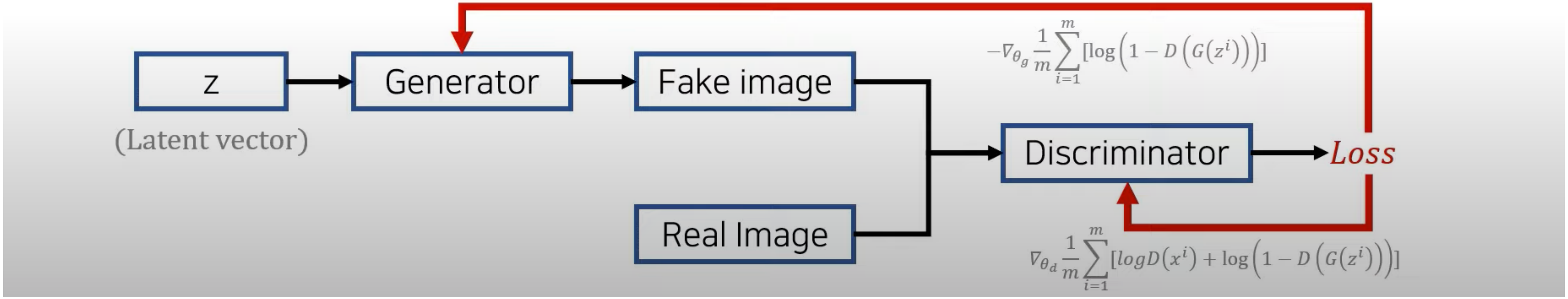
2. Related

- GSN (Generative Stochastic Network Framework)
  - 역전파로 훈련시킬 수 있도록 설계됨
  - 샘플링을 위해 마르코프 체인 활용
  - Feedback Loops로 인해 무제한 활성화 문제가 발생할 수 있다.

3. Adversarial Nets

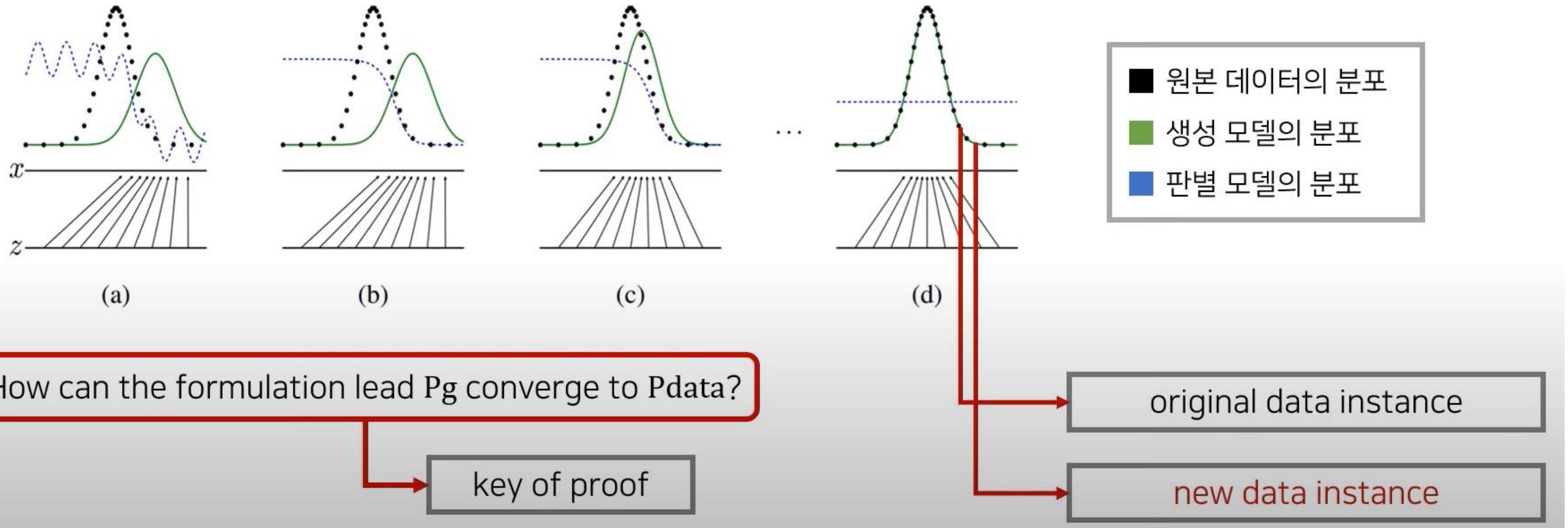
$$\min_G \max_D V(D,G) = E_{x \sim p_{data}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

Generator	$G(z)$: *new data instance*
Discriminator	$D(x) = Probability$: *a sample came from the real distribution* ($Real$: 1 ~ $Fake$: 0)



3. Adversarial Nets

- 공식의 목표(Goal of Formulation)
  - $Pg \rightarrow Pdata, D(G(z)) \rightarrow 1/2$ $(G(z)$ *is not distinguishable by* $D)$



- How can the formulation lead Pg converge to Pdata?

key of proof

4. Theoretical Results

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

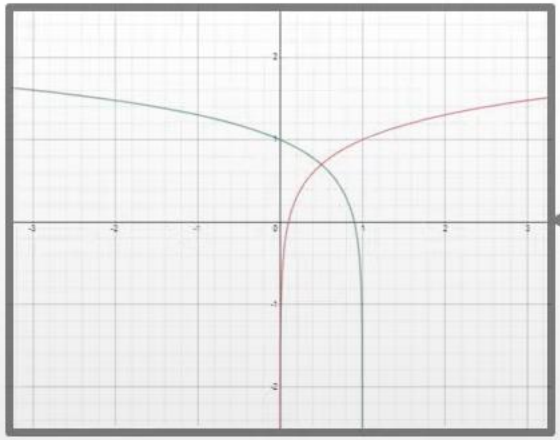
Proof: For G fixed,

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

$$= \int_x p_{data}(x)\log(D(x))\,dx + \int_z p_z(z)\log(1 - D(g(z)))\,dz$$

$$E[X] = \int_{-\infty}^{\infty} xf(x)dx$$

$$= \int_x p_{data}(x)\log(D(x)) + p_g(x)\log(1 - D(x))\,dx$$



$$\boxed{function\ y \rightarrow alog(y) + blog(1-y)\ achieves\ its\ maximum\ in\ [0,1]\ at\ \frac{a}{a+b}}$$

same as ***optimal control***: $\frac{\delta V(G,D)}{\delta D}[D^*(x)] = 0$

4. Theoretical Results

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

Proof:

$$C(G) = \max_D V(G, D) = E_{x \sim p_{data}(x)}[log D^*(x)] + E_{Z \sim p_z(z)}[\log(1 - D^*(G(z)))]$$

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

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

removed when $p_g = p_{data}$

$$KL(p_{data}||p_g) = \int_{-\infty}^{\infty} p_{data}(x) \log\left(\frac{p_{data}(x)}{p_g(x)}\right) dx$$

$$= \boxed{E_{x \sim p_{data}(x)}\left[log \frac{2 * p_{data}(x)}{p_{data}(x) + p_g(x)}\right] + E_{x \sim p_g(x)}\left[log \frac{2 * p_g(x)}{p_{data}(x) + p_g(x)}\right]} - \log(4)$$

$$= KL(p_{data}|| \frac{p_{data}(x) + p_g(x)}{2})) + KL(p_g|| \frac{p_{data}(x) + p_g(x)}{2}) - \log(4)$$

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

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

4. Theoretical Results

for the 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 $p_{data}(x)$.

Update the discriminator by ascending its stochastic gradient:

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

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:

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

end for

The gradient-based updates can use any standard gradient-based learning rule. They used momentum.

- Discriminator : Ascending (Fake Image & Original Image)
- Generator : Descending (Fake Image 에 대해)

5. Experiment

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


a) b) c) d)

- Random 하게 만든 이미지
- 단순 암기가 아님