

메뉴바를 클릭하면
계정을 관리할 수 있어요

카테고리 없음

[Euron] Week3_ Generative Adversarial Nets

yejji 2025. 3. 24. 15:48

0. Abstract

본 논문은 기존에 사용되던 일반적인 모델 대신에 적대적 과정(adversial process)을 통해 추정하는 새로운 틀을 제안한다.

생성자 G는 데이터의 분포를 포착하는 역할을,
판별자 D는 샘플이 생성자 G보다 원래의 train data로부터 유래될 가능성을 추정하는 역할을 한다.

생성자 G는 판별자 D가 오류를 생성할 가능성을 최대화하고자 하는데,
이러한 과정은 minimax two-player game과 비슷한 개념을 가진다.

임의의 G와 D에서 G가 실제 train data를 완벽히 학습하여 동일한 분포의 데이터를 생성해 내고,
이에 대해 D가 모든 데이터 입력값에 대해 1/2 확률을 출력할 경우 유일한 해가 존재한다고 말을 한다.

G와 D가 다중 퍼셉트론으로 정의된 경우, 전체 시스템은 역전파 과정 즉, 오차값을 출력층에서 입력층 방향인 역방향으로 전파하고 가중치를 업데이트하는 과정으로 훈련이 가능하다.

메뉴바를 클릭하면
계정을 관리할 수 있어요

이런 방법으로 훈련시킬 경우, train 과정과 샘플 일반화 과정에서 Markov chains와 근사 추론 네트워크(unrolled approximate inference net-works)를 사용할 필요성이 없어진다.

본 논문의 연구에서는 일반화된 샘플들을 질적, 양적으로 평가하여 제시된 틀의 잠재력을 증명할 수 있게 된다.

1. Introduction

Deep learning의 목적은 인공 지능 분야에서 다양한 데이터들의 확률 분포, 즉 데이터가 특정 값이나 패턴을 가질 확률을 표현하는 풍부하고 계층적인 모델을 찾는 것이다.

예를 들면, natural images, audio waveforms containing speech, 자연어 분야의 symbols 등이 있다.

지금까지, Deep learning 분야에서 가장 눈에 띄게 보이는 성공은 고차원 센서 입력값을 class label로 맵핑하는 판별 모델을 구현한 것이다.

이런 성공들은 주로 역전파, dropout 알고리즘에 기반을 두는데, 특히 잘 작용하는 gradient 값을 가진 PLU를 사용한다.

(다양한 구간에서 선형 활성화 함수를 사용한다는 것이다.)

Deep generative model은 최대 가중치를 추정하고, 이와 관련하여 유발되는 많은 확률적 수학적 연산 과정과 PLU의 이점들을 최대한 활용하기에 어렵다는 이유로 그 영향이 적다.

위와 같은 단점들을 피하고자 새로운 **generator model**의 추정 과정을 제안한다.

제안된 적대적 네트워크에서, **생성자는 판별자와 적대적으로 경쟁한다**. 이때, 판별자는 샘플이 모델이나 원래의 데이터 분포로부터 유래되었는지 판단하는 과정을 학습한다.

메뉴바를 클릭하면
계정을 관리할 수 있어요

생성자는 위조 화폐를 생산하고, 들키지 않고 사용하려고 하는 도둑, 판별자 폐를 적발하고자 하는 경찰이라고 비유한다.

도둑과 경찰 게임에서 경쟁은 진짜 지폐로부터 위조 지폐가 구분 불가능해질 때까지 생성자와 판별자를 학습시킨다.

이러한 프레임워크는 많은 종류의 모델과 최적화 알고리즘에 새로운 특이한 훈련 과정을 만들어낸다.

본 논문에서는 생성자가 다중 퍼셉트론을 통해 랜덤 노이즈를 거쳐 샘플들을 만들어 내고, 판별자가 역시나 다중 퍼셉트론일 경우를 연구한다.

본 논문에서 이러한 특별한 경우를 적대적 신경망으로 명칭한다.

이 경우 오로지 매우 성공적인 역전파, dropout 알고리즘과 순전파 알고리즘만을 사용하여 생성자와 판별자 모델을 학습시킨다.

따라서, Markov chains와 approximate inference 과정은 필요하지 않다.

2. Related work

잠재 변수 즉, 관측이 안된 숨겨진 변수를 포함한 방향 그래프 모델의 대안은 RBM과 DBM과 같이 잠재 변수를 포함한 비방향 그래프 모델이다.

RBM과 DBM 모델들은 서로 상호작용하는데, 일반화되지 않은 잠재 변수들과 랜덤 변수들의 모든 상태를 포함하여 일반화된 산물로 표현된다.

비록 MCMC 방식에 의해 추정될 수는 있을지라도,

그 양과 그래디언트값은 가장 사소한 경우를 제외하고는 그 나머지 모든 경우들에 대해 계산될 수는 없다.

Mixing은 MCMC에 기반한 학습 알고리즘의 중요한 문제점들을 보여준다.

메뉴바를 클릭하면
계정을 관리할 수 있어요

DBN은 1개의 방향성이 없는 층과 여러 방향성이 있는 층들을 포함한 hybrid 모델이다. 빠른 근사 계층 훈련의 기준값이 존재할지라고, DBN은 비방향성, 방향성 모델들이 합쳐져 만들어지는 계산적 어려움을 초래한다.

로그 가능성을 근사하거나 뛰어넘지 않는 대안의 기준이 score matching과 NCE로 새롭게 제안되었다. 2가지 방법 모두 일반화 과정으로 분석적 특성화가 가능한 학습된 확률 밀도값이 필요하다.

DBN과 DBM과 같이 잠재 변수로 구성된 여러 개의 층을 가진 많은 생성자들이 풀 수 있는 비일반화된 확률 밀도값을 도출해낼 수는 없다.

NCE에서 이전과 차별화된 학습 규칙이 생성자에 적용된다. 그러나 분리된 차별화된 모델을 fitting하는 것보다는, 생성자가 스스로 노이즈 분포로부터 sample들의 생성된 데이터를 차별화하는 것이 목적이다.

이는 NCE가 특정 노이즈 분포를 사용하기 때문인데, 이로 인해 모델이 작은 변수 집합으로부터 대략적으로나마 옳은 분포가 무엇인지를 학습한 이후에 모델의 학습 속도가 급격히 느려진다.

최종적으로, 몇몇 기술은 구체적인 확률 분포를 정의할 수 없고, 오히려 generative model을 훈련시켜 원하는 분포에서 샘플값들을 추출해낼 수 있게 된다.

이 분야에서 저명한 최근의 연구는 GSN 프레임워크를 포함하고, 일반화된 denoising 오토인코더를 확장시킨 매개변수화된 Markov chains를 정의한다.

이런 GSN과 비교하였을 때, 적대적 신경망은 sampling 과정에서 markov chains가 필요하지 않다.

적대적 신경망이 일반화 과정에서 피드백 루프를 요구하지 않기 때문에, 다시 말하면 이전 상태의 생성결과가 이후 결과에 직접적인 영향 없이 즉시 업데이트되기 때문에, 데이터 조각별로 선형 단위에 더 많은 영향을 끼친다. 이로 인해 역전파 과정이 더 느릴 정도로도 향상되지만, 여전히 피드백 루프에서 사용될 때 무한 활성화와 관련이 존재한다.

메뉴바를 클릭하면
계정을 관리할 수 있어요

역전파로 generative model을 훈련시키는 더 최근의 연구들은 auto-encoding variational Bayes와 확률적인 역전파 개념을 포함한다.

3. Adversial nets

적대적 모델 프레임워크는 모델들이 모두 다중 퍼셉트론일 때 적용하기가 간단하다.

p_g 라는 생성자의 분포를 학습하기 위해, input 노이즈 변수들을

$p_z(z)$ 로 정의하고,

$G(z; \theta_g)$ 로서 데이터 공간으로의 맵핑 과정을 나타낸다.

이때, G는

θ_g 파라미터로 다중 퍼셉트론에 의해 표현되는 미분 가능한 함수이다.

또한, 2번째 다중 퍼셉트론을

$D(x; \theta_d)$ 라고 정의하는데, 이 퍼셉트론은 output값으로 단일 스칼라값을 출력한다.

D(x)는 x가 p_g 가 아닌 데이터로부터 나올 확률을 나타낸다.

본 연구에서는 D를 train예시와 G로부터의 샘플 모두에 맞는 label값을 할당하는 가능성을 최대화하기 위해 훈련시킨다.

값은 이와 동시에 최소화하기 위해 똑같이 훈련시킨다.

$$\log(1 - D(G(z)))$$

다시 말하면, D와 G는 가치함수 $V(G, D)$ 로 two-player minimax game을 하는 하다.

메뉴바를 클릭하면
계정을 관리할 수 있어요

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

Equation 1

다음 섹션에서, 적대적 신경망의 이론적인 분석본을 제시하고, 특히나 train 기준이 파라미터가 없다는 제한 아래 G와 D에 충분한 용량이 주어질 경우 데이터의 생성 분포를 복구해낼 수 있도록 하였다.

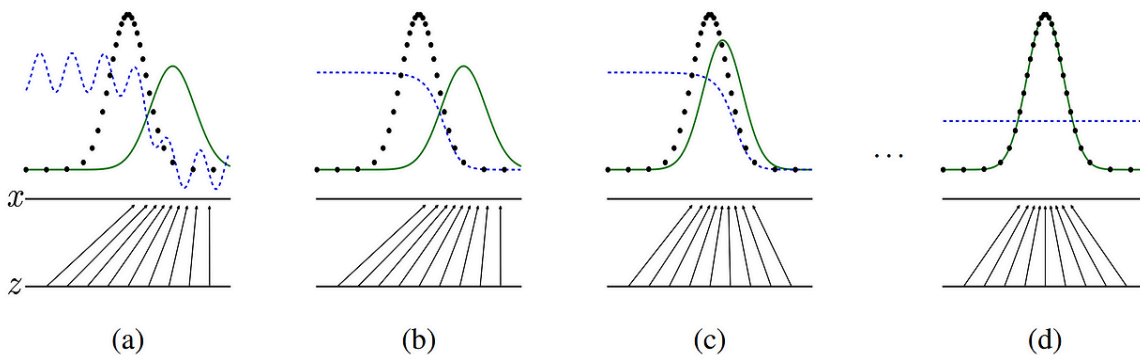


Figure 1 : GAN의 학습 과정에서 G와 D의 변화 (검정색 점선 : 실제 데이터 분포 / 초록색 실선 : G의 데이터 분포 / 파란색 점선 : D의 확률 출력값)

바로 위의 Figure 1 그림을 보게 되면 이런 접근법에 대해 더 잘 이해 가능하다.

(a) : 학습 초반으로, 생성 데이터와 실제 데이터와 눈에 띄게 달라, 판별자가 진짜와 가짜 데이터를 쉽게 구분할 수 있다.

(b) : 판별자 D가 학습하는 과정으로, 판별자가 더 잘 구분할 수 있도록 최적화하는 과정이다. 이에 대응하여 생성자 G는 경쟁적으로 더 정교한 가짜값을 생성해낸다.

(c) : 생성자 G를 업데이트하는 과정으로, 점점 생성자 G가 생성한 분포 p_g 가 실제 데이터 분포인 p_x 에 가까워진다.

즉, G가 실제값과 매우 비슷한 가짜값을 점점 생성한다는 것이다.

(d) : 학습이 완료된 GAN의 최적점으로, p_g 와 p_x 가 거의 똑같아 판별자 D가 더이상 구분 불가능한 상태이다.

이때, 판별자 D가 더이상 진짜와 가짜를 구분하지 못한다는 것을 **50% 확률로 무**다.

다시 말해, 제대로 구분하지 못 한다는 것을 의미한다.

메뉴바를 클릭하면
계정을 관리할 수 있어요

실험에서, 반복적이고 수치적인 접근법을 사용하였다.

Training의 내부 loop에서 D를 최적화하는 것은 계산적으로 금지되고, 유한한 데이터셋에서 과적합을 유발한다.

그 대신에, D를 최적화하는 k개의 단계들, G를 최적화하는 1개의 단계를 번갈아서 진행한다.

이는 D가 최적의 해답과 비슷한 값으로 유지되도록 하고, G는 충분히 느리게 바뀐다.

이러한 전략은 SML/PCD training이 학습의 내부 루프의 한 부분으로서 Markov chains에서의 연쇄적인 연소를 피하기 위해,

학습의 한 단계에서 다음 단계로 Markov Chains로부터 샘플값을 유지할수 있다.

본 연구에서, equation 1은 G가 더 잘 학습하도록 충분한 그래디언트값을 제공하지는 못 한다.

학습 초기에 G의 학습이 충분치 않을 때, D는 높은 신뢰도값을 가진 샘플들을 거부할 수 있다. 그 샘플들이 training 데이터값과 꽤나 다를 것이기 때문이다.

이러한 경우,

$\log(1 - D(G(z)))$ 는 포화 상태, 즉 그 값이 최대가 된다.

이와 반대로,

$\log(1 - D(G(z)))$ 값을 최소화하기 위해가 아니라

$\log D(G(z))$ 값을 최대화시키도록 훈련시킬 수 있다.

(두 경우는 잘 보면 똑같다.)

이 objective function은 G와 D에서 같은 fixed point를 유래하지만, 학습 과정
한 그래디언트값을 제공한다.

메뉴바를 클릭하면
계정을 관리할 수 있어요

4. Theoretical Results

생성자 G는 z 가 p_z 에 거의 비슷해질 때 얻은 $G(z)$ 의 분포로서 p_g 확률 분포를 가진다.

그러므로, Algorithm 1이 충분한 용량과 training 시간이 주어진 경우 P_{data} 의 옳은 추정치
값으로 수렴한다.

파라미터가 없는 세팅 아래 확률밀도함수의 공간에서 수렴되는 것을 학습함으로서 무한한 용
량을 가진 모델을 새롭게 제시한다.

Section 4.1 에서 이 minimax game이 $p_g = p_{data}$ 라는 최적값을 가진다는 것을 보여준다.
Section 4.2 에서는 algorithm 1이 equation 1을 최적화하고 바람직한 결과값을 얻는 것도
보여줄 것이다.

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 $p_{\text{data}}(x)$.
- Update the discriminator by ascending its stochastic gradient:

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

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:

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

end for

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

메뉴바를 클릭하면
계정을 관리할 수 있어요

Algorithm 1 : GAN의 minibatch 확률적 경사 하강법

판별자 D는 k번 업데이트, 생성자 G는 판별자 업데이트 이후에 1번 업데이트한다. 훈련 반복 횟수만큼 실행한다.

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

판별자 업데이트 과정에서, $\log D(\sim)$ 부분은 진짜 데이터 x_i 에 대해 판별자가 1이라고 예측할 확률을 의미하고,

$\log(\sim)$ 부분은 가짜 데이터 $G(z_i)$ 에 대해 판별자가 0이라고 예측할 확률을 의미한다.

판별자는 위 식의 값을 증가시키는 방향, 즉 예측을 잘 하게 하는 방향으로 훈련시킨다.

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

생성자는 위 식의 값을 감소시키는 방향, 즉 $G(z)$ 가 더 진짜같은 데이터를 생성하도록 하는 방향으로 훈련시킨다.

4.1 Global Optimality

주어진 생성자 G에 대한 최적의 판별자 D를 먼저 고려하고자 한다.

메뉴바를 클릭하면
계정을 관리할 수 있어요

Proof 1 : G가 고정된 경우 최적의 판별자 D는 다음과 같다.

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

증명: 생성자 G에 대해 판별자 D에 대한 training 기준은 $V(G, D)$ 를 최대화하는 것이 목적이다.

$$\begin{aligned} V(G, D) &= \int_{\mathbf{x}} p_{data}(\mathbf{x}) \log(D(\mathbf{x})) d\mathbf{x} + \int_{\mathbf{z}} p_{\mathbf{z}}(\mathbf{z}) \log(1 - D(g(\mathbf{z}))) d\mathbf{z} \\ &= \int_{\mathbf{x}} p_{data}(\mathbf{x}) \log(D(\mathbf{x})) + p_g(\mathbf{x}) \log(1 - D(\mathbf{x})) d\mathbf{x} \end{aligned}$$

라는 식을

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

에 대입하였다. 최적화하는 과정에서

$$\mathbb{E}_{\mathbf{x} \sim p_{data}} \left[\log \frac{p_{data}(\mathbf{x})}{p_{data}(\mathbf{x}) + p_g(\mathbf{x})} \right] + \mathbb{E}_{\mathbf{x} \sim p_g} \left[\log \frac{p_g(\mathbf{x})}{p_{data}(\mathbf{x}) + p_g(\mathbf{x})} \right]$$

라는 결론을 도출해낼 수 있었다.

Theorem 1. *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 $-\log 4$.*

Proof. For $p_g = p_{data}$, $D_G^*(x) = \frac{1}{2}$, (consider Eq. 2). Hence, by inspecting Eq. 4 at $D_G^*(x) = \frac{1}{2}$, we find $C(G) = \log \frac{1}{2} + \log \frac{1}{2} = -\log 4$. To see that this is the best possible value of $C(G)$ only for $p_g = p_{data}$, observe that

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

and that by subtracting this expression from $C(G) = V(D_G^*, G)$, we obtain:

$$C(G) = -\log(4) + KL \left(p_{data} \left\| \frac{p_{data} + p_g}{2} \right\| \right) + KL \left(p_g \left\| \frac{p_{data} + p_g}{2} \right\| \right) \quad (5)$$

where KL is the Kullback–Leibler divergence. We recognize in the previous expression the Jensen–Shannon divergence between the model’s distribution and the data generating process:

$$C(G) = -\log(4) + 2 \cdot JSD(p_{data} \| p_g) \quad (6)$$

Equation 2에 의해 판별자가 진짜와 가짜를 제대로 구분해내지 못할 때, 확률값으로 1/2를 가진다.

$$C(G) = -\log(4) + KL \left(p_{data} \left\| \frac{p_{data} + p_g}{2} \right\| \right) + KL \left(p_g \left\| \frac{p_{data} + p_g}{2} \right\| \right)$$

Equation 5

에서 중간 항은 실제 데이터와 실제 데이터와 생성 분포의 평균적인 분포의 차이를 의미하며, 마지막 항은 생성 분포와 실제 데이터와 생성 분포의 평균적인 분포의 차이를 의미한다.

이는 모델의 실제 분포와 데이터 생성 과정 간의 JSD로 표현할 수 있는데, 다음과 같다.

$$C(G) = -\log(4) + 2 \cdot JSD(p_{data} \| p_g)$$

Equation 6

두 확률 변수에 대한 JSD가 항상 음수값이 절대 아니고, 두 확률변수가 같을 때에만 값으로 0을 가지기 때문에

$C(G)$ 함수의 최솟값은 $-\log(4)$ 이다.

이는 생성 모델이 실제 데이터를 완전히 대체했을 때 유일하게 가지는 값이다.

4.2 Convergence of Algorithm 1

Proposition 2. G와 D가 충분한 용량을 가진 경우, Algorithm 1의 각 단계에서 D는 주어진 최적의 G에 할당되고,
원래의 기준을 향상시키기 위해 업데이트된다.

메뉴바를 클릭하면
계정을 관리할 수 있어요

Proof.

$V(G, D) = U(p_g, D)$ 를 가정한다.

G와 D가 충분한 용량을 가지고 있는 경우, Algorithm 1의 각 단계에서 판별자 D는 주어진 생성자 G의 최적값에 도달하고,

p_g 는 원래의 기준을 향상시키는 방향으로 업데이트가 일어난다.

convex function의 상한값에 대한 미분은 최댓값에 도달하는 지점에서의 미분 과정을 포함한다.

최적의 판별자 D가 주어진 상태에서 적절한 생성자 G가 주어지면

p_g 값을 업데이트할 때 gradient descent를 적용할 수 있다.

본 연구에서는 적대적 신경망이

$G(z; \theta_g)$ 함수를 통해

θ_g 를 최적화한다.

이와 같이, G를 정의하고자 다중 퍼셉트론을 사용하는 것은 파라미터 공간에서 여러 중요한 점들을 도입한다는 것이다.

그러나, 다중 퍼셉트론이 최고의 수행을 보이는 것은 이론적 보장의 부족에도 불구하고 사용하기에 합리적인 모델이라는 것을 내포한다.

5. Experiments

MNIST, the Toronto Face Database, CIFAR-10 데이터셋들로 적대적 신경망 GAN을 훈련시켰다.

생성자는 ReLU와 Sigmoid 활성화 함수가 섞인 형태를 사용하나, 판별자는 Max 함수를 사용한다.

메뉴바를 클릭하면
계정을 관리할 수 있어요

Dropout도 판별자를 훈련시키는 데에 사용된다.

본 실험의 이론적 프레임워크가 Dropout과 다른 noise들을 생성자의 중간층에서 사용하나, 실제 실험에서는 생성자의 하위층의 Input값으로서 noise를 사용한다.

p_g 가 성립한다는 하에 test 데이터의 가능성을 G에 의해 생성된 샘플들에 Gaussian Parzen window를 fitting 시키고, 이러한 분포 아래 log 가능성을 사용하여 추정한다.

σ 파라미터는 Gaussian 분포에서 검증 데이터셋을 cross로 검증할 때 얻어진다. 이러한 과정은 Breuleux et al에서 소개되었고, 정확한 계산이 불가능한 다양한 생성 모델들에서 정확한 가능성값을 계산하는 데에 사용된다.

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

Table 1에서 결과 확인이 가능하다.

가능성을 추정하는 이러한 방식은 큰 분산값을 가진 고차원 공간에서는 잘 수행하지 못하지만, 현재로서는 이용하기에 가장 좋은 방법이다.

가능성을 추정하지 않고 샘플화하는 발전된 생성 모델의 경우 모델을 어떻게 평가할지에 대해 더 많은 연구를 하도록 동기를 부여한다.

메뉴바를 클릭하면
계정을 관리할 수 있어요

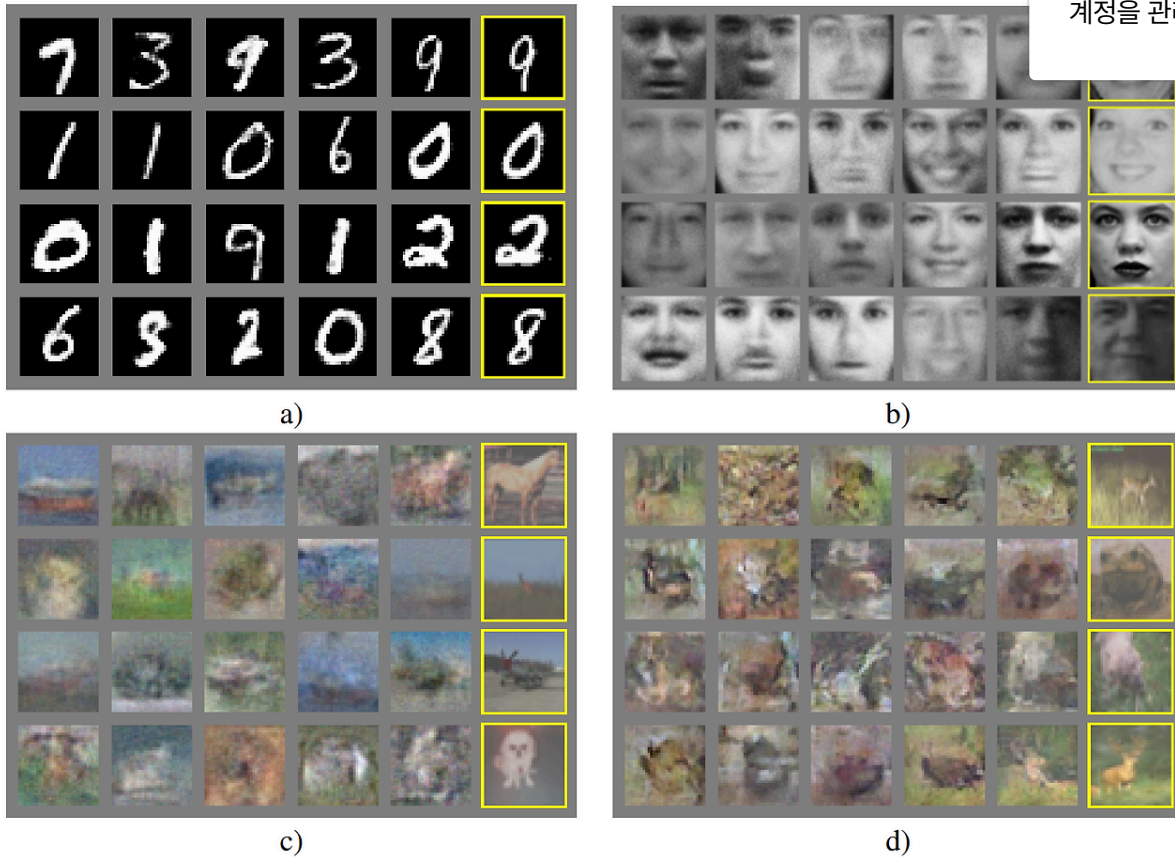


Figure 2 : 모델이 생성한 샘플 시각화 부분

Figure 2에서 yellow box 부분은 모델이 생성한 샘플과 가장 유사한 실제 학습 데이터로, 생성 모델이 단순히 학습1 데이터를 복사하는 것이 아니라, 새로운 데이터를 실제로 생성해낸다는 것이다.

(a) : MNIST(은글씨로 된 숫자)

(b) : TFD 얼굴 이미지

(c) : CIFAR-10(fully connected model) : 해상도가 낮고 흐릿하다. 이는 완전 연결 모델을 사용했기 때문이다.

(d) : CIFAR-10(CNN 기반 생성 모델) : 해상도가 (c)보다는 좋으나 아직까지 낮고 흐릿하다.



Figure 3 : 생성 모델의 잠재 공간

Figure 3은 모델에서 선형 보간을 수행한 형태이다.

보간이란 두 잠재 벡터 사이에서 점진적으로 변화를 주면서 새로운 샘플값을 생성하는 것이다.

메뉴바를 클릭하면
계정을 관리할 수 있어요

1이 5에서 7 모양을 거쳐 자연스럽게 /(슬래시 형태)로 변화한다는 것을 확인할 수
이는 모델이 데이터의 근본적인 구조를 잘 학습한다는 것이다.

위의 Figure 2와 Figure 3에서 훈련 이후 생성자로부터 그려진 샘플들을 보여주고 있다.

이 샘플들이 존재하는 방식에 의해 생성된 샘플들보다 더 낫다고 확실히 말할 수는 없으나,
이 샘플들이 더 나은 생성 모델로서 어느 정도는 경쟁력이 있고 적대적 생성모델이 잠재력이
있다는 점을 잘 보여준다.

6. Advantages and disadvantages

새로운 프레임워크는 이전의 모델들과 비교했을 때 장점과 단점을 각각 가진다.

먼저 단점은

$p_g(\mathbf{x})$ 에 대한 명시적 표현이 없고, D가 훈련 동안에 G와 동시에 작용해야 한다는 것이다.

특히, D를 업데이트하지 않고 G만 과도하게 훈련되면 안된다.

이 경우, Helvetica scenario가 발생하는데, 이는 G가 너무 많은 z값을 동일한 x값으로 맵핑
하여

p_{data} 라는 원래 데이터 분포를 충분히 잘 표현해낼 수 없다는 것이다.

이 부분은 볼츠만 머신에서 학습 단계마다 음의 체인값이 최신 상태로 유지되어야 한다는 점
과 유사하다.

이에 반해 장점은

Markov Chains이 절대로 필요하지 않고, 그래디언트값을 얻기 위해 backprop를 사용하고,
학습 과정 동안에 어떤 추론도 필요하지 않다는 것이다.

이로 인해 다양한 형태의 함수들이 모델에 적용될 수 있다.

바로 아래의 Table 2가 다른 생성 모델들과 적대적 신경망을 비교한 결과값을 요약

메뉴바를 클릭하면
계정을 관리할 수 있어요

	Deep directed graphical models	Deep undirected graphical models	Generative autoencoders	Adversarial
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 : 다양한 생성 모델의 학습, 평가 난이도 평가 결과

앞서 말한 이점들은 주로 계산 과정을 요구한다.

적대적 신경망들은 생성자가 데이터 예시들로 직접 업데이트되는 것이 아니라, 판별자에 의해 그래디언트값이 변한다는 점에서 통계적으로 이점을 가진다.

즉, Input의 구성들이 생성자의 파라미터값으로 그대로 사용되는 것이 아니라는 것이다.

적대적 신경망의 또다른 이점은 Markov Chains를 활용한 방식의 경우 Chain에 대해 blurry한 분포를 요구하는데 반해, 적대적 신경망은 매우 날카롭게 비선형적인 분포를 잘 표현해낼 수 있다는 것이다.

7. Conclusions and future work

본 논문에서 제시한 GAN 프레임워크는 다양한 방향으로 확대될 수 있다.

$p(\mathbf{x} | \mathbf{c})$ 라는 조건부 생성 모델이 G와 D 모두에 Input값으로 \mathbf{c} 를 추가함으로써 얻을 수 있다.

메뉴바를 클릭하면
계정을 관리할 수 있어요

또, 학습된 근사 추론은 보조 신경망을 훈련시켜 \mathbf{x} 가 주어졌을 때 \mathbf{z} 를 예측하도록 다.

이는 생성 모델이 훈련이 끝난 이후에 정해진 생성 신경망을 기반으로 별도로 더 훈련시킬 수 있다는 이점이 있다.

또, 부분적인 정보만으로도 전체 데이터를 예측 가능하다.

즉, 적대적 신경망을 사용하여 결정적 MP-DBM의 확률적 확장을 구현해낼 수 있다는 것이다.

부분적인 입력 정보만으로도 나머지를 예측하는 모델을 학습하는 방식으로, GAN 모델을 좀 더 확장시켜 확률적 예측이 가능하다.

또, 반지도 학습을 통해 성능을 더 개선시킬 수 있다.

판별기 또는 추론 네트워크에서 추출한 특징들을 활용하면, label이 제한된 데이터에서도 판별자의 성능을 향상 가능하다.

적은 라벨 데이터값을 가지더라도 충분히 효과적인 학습이 가능해진다.

마지막으로, 학습 과정을 최적화함으로써 훈련 속도가 향상된다.

G와 D를 학습할 때 더 효과적으로 조율하거나,

훈련 과정 동안 잠재 변수 \mathbf{z} 를 샘플링할 더 나은 분포를 결정함으로써 훈련 속도를 크게 향상
이 가능하다.

[의문점]

1. 왜 GAN모델은 생성자는 1번 학습시키는 반해, 판별자는 여러 번 학습시키는 것인가?

공감