Generative Adversarial Networks(GAN)

Generative Adversarial Networks(

Abstract

적대적 과정(서로 경쟁하며 학습)을 통해 생성 모델을 추정함.
두 모델을 동시에 훈련함.

- 데이터 분포를 포착하는 생성자 G(Generative model)
- 샘플이 G가 아닌 훈련 데이터로부터 왔을 확률을 추정하는 판별자 D(Discriminative model)

G의 훈련 절차는 D가 실수(G가 만든 것을 진짜라고 판단)할 확률을 최대화하는 것. 즉, D의 출력이 1에 가깝게 학습됨. G는 D를 속이는 방향으로 파라미터를 업데이트함.

D의 목표는 진짜(x)와 가짜($G(z)$)를 구별하는 정확도를 최대화함. 진짜는 1, 가짜는 0으로 정확히 예측하는 것이 목표임.
G는 D가 구별에 실패하도록(D의 정확도를 낮추도록) 데이터를 생성함. 이는 D의 성공 확률을 최소화하는 것과 동일함.

이 minmax 게임은 가치함수 $V(G, D)$를 최대화하는 D와 최소화하는 G를 찾는 문제로 표현됨.

임의의 함수 G와 D의 공간에서(이론적으로 optimal한 G, D가정할 때), G가 훈련 데이터 분포($p\_data(x)$)를 완벽히 복원하고 D는 모든 곳에서 $1/2$의 값을 갖는 고유한 해가 존재함. 이 상태가 이 게임의 유일한 안정적인 균형점(내시 균형, Nash equilibrium)임.

> Q. 실제 데이터와 구별이 불가능한 가짜 데이터를 D가 받으면 1을 출력해야하는게 아닌가?
> A. D가 추정해야 하는 분포는 입력 x가 실제 데이터 p_data에서 왔을 확률 $P(Y=1|X=x)$임. 하지만 $p_data = p_g$이므로 이 x가 p_data에서 왔을 수도 있고, p_g에서 왔을 수도 있음. 두 가능성은 정확히 50% : 50%임. 따라서 $P(Y=1|X=x) = 0.5$임. 이는 D가 G에게 완벽하게 속은 상태임.

G와 D가 다층 퍼셉트론으로 정의될 때, 전체 시스템은 역전파(미분 가능한 신경망)로 훈련(구현)됨.
D는 V를 최대화 하기 위해 ascent, G는 V를 최소화 하기 위해 descent.

훈련이나 샘플 생성 과정 모두에서 $MCMC$, 전개된 근사 추론 네트워크(unrolled approximate inference networks, VAE)도 필요하지 않음.

> Q. 이때 unrolled되있다는 의미가 무엇인가?
> A. 이 모델은 markov chain처럼 여러 단계를 반복하는 계산 과정이 필요 없이 순전파, 역전파로 바로 계산 가능함.

1.Introduction

딥러닝 모델의 궁극적인 목표는 복잡한 고차원 확률 분포를 표현(학습, 추정, 발견)하는 것임. 즉, 데이터가 어떻게 생겨나는지에 대한 원리를 학습하는 것.

지금까지 딥러닝에서 화두인 모델은 주로 판별 모델(discriminative model)이었음. 판별 모델은 관측된 데이터 x가 주어졌을 때 레이블 y의 조건부 확률 분포 $P(Y|X=x)$를 모델링함. 반면 생성 모델(generative model)은 관측된 데이터 x의 전체 분포 $P(X=x)$를 모델링함. 이러한 모델은 주로 역전파 및 드롭아웃 알고리즘 기반이고 특히 조각별 선형 유닛(piecewise linear unit)를 사용함.

> Q. 판별 모델이란 분류 모델인가?
> A. 두 모델은 같은 모델이지만 GAN의 맥락에서 판별 모델 D는 입력 데이터 x를 오직 real, fake라는 두 개의 클래스로 분류하는 이진 분류 모델임. 실제로 D는 입력 데이터 x가 '진짜'일 확률을 0과 1 사이 값으로 출력함.(0.9: 90% 확률로 진짜라고 판단)

> Q. 조각별 선형 유닛이란?
> A. 함수 전체는 선형이 아니지만 여러 조각(구간)으로 나누어 보면 각 조각이 선형인 함수를 말함. 대표적인 예시로 $ReLu(f(x)=max(0,x))$와 그 변형들을 의미.

반면, 딥러닝 생성 모델은 그 영향력이 적었는데 이는 MLE및 관련 전략에서 발생하는 다루기 힘든(intractable) 많은 확률적 계산을 근사하는 것의 어려움 때문임. 또란 생성 모델의 맥락에서 조각별 선형 유닛의 이점을 활용하기 어려움.

> Q. 다루기 힘든 확률적 계산이란?
> A. 계산량이 너무 많아 현실적으로 계산이 불가능한 확률 계산을 의미. 주로 분배 함수(partition function) Z의 계산과 관련있음.
> 볼츠만 머신과 같은 전통적인 생성 모델은 데이터의 확률 $p(x)$를 에너지 함수 $E(x)$를 이용해 $p(x)=[exp(-E(x))/Z]$로 정의함. 여기서 Z는 $p(x)$의 총합을 1로 만들기 위한 정규화 상수로 $Z=\sum_xexp(-E(x'))$는 모든 가능한 상태 x에 대해 합산해야하는 분배 함수임. 이 Z의 계산이 다루기 힘든(intractable) 확률적 계산임.
> 예를 들어 mnist 이미지일때 28x28 픽셀 흑백 이미지라고 해도 이때 가능한 이미지의 총 개수는 256^{28x28}로 이 모든 경우의 수를 더해서 Z를 계산하는 것은 사실상 불가능함.

> Z를 모르니 $p(x)$를 정확히 알 수 없고, 따라고 MLE를 수행할 수 없음. 이 때문에 MCMC같은 복잡한 근사 기법에 의존해야 했음.

> Q. 생성 모델의 맥락에서 왜 조각별 선형 유닛의 이점을 활용하기 어려운가?
> A. 많은 생성 모델은 범위가 정해진([0~1], [-1~1]) 데이터를 다루는데 $ReLU$는 양수 방향으로 값이 무한정 커질 수 있음. 따라서 보통 $Sigmoid$, $Tanh$를 출력층에 사용.
> $MCMC$나 RBM(Restricted Boltzmann Machine)같은 모델은 샘플을 생성하거나 학습할 때 반복적인 피드백 루프 구조를 가지는데 이때 값이 무한정 커질 수 있는 $ReLU$를 사용하면 값이 발산하여 계산이 불안정(unstable)해지기 쉬움.
> GAN은 이러한 피드백 루프나 $MCMC$가 필요없기에 은닉층에 $ReLU$를 사용하고 최종 출력층에만 $Tanh$ 등을 사용하여 값의 범위를 조절할 수 있음.

이때 GAN은 $p(x)$의 MLE를 직접 계산하거나 근사하는 대신 $p(x)$로부터 샘플링하는 방법을 학습함. 또, G와 D를 표준적인 MLP로 구성하여 역전파를 통해 훈련하기에 $ReLU$와 같은 활성화 함수의 이점을 그대로 활용함.

적대적 신경망 프레임워크에서 G는 샘플이 G에서 왔는지 p_data에서 왔는지 판별하는 D와 경쟁해야 함.

이때 G를 위조지폐범 팀, D를 경찰로 비유할 수 있음. 이 게임은 G가 D가 구별 불가능하게 될 때($p\_g = p\_data$)까지 계속됨.

이 논문에서는 G와 D를 가장 기본적인 신경망 MLP로 구현했지만 이 프레임워크는 MLP에 국한되어있지 않고 CNN, RNN 등 다양한 신경망 구조에 적용 가능함.

2.Related Work

잠재 변수를 가진 방향성 그래픽 모델(directed graphical model, DGM)의 대안으로 제한된 볼츠만 머신(restricted Boltzmann machine, RBM), 심층 볼츠만 머신(deep Boltzmann machine, DBM)과 같은 무방향성 그래픽 모델(undirected graphical model, UGM)이 있음.(참고: BM과 RBM, MCMC)
(다루기 힘든 분배 함수 Z, $MCMC$ 의존성, 느린 학습 속도, 불안정함.)

> Q. 방향성 그래픽 모델(Deriected Graphical Model, DGM)이란?
> A. 베이지안 네트워크(Bayesian Network)라고도 불림. 이 모델은 확률 변수 간의 조건부 의존성(conditional relationship)을 방향이 있는 화살표(edge)로 표현함.
> 예를 들어 $A \rightarrow B$ 라면, B의 확률은 A에 조건부로 의존함을 의미함. 즉, $P(B|A)$로 표현됨. 또, $P(A,B,C) = P(A)P(B|A)P(C|B)$처럼 전체 결합 확률을 조건부 확률의 곱으로 간단하게 분해할 수 있음. VAE의 인코더/디코더나 DBN의 하위 층이 여기에 속함.

> Q. 무방향성 그래픽 모델(Undirected Graphical Model, UGM)은 EBM과 어떤 관계인가?
> A. UGM은 markov random field라고도 불리며, 변수 간의 상관관계를 방향이 없는 선으로 표현함. EBM은 UGM을 수치화하는 방법인데 UGM의 연결된 노드집합(clique)마다 포텐셜 함수 혹은 에너지 함수($E(x)$)를 정의함. UGM의 결합 확률 $P(x)$는 이 에너지 함수들을 이용해 $P(x) = \frac{1}{Z}\exp(-E(x))$ 형태로 정의됨. 따라서 RBM, DBM 같은 UGM은 EBM의 대표적인 예시임.
>
> 향후에 EBM에 대해 자세히 다루기

이러한 (무방향성) 모델 내 상호작용은 정규화되지 않은 포텐셜 함수의 곱으로 표현되며, 이는 모든 확률 변수 상태에 대한 전역적인 합산/적분에 의해 정규화됨.

$$P(x) = \frac{\tilde{P(x)}}{Z}$$

위 식에서분자 $\tilde{P(x)}$가 '정규화되지 않은 포텐셜 함수'이고 Z가 분배함수(partition function, normalizing constant)임.
이 Z의 계산과 Z의 gradient는 대부분의 경우에서 다루기 힘든(intractable) 문제임. 비록 $MCMC$ 방법으로 복잡한 기법을 통해 근사할 수는 있음.

> Q. 정규화 상수 Z가 다루기 쉬운 경우는 어떤 경우인가?
> A. 상태 공간이 매우 작은 경우. 예를 들어 16개의 픽셀만 가진 이진 이미지(4x4)라면 가능한 이미지의 총 개수는 $2^{16} = 65536$개로 이 모든 경우의 수를 더해서 Z를 계산하는 것이 비교적 수월함.
> 혹은 모델 구조가 매우 단순한 경우: 변수 간의 의존성이 트리(tree)구조처럼 매우 단선해 Z를 ($MCMC$ 없이) 효율적으로 계산할 수 있는 알고리즘이 존재할 때.
> RBM이나 DBM은 위 두 경우에 해당하지 않으므로 Z가 intractable함.

$MCMC$ 샘플링은 전체 분포를 고루 탐색하지 못하고 특정 mode에 갇히는 경향이 있음.

또 다른 주요 생성 모델인 DBN은 RBM을 쌓아올린 구조로 최상위 층은 무방향성(RBM), 그 아래는 방향성이라는 독특한 하이브리드 구조를 가짐. DBM은 층별로 빠르게 근사할 수 있지만 무방향성 모델과 방향성 모델 양쪽 문제를 모두 가짐.

$MCMC$를 피하기 위한 다른 다른 접근법으로 $ScoreMatching$, NCE(Noise-Contrastive Estimation)이 있는데 두 기법은 Z계산을 피하는 대신 $P(x) \propto \tilde{P}(x)$에서 $\tilde{P}(x)$를 직접 계산해야 함.
DBN이나 DBM은 여러 층의 잠재 변수를 가지는 생성모델에서는 구조가 너무 복잡해서 $\tilde{P}(x)$조차 유도하기 힘듦.

DAE같은 일부 모델은 이론적으로 RBM에 적용된 스코어매칭과 관련이 있음.

NCE에서는 GAN처럼 생성모델을 학습시키기 위해 판별적 훈련 기준이 사용됨. NCE에서는 생성자 G가 D의 역할까지 겸임한다는 점이 GAN과의 차이점임. 또한, NCE에서 G의 적수는 '고정된'노이즈라 문제가 너무 쉬움. 반면 GAN에서 D는 G가 강해짐에 따라 함께 강해지는 적응형(adaptive) 적수이므로 G에게 지속적인 학습 동기(gradient)를 제공할 수 있음.

확률 분포를 명시적으로(explicitly)정의하지 않고, 대신 원하는 분포에서 샘플링하는 절차만 제공하는 암시적 모델을 소개. $MCMC$나 복잡한 확률 계산 대신 딥러닝의 역전파를 사용.

GAN과 같은 암시적 모델인 GSN이 있는데 샘플링 과정에서 markov chain을 정의해야 하는 한계를 가짐.
하지만 GAN은 $MCMC$가 필요 없이 단 한번의 순전파로 샘플링이 가능함.
또한, 샘플링 중 피드백 루프를 가지지 않기에 $ReLU$같은 활성화 함수를 은닉층에 자유롭게 사용하여 학습 성능을 높일 수 있음.
역전파를 통해 생성 기계를 훈련시키는 최근 연구로 VAE가 있는데 GAN과 달리 별도의 추론 네트워크(인코더)가 필요하며 목적함수($ELBO$)가 다름.

3.Adversarial Nets

적대적 프레임워크 자체는 G와 D를 MLP로 구현하는 가장 기본적인 사례에 집중함.

데이터 공간 X에 대해 G가 $p_g(x)$를 학습하기 위해 먼저 입력 노이즈 공간 Z에서 노이즈 변수에 대한 사전 분포(prior, p_z)를 정의함.

그 다음 X로의 매핑을 $G(\mathbf{z};\theta_g)$로 표현. 여기서 G는 θ_g라는 파라미터(weights, bias)를 가지는 MLP(신경망)로 표현되는 미분 가능한 함수(역전파로 학습시켜야 하므로)임.
p_z는 G의 입력이 되는 무작위 노이즈의 분포임.(보통 간단한 정규 분포 또는 균등 분포로 설정).
$\mathbf{z}$를 '잠재 변수(latent variable)'라고 부름.
p_g는 G가 생성하는 데이터들의 분포임.

> Q. 데이터공간 X와 노이즈 공간 Z의 차이는 무엇인가?
> A. 노이즈 공간 Z는 생성자 G의 입력 z가 샘플링되는 공간임. 보통 N, U를 따르며 데이터 공간 X보다 저차원인 경우가 많음.
> 데이터 공간 X는 우리가 생성하려는 실제 데이터 x가 존재하는 공간임.
> 생성자 G는 Z에서 샘플링된 z를 입력받아 X로 매핑하는 함수($G : Z -> X$)임.

이어서 단일 스칼라 값을 출력하는 두번째 MLP $D(\mathbf{x};\theta_d)$를 정의함.
D는 데이터 x를 입력으로 받는 함수(신경망)임. θ_d는 D 신경망의 학습 파라미터(weights, bias)이

고 최종 출력값은 숫자 하나임.

D는 입력 x가 p_g가 아닌 p_data로부터 왔을 확률을 나타냄. $P(Y=1|X=x)$가 D가 추정하는 확률 분포임.
이때 D의 목표는 입력 x가 진짜, 가짜인 상황에서 올바른 레이블을 할당할 확률을 최대화 하는 것임. 전형적인 이진 분류 문제.

D를 학습시킴과 동시에 $\log(1-D(G(\mathbf{z})))$를 최소화하도록 G를 훈련. 다시 말해 D가 $G(z)$를 받았을 때 0에 가까운 스칼라 값을 내뱉도록 함.

> Q. 이때 x와 y는 확률변수인가? 그렇다면 각 x, y에 대해 결합확률분포, 주변확률분포, 조건부 확률분포로 정의할 수 있는가?
> A. 가능함. x는 데이터공간 X에서 샘플링된 데이터로 $x \sim p\_data$ 또는 $x \sim p_g$로 표현할 수 있음.
> Y는 해당 데이터 x의 실제 레이블을 나타내는 이진 확률 변수임. $y=1 : x \sim p\_data$,
> $y=0 : x \sim p_g$로 표현 가능하고 이때 y의 주변확률분포
> $P(Y=y) = p(Y=1) + p(Y=0) = 0.5 + 0.5 = 1$로 정의할 수 있음(보통 알고리즘에서 D를 업데이트할 때 동일한 개수(m)dml 실제 데이터와 가짜 데이터를 사용함. D가 학습하는 미니배치는 총 $2m$개의 데이터로 이루어져 있으며 진짜와 가짜는 각각 m개임. 따라서 $P(Y=1)$과 $P(Y=0)$은 $m/2m = 0.5$로 정확히 같음. D가 두 클래스(진짜, 가짜)를 공평하게 구별하도록 학습시키기 위한 장치임.)
> 이때 x의 주변확률분포
> $P(X=x) = P(x|Y=1)P(Y=1) + P(x|Y=0)P(Y=0) = 0.5p_data(x) + 0.5p_g(x)$로 정의할 수 있음. 이는 D가 보는 전체 데이터 x의 분포임.
> x, y의 조건부 확률 분포는 아래와 같이 정의할 수 있음.
> $p(x|Y=1) : x$가 진짜일때 데이터 분포 = $p\_data(x)$
> $p(x|Y=0) : x$가 가짜일때 데이터 분포 = $p\_g(x)$
> $p(Y=1|x)$:데이터 x가 주어졌을 때, 그것이 진짜일 확률임. 이것이 D가 추정하려는 분포임.
> 베이즈 정리에 의해
> $p(Y=1|x) = [p(x|Y=1)P(Y=1)]/P(X=x) = [p\_data(x) * 0.5]/[0.5p\_data(x) + 0.5p\_g(x)] = 1$
> 로 표현 가능함. 또한 이 수식은 최적의 판별자 $D*(x)$의 형태와 동일함.
> x, y의 결합확률 분포 $p(x,y)$는 $p(x|Y=y)P(Y=y)$로 정의할 수 있음. 이때
> $p(x, Y=1) = 0.5p\_data(x)$, $p(x, Y=0) = 0.5p_g(x)$임.

D와 G는 가치함수 $V(G,D)$를 이용해 minimax게임을 함.

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

이때 D의 목표는 $V(G,D)$ 전체를 최대화 하는 것임.
먼저 D는 V를 최대화하려함.

첫 번째 항: $\mathbb{E}\_\mathbf{x} \sim p\_data(\mathbf{x})[\log D(\mathbf{x})]$ 은 D가 진짜 데이터 x를 받았을 때 $D(x) = 1$이라고 올바르게 판단하여 이 항의 값을 최대화(log1=0)해야 함.
두 번째 항: $\mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))]$은 D가 가짜 데이터 $G(z)$를 받았을 때 $D(G(z)) = 0$이라고 올바르게 판단하여 이 항의 값을 최대화(log1=0)해야 함.
(참고로 이 가치함수 V는 이진 분류에 사용되는 이진 교차 엔트로피와 정확히 일치함. D를 이 손실을 최소화하는 분류기로 볼 수 있음.)
G는 V를 최소화해야함. 이때 첫번째 항에은 관여하지 못함.
G가 두번째 항을 최소화한다는 것은 $\log(1 - D(G(z)))$값을 가능한한 음수($-\infty$)로 만드느 것임.
즉, D가 $G(z)$를 받았을 때 $D(G(z)) = 1$이라고 잘못 판단하도록 만드는 것임.

4장에서 G와 D에 충분한 용량(capacity)이 주어진다면, 즉 비모수적 한계(non-parametric limit)에서 G가 추정하는 $p_g(x)$가 $p_data(x)$에 수렴할 수 있음을 보여줌. 이때 충분한 용량, 비모수적 한계란 G와 D가 모든 가능한 함수(신경망)도 표현할 수 있는 이상적인 모델이라고 가정함을 의미함. 이때 이상적인 G와 D가 추정하는 분포가 같아지는 저짐에서 균형을 이룬다는것을 4장에서 수학적으로 증명함.

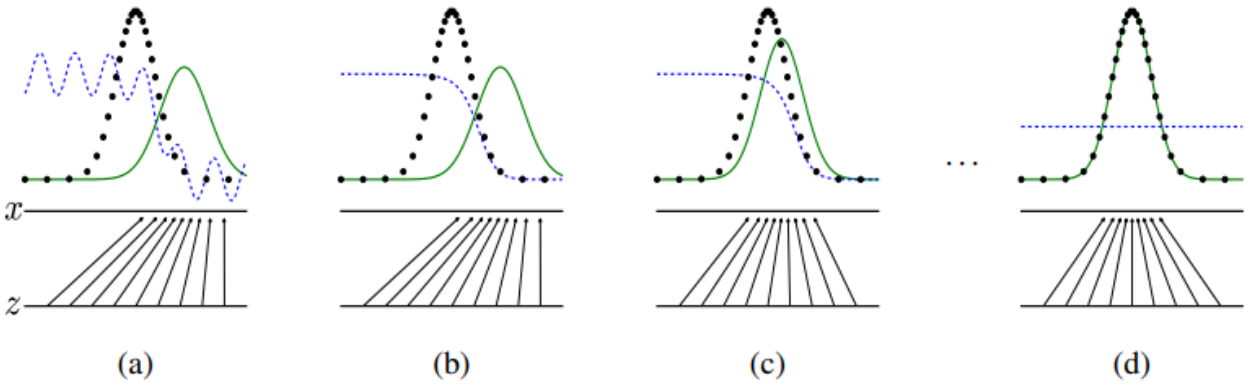
figure1은 p_g(초록색)가 p_data(검은 점선)에 점차 가까워지고 D(파란 점선)가 $1/2$로 수렴하는 과정을 시각적으로 보여줌.

실제로 이 minimax게임을 훈련시키기 위해 D와 G를 번갈아가며 업데이트하고 그 방식이 바로 미분을 이용한 경사하강법과 같은 수치 최적화 기법임.

훈련 과정에서 D를 완전히 최적화하는 것은 계산적으로 매우 비싸고, 유한한 데이터셋에서는 과적합을 초래함. $V(G, D)$를 이론적으로 보면, G를 1스텝 업데이트하기 전 D를 먼제 계산해야 함. 이 의미는 현재의 G를 상대로 D가 완벽한 최적의 판별자 $D*$가 될 때까지(수없이) D를 수렴시키는 것을 의미.
대신 실제 구현시에는 D를 k번의 미니배치 업데이트 한 후에 G를 1번의 미니배치 업데이트를 수행함. (Algorithm 1에서 k를 1로 정의)
G가 충분히 천천히 변하는 한 D가 그때그때 최적의 해 근처에 머물도록 하는 결과를 유도.
이 전략은 SML/PCD 훈련 방식이 학습 과정에서 Markov chain을 버닝(burning in) 하는것을 피하기 위해 한 학습 단계에서 다음 단계로 markov chain의 샘플을 유지하는 방식과 유사함.

> Q. SML/PCD란 무엇인가?
> A. SML/PCD란 볼츠만 머신(RBM)을 학습시킬때 쓰던 기법임. RBM 학습 시 $MCMC$샘플링을 하는데 이 $MCMC$가 안정적인 분포에 도달하려면 초기에 많은 반복(burn-in)이 필요함. 이 과정이 매우 비쌈.
> SML은 매 스텝마다 이 버닝을 새로 하지 않고, 이전 스텝의 $MCMC$샘플을 재활용해서 계싼을 아낌. GAN도 비싼 D를 완전히 최적화하는 과정을 k스텝으로 생략함. G와 D 둘 다 이론적으로 완벽한 계산을 실용적인 근사치로 대체하여 효율성을 높임.

실제로 $V(G,D)$는 G가 학습하기에 충분한 그래디언트를 제공하지 못할 수 있음. 학습 초기 G가 매우 약할 때 $G(z)$는 거의 완벽하게 x와 다르기에 D가 높은 확인을 가지고 $G(z)$를 기각할 수 있음. 이때 $\log(1-D(G(z)))$가 포화(기울기가 0에 가까워짐)되어 G가 업데이트 되지 않을 수 있음. $\log(1-D(G(\mathbf{z})))$를 $D(G(\mathbf{z}))$로 미분한 그래디언트($\frac{1}{1-D(G(\mathbf{z}))}\cdot(-1)$가 0 근처에서 매우 작아져 학습이 멈추게 됨.
위 포화 문제를 해결하기 위해 목적함수를 대체함.
$\min_G \log(1-D(G(\mathbf{z})))$ -> $\max_G \log D(G(\mathbf{Z}))$
두 함수는 G가 D를 속이려 한다는 점에서 목표는 동일하지만 학습 초기에 그래디언트 특성이 완전히 다름. $\log D(G(z))$는 $D(G(z))$가 0에 가까울 때 기울기가 매우 가파름. 따라서 G가 약할 때도 강한 그래디언트를 제공하여 학습이 멈추지 않도록 함.



Z에서 샘플링된 z가 G에 입력으로 들어가 X공간으로 매핑됨.
검은색 점선(p_data), 초록색 선(p_g), 파란색 선(D의 출력값).
a: D가 p_data와 p_g를 잘 구별하려고 시도함.
b: G는 고정된 채로 D만 학습함. a보다 명확하게 p_data와 p_g를 구별함. 이론적으로 $D*(x)=p_data(x)/(p_data(x)+p_g(x))$에 수렴함.
c: D가 고정된 채로 G만 학습함. b에서 D가 높게 평가했던 영역으로 p_g를 이동시킴.
d: 무수한 단계를 거친 후(b, c 반복) G와 D는 균형점($D(x)=0.5$)에 도달함. $p\_g=p\_data$. G가 실제 데이터 분포를 완벽하게 복제함.따라서 어떤 x가 입력되어도 그게 $x\sim p_data$에서 왔는지 $x\sim p_g$에서 왔는지 D가 구별할 수 없는 상태임. 파란색 파선이 모든 x에 대해 0.5로 평평해진 것이 이를 의미함.

4.Theoretical Results

G는 $\mathbf{z}\sim p_z$에서 얻어지는 $G(z)$의 분포로서 확률분포 p_g를 암시적으로(implicitly)정의함. 이때 암시적이란 p_g의 확률 값을 직접 알려주지 않고 p_z라는 간단한 분포에서 노이즈 z를 뽑아 $G(z)$를 계산함으로서 p_g분포에 속하는 샘플을 생성하는 절차만 제공함.
반대로 명시적(explicit) 모델 $p\_g(x)$=어떤 복잡한 함수로 표현되며, x가 나타날 확률값을 직접 계산할 수 있는 모델임.(VAE, RBM 등)

따라서 Algorithm 1(D와 G를 번갈아가며 업데이트)이 충분한 용량과 훈련 시간이 주어진다면 (G가 추정하는 p_g가)p_data의 좋은 추정치로 수렴하기를 기대함.

이 결과는 비모수적(non-parametric)환경에서 수행됨. 이때 비모수적 환경이란 G와 D가 모든 가능한 함수(신경망)를 표현할 수 있는 이상적인 모델이라고 가정함을 의미함. 모수적(parametric)이란 모델이 MLP, CNN 등 특정 구조와 유한한 개수의 파라미터로 고정된 경우임. GAN이라는 minimax 게임 아이디어가 '최적 상태에 수렴한다'를 증명하기 전에, 자체가 이론적으로 완벽한 해답($p\_g = p\_data$)을 갖고 있는가?를 먼저 따져보는 것.

4.1절에서 minimax게임이 global optimum을 가짐을 보임.

Algorithm 1

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.

미니배치 SGD를 사용함으로서 전체 데이터가 아닌 m개의 샘플로 구성된 미니배치를 사용해 그래디언트를 근사하고 파라미터를 업데이트함.

k: D에 적용할 스텝 수(G와 D를 1:1 비율로 동기화하는 $k = 1$을 선택)

p_z로부터 노이즈 샘플 미니배치를 샘플링.(m개의 z벡터를 준비)

$p_data(x)$로부터 m개의 실제 데이터 샘플 미니배치를 샘플링.

총 노이즈 m개, 실제 데이터 m개를 뽑는 것은 $P(Y = 1) = P(Y = 0) = 0.5$로 D가 두 클래스를 공평하게 구별하도록 학습시키기 위한 장치임.

D의 Stochastic gradient ascent를 사용해 D의 파라미터 θ_d를 업데이트함.

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

D가 k번 학습했으면 G를 학습함.
p_z로부터 m개의 노이즈 샘플 미니배치를 다시 샘플링. 이때 D를 학습시킬 때 사용했던 z샘플을 재사용하지 않고 새로운 m개의 z를 샘플링.

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

G의 Stochastic gradient descent를 사용해 G의 파라미터 θ_g를 업데이트함.(실제 구현에서는 이 수식 대신 대체 목적 함수 $\nabla_{\theta_g}\frac{1}{m}\sum_{i=1}^{m}\log D(G(z^{(i)}))$를 사용는 것이 학습 초기에 더 효과적임.)
이때 옵티마이저는 모멘텀을 사용.(다른 최적화기와 호환 가능)

4.1 Global Optimality of $p\_g = p\_data$

minmax $V(D,G)$에서 안쪽 max_D먼저 고려함.(임의의 주어진 G에 대해 최적의 D를 고려) 고정된 G가 생성하는 p_g를 상대로 D가 자신의 성능($V(D,G)$)을 최대화했을 때 도달하는 판별자 $D*$를 찾으려고 함.
명제 1. G가 고정되어 있을 때, optimal한 D는 다음과 같음.

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

직관적으로 봤을때 만약 입력 x가 실제 데이터같이 보인다면 $0.9/0.9+0.1=0.9$(90%확률로 진짜)가 됨.
만약 입력 x가 가짜같이 보인다면 $0.1/0.1+0.9=0.1$(10%확률로 진짜)가 됨.
만약 $p_data(x)=p_g(x)$가 되는 지점(D가 구별 불가능한 지점)에서는 $D_G*(x)=0.5$가 됨.
이 공식은 D가 두 확률 밀도($p\_data(x)$, $p\_g(x)$)의 비율을 완벽하게 파악하고 있음을 보여줌.(이전에 $P(Y=1|x)$ 수식과 동일함)

증명. 주어진 G에 대해 D의 훈련기준은 $V(G,D)$를 최대화 하는 것. $V(G,D)$를 적분형태로 쓰면 다음과 같음.

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

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

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

> Q. 기대값 형식의 V가 적분형태로 바뀌며 수식이 어떻게 변형되는지?
> A. 기댓값은 개념적(이론적)으로 확률변수의 기댓값을 의미함. 이를 구현하기 위해 적분 형태(수학적 정의)로 바꾼 것임. 의미는 동일함. 적분 형태로 표시하면 $D(x)$에 대해 편미분(정확히는 변분법을 이용한 함수 미분)을 수행하여 그 값을 0으로 만드는 D를 찾을 수 있음.
>
> Q. 변분법을 이용한 함수 미분이란?
> A. 범함수를 최적화하는 방법임. 이때 범함수(functional)란 함수를 입력받아 숫자(V)를 출력함. GAN 증명에서 우리가 최적화(최대화)하려는 $V(G,D)$는 $D(x)$라는 (일반)함수 전체에 의존하는 적분값임. 즉, V는 범함수임.
> 이때 $D(x_1)$의 값은 오직 x_1지점의 V값에만 영향을 주고 $D(x_2)$의 값은 오직 x_2지점의 V값에만 영향을 줌.(서로 다른 x지점의 D값이 서로 영향을 주지 않음.)
> 따라서 V의 전체 적분 값을 최대화하는 가장 간단한 방법은, 모든 개별 x지점에서 적분 안의 값을 각각 최대화하는 것.
> 결국 범함수 V를 D에 대해 미분한다는 목잡한 문제가 모든 x에 대해 $L(x,D(x))$라는 일반 함수를 $D(x)$라는 (해당 x에서의)단일 값에 대해 미분한다는 단순한 문제로 바뀜.
>
> Q. 실제 데이터x와 가짜 데이터 $G(z)$의 분포를 하나의 적분으로 표현할 수 있는가?
> A. 이를 치환 적분이라고 함. G는 z를 데이터 공간 X로 매핑($x'=G(z)$)하는데 $z\sim p_z$일 때, $x'\sim p_g$를 따르므로 $D(G(z))$에 대한 z의 기댓값은 $D(x')$, $x'\sim p_g$의 기댓값과 같음.

엄밀하게 말하자면 $\mathbf{x}=G(\mathbf{z})$이고, 확률론의 '무의식적인 통계학자의 법칙(Law of the unconscious statistician)'에 따라, $\mathbf{z}\sim p_z$일 때 $f(G(\mathbf{z}))$의 기댓값($\mathbb{E}[f(G(\mathbf{z}))]$)은 $\mathbf{x}\sim p_g$일 때 $f(\mathbf{x})$의 기댓값($\mathbb{E}[f(\mathbf{x})]$)과 같음.

즉, 다음이 성립함.

$$\int_{\mathbf{z}} p_z(\mathbf{z})\log(1-D(G(\mathbf{z})))d\mathbf{z}=\int_{\mathbf{x}} p_g(\mathbf{x})\log(1-D(\mathbf{x}))d\mathbf{x}$$

For any $(a,)\in {}^2\mathbf{0},\mathbf{0}$, the function $y\to a\ \log(y)+\ \log(1-y)$ achieves its maximum in $[0,1]$ at $\frac{a}{a+}$.
$a,$ 가 0이 아닌 임의의 실수 쌍일 때, $y\to a\log(y)+\log(1-y)$ 함수는 $[0,1]$ 구간에서 최대값을 $\frac{a}{a+}$에서 가짐.

> Q. 왜 이 y함수는 $[0,1]$ 구간에서 최대값을 $\frac{a}{a+}$에서 가지는가?(y에 대한 함수 f미분 증명 유도)
> A. $d\log(y)/dy=1/y$, $d\log(1-y)/dy=-1/(1-y)$임.
> 이때 $f'(y)=0$이 되는 지점을 찾으면
> $a/y=/(1-y)\to a(1-y)=y\to a-ay=y\to a=ay+y\to a=(a+)y\to y=a/(a+)$
> 임.

> 2차 미분을 유도하여 볼록성을 확인해보면 $a, >= 0$이므로 $f(y)$는 항상 음수이므로 위로 볼록. 따라서 $f'(y) = 0$인 지점인 $y = a/(a +)$에서 global maximum을 가짐.

D는 p_data또는 p_g의 지지 집합(Support)밖에서는 정의될 필요가 없으므로, 증명이 완료됨.

> Q. 지지 집합이란?
> A. 지지 집합이란, 확률 분포에서 확률(밀도)값이 0이 아닌 모든 값들의 집합을 의미함. 예를 들어 주사위를 던지는 시행에서 확률변수 $X = 1, 2, 3, , 5, 6$이고 이때 지지집합은 $1, 2, 3, , 5, 6$임. 이때 $P(Y|X = x) = 1/6$일때 $P(Y|X =) = 0$이므로 은 지지집합에 속하지 않음. $U(0, 1)$분포에서 지지 집합은 $[0, 1]$임.
> 논문에서 말하는 지지 집합은 실제 데이터가 존재할 확률이 0보다 큰 x영역, G가 생성한 가짜 데이터가 존재할 확률이 0보다 큰 x영역을 의미함. 두 영역의 **합집합**을 판별자 D가 정의되는 영역임.
> 이 영역(지지 집합 밖)에서는 $p_data(x) = p_g(x) = 0$이므로 V의 적분값은
> $0 \log[D(x)] + 0 \log[1 - D(x)] = 0$이 되어 $D(x)$의 값에 영향을 받지 않음. 따라서 D가 정의될 필요가 없음.
>
> Q. 추가로 만약 지지집합 밖에서 데이터에 대해 V의 적분값에 영향을 주고 싶은 상황이 생긴다면 어떻게 하는가?
> A. $V(G, D)$자체를 수정하거나 D의 입력 방식을 바꿔야 함. 현재 V의 적분항에서 지지 집합 밖의 데이터가 들어왔을 시 $D(x)$값이 1이든 0이든 상관없이 그래디언트가 0이 되어 학습할 수 없음.
> **지지 집합의 확장**
> 실제 데이터 $x(p\_data(x) > 0$ or $p_g(x) > 0)$지점에 노이즈를 더한 새로운 실제 데이터 분포 $p_data_noisy(x)$는 $x + noise$지점에서 더이상 그래디언트가 0이 아님. 이는 본래 x영역 뿐만 아니라 지지 집합 밖의 데이터 x에 대해서도 D를 정의할 수 있게 함. 이때 노이즈를 정규분포를 썼다면 이론적으로 데이터공간 전 영역에서 0이 아닌 값을 갖게 됨으로 지지 집합 밖을 포함하는 전체 영역에서 D가 정의될 수 있음.
>
> Q. prior p_z의 지지집합이 모든 실수공간 내라고 정의될때, G가 추정한 p_g의 지지집합도 모든 실수공간 내라고 정의할 수 있을까?
> A. 그렇지 않음. p_z의 지지집합을 어떤 100차원 내 모든 공간이라고 가정했을 때 G의 출력인 실제 데이터공간(ex. 784dimension)에서 p_g의 지지집합은 기껏해야 100차원의 낮은 차원의 매니폴드를 형성할 뿐임.

생성 모델(여기선 GAN)을 학습하는 주 목표는 실제 데이터 분포 p_data를 근사하는 p_g를 찾는 것인데 이 복잡성을 모델링하는 방법은 크게 p_z를 복잡하게 만들거나 G를 복잡하게 만들어야 함. 이때 GAN은 p_z가 아무리 단순해도 z를 X로 매핑하는 생성자 G가 충분히 강력하다면(즉, 깊은 신경망이라면), G는 단순한 분포를 통해 p_data와 같은 복잡한 분포를 근사할 수 있음.

또한 prior로 간단한 분포(N, U)를 사용하는 이유는 GAN의 장점 중 하나인 학습과 샘플링해서 $MCMC$같은 복잡한 추론 과정이 필요없기 때문임.

> Q. p_g가 G에 의해 암시적(Implicit)으로 정의된다는 의미가 무엇인가?
> A. $p_g(x)$의 확률함수 공식을 직접 알 수는 없지만, 그 분포로부터 샘플을 생성하는 절차는 알고 있다는 의미임. 반대로 명시적(Explicit)모델은 어떤 확률함수를 수학 공식으로 직접 정의하고 계산할 수 있는 모델임. 예를 들어 VAE나 볼츠만 머신과 같은 모델들은 $p(x)$를 명시적으로 계산하거나 근사하려고 시도함.

지지집합의 확장은 D가 훈련 샘플을 암기하는것이 아닌 p_data 주변 공간의 전체적인 구조를 학습하도록 강제하는 규제라고 볼 수 있음.

D의 목적함수는 Y가 x의 출처($y=1$이면 p_data, $y=0$이면 p_g)를 나타낼 때, conditional probability $P(Y=y|X)$를 추정하기 위한 log-likelihood를 최대화하는 것으로 해석됨. $V(G,D)$의 의미를 의진 분류 관점에서 명확히 하는데, $\log D(x)$는 x가 p_data에서 왔을 때의 로그 가능도이고, $\log(1-D(x))$는 x가 p_g에서 왔을 때의 로그 가능도임. D의 목표는 두 클래스에 대한 로그 가능도를 최대화하는 것임.

minimax 게임의 min_G 부분을 고려하기 위해 안쪽 $max_D\ V(G,D)$를 G에 대한 함수 $C(G)$로 정의했을때 아래와 같음.

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

$$C(G) = \max_D V(G,D) = \mathbb{E}_{x\sim p_{data}}[\log D^*_G(x)] + \mathbb{E}_{z\sim p_z}[\log(1-D^*_G(G(z)))] = \mathbb{E}_{x\sim p_{data}}[\log D^*_G(x)] + \mathbb{E}_{x\sim p_g}$$

이때 G가 최소화해야 할 $C(G)$는 p_data와 p_g의 함수로 표현되는데 이 값을 최소화하는 p_g를 찾는 것이 G의 목표임.
이 $C(G)$의 전역 최솟값은 이론적으로 $p_g = p_data$일 때 보장되고, 이때 $C(G)$의 값은 $-log$라는 값을 가짐.
증명은 수식 (2, 4)로 가능함.

$p_g = p_data$일때 전역 최솟값임을 보이기 위해 $C(G) - (-\log) >= 0$임을 보임.
$C(G) = \mathbb{E}\_x \sim p\_data[\log \frac{p\_data(x)}{p\_data(x)+p\_g(x)}] + \mathbb{E}\_x \sim p\_g[\log \frac{p\_g(x)}{p\_data(x)+p\_g(x)}]$, $\log\ = \log 2 + \log 2$,
상수의 기댓값은 상수이므로 위 식을 $E_x \sim p_data[\log 2] + E_x \sim p_g[\log 2]$로 쓸 수 있음.
이때 $C(G) + \log$를 계산해보면 다음과 같음.

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

이때 두 확률분포 p_data, p_g에 대해 평균을 낸 M을 정의할때 위 식을 JSD로 쓸 수 있음.JSD 참고
결론적으로 $C(G) + \log = 2SD$와 같음. 이때 JSD는 distance measure이므로 그 값은 항상 0 이상임. 따라서 $C(G) >= -\log$임을 알 수 있음.
$SD = 0$이 되는 것은 $p_data = p_g$일 때 뿐임. 따라서 $C(G)$의 전역 최솟값($-\log$)은 $p_g = p_data$일 때 달성됨을 증명함.

4.2 Convergence of Algorithm 1

앞서 GAN의 minimax 게임이 이론적으로 $p_g = p_data$라는 전역 최솟값을 가진다는 것을 증명했는데 이가 실제 훈련에서 G와 D를 번갈아가며 gradient descent/ascent로 업데이트하는 Algorithm 1이 정말 그 이론적 균형점으로 수렴하는가를 따져봐야함.

만약 G와 D가 충분한 용량(capacity)을 가지고 Algorithm 1의 각 스텝에서 D가 그 시점 G에 대해 최적점에 도달($k = \text{in}$일때, D를 무한히 훈련)하는 것이 허용되고, p_g가 $C(G)(= 2SD(p_data||p_g) - \log)$로 정의된 기준을 개선하는 방향으로 업데이트된다면 p_g는 p_data로 수렴한다.
이때 $C(G)$를 p_g에 대한 함수의 관점에서 보면 convex함. 함수 $C(G)$가 convex하다는 것은 p_g의 local minimum이 global minimum임을 의미함. 다시 말해, p_g를 경사하강법을 통해 최적해에 수렴이 보장된다는 것임.

하지만 이론적으로 완벽하다는거지 실제 훈련에서 p_g를 직접 조작할 수 없고 신경망 G의 파라미터만 조작할 수 있음. $C(G)$가 분포 p_g에 대해서 convex지만 파라미터 θ_g에 대해서 non-convex함. 따라서 Algorithm 1이 실제 $p_g = p_data$에 수렴한다는 보장은 없음. 그럼에도 실제로 돌려보니(경험적으로) MLP가 잘 작동하니 이 framework는 잠재력이 있다고 저자들이 주장하는 것임.

5. Experiments

이 섹션은 앞서 제시한 이론이 실제로 작동하는지, 기존 생성모델들과 비교를 보여줌.
MNIST, TFD, CIFAR-10 데이터셋에서 GAN을 실험.
G는 $ReLU$와 $Sigmoid$를 혼합하여 사용, D는 $maxout$, $dropout$ 사용.
이론적으로 G의 입력(z)외에도 G의 중간층에 노이즈를 주입할 수도 있지만 이 실험에서는 입력 노이즈만 사용함.
GAN에서 p_g의 형태를 p_data와 직접 비교할 수 없기 때문에 파르젠 윈도우 방식을 사용하여 p_g를 검증.

> Q. 파르젠 윈도우(Parzen Window) 방식이란?
> A. 커널 밀도 추정(KDE, Kernel Density Estimation)의 다른 이름임. PDF의 정확한 수식은 모르지만, 그 분포에서 나온 샘플들만 가지고 있을 때, 그 샘플들을 이용해 원래 PDF를 추정하는 통계적 기법임.

결과적으로 GAN이 "기존모델보다 압도적으로 좋다"기보다는, 기존의 복잡한 $MCMC$기반 모델들과 비슷한 성능을 내면서도 샘플링이 매우 빠르다는 점을 강조함.
이 가능도 추정 방법(파르젠 윈도우)는 분산이 다소 높고 고차원 공간에서 잘 작동하지 않지만, 다른 방법이 없기 때문에 최선의 방법임.
GAN의 등장으로 생성모델을 평가하는 measure에 대한 연구가 필요함.(현재 $inceptionscore$, ID 등이 제안됨.)
figure2의 오른쪽 끝 열이 G가 생성한 샘플(왼쪽 5열)과 가장 유사한 훈련 데이터임. 이는 G가 단순히 훈련 데이터를 암기하지 않았음을 보여줌.

6. Advantages and disadvantages

GAN은 이전 모델들에 비해 장점과 단점을 모두 가짐.
단점은 $p_g(x)$에 대한 명시적 표현이 없는 암시적 모델임. 이는 샘플링은 가능하지만 특정 x의 확률값을 직접 계산할 수 없어 p_g를 평가하기 어렵게 만듦.
또한 훈련 중 D가 G와 잘 동기화되어야 한다는 것인데 D가 너무 강력해지면 G가 유용한 그래디언트를 받지 못함. 반대로 D가 너무 약하면 G가 쓸모없는 방향(D를 속이는 가장 쉬운 단 하나의 샘플만 샘플링)으로 업데이트 될 수 있음. 이를 Helvetica Scenario라고 하는데 이후 $modecollapse$로 알려짐. 이 동기화 문제는 $MCMC$체인을 계속 업데이트하는 것과 유사한 문제임.

장점은 $MCMC$가 필요없다는 것.
gradient를 얻기 위해 역전파만 사용된다는 것.
학습 중 추론이 필요 없다는 것.
다양한 함수에 적용가능할 수 있음(G와 D가 미분 가능하기만 하다면).

GAN이 다른 방식에 비해 학습/추론/샘플링 관점에서 우수하고 실제 구현에서 학습이 효율적이다를 강조함.

다른 모델은 재복원오차로 G(Encoder)를 학습시키는 반면 G는 실제 x를 직접 보지 못하고 오직 D로부터 간접적인 신호(gradient)만 받음. 이는 G가 x의 픽셀값이나 특징을 단순 재현(암기)할 위험이 적고 모델의 일반화 성능에 도움을 줄 수 있음.
또, $MCMC$기반 모델에서 모드간 mixing이 잘 되려면 각 mode 사이 확률 값이 0이 아닌 작은 값인 공간(흐릿한)이 필요함. 이 때문에 실제 샘플링이 blurry해지는 경향이 있음. VAE에서도 재복원 손실로 보통 MSE나 BCE를 사용하는데 이는 모델이 픽셀의 평균적인 정답을 맞히도록 유도하여 blurry 문제가 발생함. 반면, GAN은 p_g가 확률 1을 갖는 매우 얇은 매니폴드가 되도록 학습할 수 있어 blurry를 피할 수 있지만 모드 붕괴 현상이 발생할 수 있음.

7. Conclusion and Future Work

GAN은 다양한 방식으로 확장될 수 있음.

- $p(x)$대신 $p(x|c)$를 학습하는 conditional GAN($CGAN$).
- GAN은 생성($p(x|z)$)은 쉽지만, 추론($p(z|x)$)은 어려움. 이때 추론 네트워크를 추가하여 $p(z|x)$를 근사하는 방법.($BiGAN$, ALI)
- 파라미터를 공유하는 조건부 모델을 적대적 신경망을 사용하여 결정론적 $MP-DBM$(Multi-Prediction Deep Boltzmann Machine)의 확률론적 확장을 구현하는 것임.
  -준지도 학습(Semi-supervised learning): 레이블이 제한적인 상황에서 D또는 추론 네트워크에서 추출한 feature가 classifier의 성능을 향상시킬 수 있음($SGAN$, $DCGAN$)
- G와 D를 조정하는 더 나은 방법($k=1$, $TTUR$)을 고안하거나 prior을 단순한 정규분포 대신 더 의미있는 z를 샘플링하는 연구를 제안(향후 $styleGAN$의 W공간등으로 발전)

끝으로 적대적 모델링 프레임워크의 실현 가능성을 입증하고, 생성모델링 연구에 새로운 길을 열었다고 주장함.

References

RBM, DBM, DBN, GSN, NCE, VAE, Score Matching, ReLU, Maxout, Parzen Window