Abstract

두 모델이 경쟁하는 과정을 통해 Generative model을 추정하는 새로운 프레임워크를 제안한다.

1. G(생성모델, Generative model): 판별모델(Discriminative model)이 sample 데이터와 생성한 training data를 구분하지 못하도록 학습한다.
2. D(판별모델, Discriminative model): Sample 데이터가 G가 아닌 실제 training 데이터로부터 오는 확률을 추정한다.

이 모델은 minimax two-player game으로 표현된다.

※ **minimax two-player game:**

임의의 함수 공간 G, D 내에서, G가 Training data 분포를 만들어내면서 이를 D가 Sample data인지 training data인지를 판별하는 확률은 ½가 된다. G와 D가 다층 퍼셉트론으로 구성된 경우, 모델은 역전파 알고리즘으로 훈련될 수 있다. (Markov-chain이나 Approximate Inference 과정을 수행할 필요는 없다.)

Introduction

지금까지 딥러닝에서의 큰 성공은 고차원 입력 데이터를 클래스 레이블에 매핑하는 방식으로 연구가 진행되었으며 이는 well-behaved gradient를 갖는 역전파 알고리즘이나 dropout 알고리즘에서 진행되었다. 하지만 심층 생성 모델(Deep Generative Model)에서는 MLE(Maximum Likelihood Estimation)과 같은 전략에서 발생하는 많은 확률 계산들을 근사하는 문제점이나 piecewise linear unit의 이점을 활용하는데 있어 나타나는 어려움 때문에 큰 성공을 이루기 어려웠다. 따라서, 본 논문에는 이러한 문제점들을 우회하는 새로운 생성 모델 추정법을 제안한다.

제안된 'Adversarial nets' 프레임워크에서 생성모델은 판별모델과 '적대적으로' 학습이 진행된다.

- 판별모델: Sample Data가 생성모델이 생성한 data인지 실제 data distribution으로부터의 data인지를 판별한다.

생성 모델은 위조 지폐를 생산하여 탐지되지 않고 사용하려는 위조자 팀에 비유할 수 있고, 판별 모델은 위조 지폐를 탐지하려는 경찰에 비유할 수 있다. 이 게임에서 경쟁은 위조품이 정품과 구별되지 않을 때까지 두 팀 모두 방법을 개선하도록 유도한다.

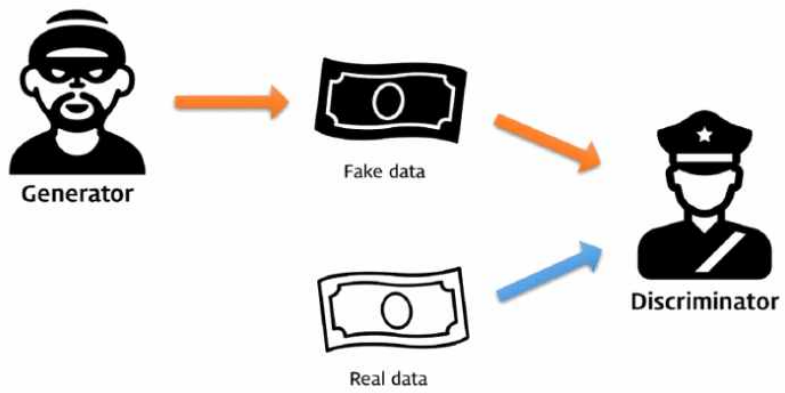

그림 1. GAN의 비유 (출처: https://velog.io/@wo7864/GAN-%EA%B0%9C%EB%85%90-%EC%9D%B4%ED%95%B4)

이 프레임워크는 많은 종류의 모델 및 최적화 알고리즘에 대한 특정 훈련 알고리즘을 생성할 수 있다. 본 논문에서는 생성모델이 다층 퍼셉트론을 통해 무작위 노이즈를 전달하여 샘플을 생성하고, 판별모델이 다층 퍼셉트론인 경우인 특별한 경우를 다룬다. 본 논문에서는 이 경우를 Adversarial net이라 부른다.

Adversarial net의 경우 복잡한 네트워크 없이 순전파, 역전파, dropout algorithm을 통해 두 모델을 훈련시킬 수 있다. (Abstract에서 명시되어있듯이 approximate inference나 markov chain 과정이 필요없다.)

Adversarial nets

적대적 모델링 프레임워크는 모델이 다층 퍼셉트론으로 이루어져 있을 때 가장 쉽게 적용할 수 있다. 데이터 x에 대한 생성자 분포 p_g를 학습하기 위해 입력 노이즈 변수 $p_z(z)$를 정의한 다음 이를 데이터 공간인 $G(z;\theta_g)$에 매핑하였다. 여기서 G는 매개변수 θ_g를 가진 다층 퍼셉트론으로 표현되는 미분 가능한 함수이다. 또한, 단일 스칼라를 출력하는 두 번째 다층 퍼셉트론인 $D(x;\theta_d)$를 정의한다. $D(x)$는 x가 p_g가 아닌 데이터에서 나온 확률을 나타낸다.

훈련 예제와 G로부터의 샘플 모두에 올바른 레이블을 할당할 확률을 최대화하기 위해 D를 훈련시킨다. 이와 동시에 $\log(1-D(G(z)))$를 최소화하기 위해 G를 훈련시킨다. 즉, D와 G가 $value\,function\ V(G,D)$의 two-player minimax game을 진행하는 것과 같다.

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

$D(x)$는 실제 데이터 x를 판별한 결과이며, $E[\log D(x)]$는 $(-\infty, 0]$값을 가지므로 $E[\log D(x)]$를 최대화하는 방식으로 훈련이 진행된다. 즉, $E[\log D(x)]$의 최댓값은 0이다. 이와 동시에 $D(G(z))$는 G가 생성한 가짜 데이터를 판별한 결과이며, 0에 가깝게 출력할수록 $\log[1-D(G(z))]$의 값은 0이 된다. 따라서, D가 잘 판별할수록 $V(D,G)$는 0에 가까워진다.

G의 경우 Fake image를 잘 생성할수록 두 번째 항에서 $D(G(z))$의 값은 1에 가까워진다. 따라서, G가 잘 훈련될수록 $V(D,G)$의 값은 $-\infty$에 가까워진다.

따라서, D는 $V(D,G)$값을 0에 가깝게 최대화시키는 방향으로 학습되고, G는 $V(D,G)$값을 $-\infty$에 가깝게 최소화시키는 방향으로 학습되기 때문에 본 논문에서는 D와 G가 $value\,function\ V(G,D)$의 two-player minimax game을 진행하는 것과 같다고 표현하였다.

학습의 내부 루프에서 D를 완전히 학습시키는 것은 과적합을 불러일으킬 수 있기때문에 계산적으로 금지하였다. 대신 D를 최적화하는 k단계와 G를 최적화하는 단계를 번갈아 수행하였다. 이렇게 하면 G가 충분히 느리게 변화하는 한 D가 최적의 값을 유지하였다.

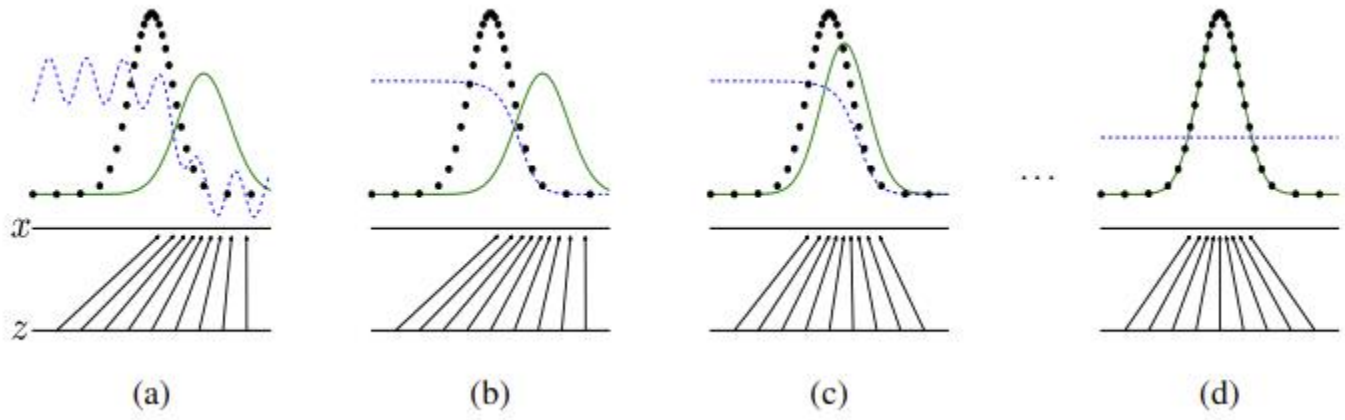

그림 2

그림을 보면 학습 초기 G의 성능은 좋지 않기 때문에 D가 높은 확률로 샘플을 구별할 수 있었다. 이 때는 $\log(1-D(G(z)))$값이 포화상태가 되기 때문에 $\log(1-D(G(z)))$값을 최소화시키기 위해 G를 훈련시키기 보다는 $D(G(z))$값을 최대화하는 방식으로 G를 훈련시켰다. 그 이유는 학습 초기 더 나은 기울기를 제공하기 때문이다.

※ 그림2 설명

그림은 GAN이 데이터 생성 분포(p_x, 검정색 점선)와 생성적 분포(p_g, 초록색 실선)의 샘플을 구분하는 판별 분포(D, 파란색 점선)를 동시에 업데이트하여 학습되는 것을 보여준다. 아래 수평선은 z가 균등하게 샘플링되는 도메인이며, 위쪽 수평선은 x의 도메인의 일부이다. 위로 향하는 화살표는 $x = G(z)$매핑이 변환된 샘플에 비균등한 분포 p_g를 부과하는 방식을 보여준다. G는 밀도가 높은 영역에서는 수축하고, 밀도가 낮은 영역에서는 확장한다.

(a): 학습 초기로, 실제 데이터(검정색 점선)와 Fake 데이터(초록색 실선)의 분포가 매우 다르다.

(b): D가 실제 데이터와 Fake 데이터를 분명하게 판별해내고 있다.

(c): G가 점점 업데이트 되면 D가 $G(z)$를 실제 데이터로 분류하는 방향으로 학습한다.

(d): 최종적으로 $p_g = p_{data}$가 되는 지점에 도달하게 되며 D는 두 분포를 구별할 수 없으므로 $D(x) = \frac{1}{2}$이다.

Theoretical Results

G는 $z \sim p_z$일 때 얻어진 샘플인 p_g를 정의한다.

Algorithm 1

- Minibatch SGD training of GAN
- k: The number of Discriminator's steps (Using k = 1)

for number of training iterations do
  for k steps do
    1. Sample mini

Global Optimality of $p_g = p_{data}$

먼저, G가 주어졌을 때 최적의 D는 다음과 같다.

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

Proof)

G가 주어졌을 때, D는 $V(G, D)$를 최대화하는 방향으로 훈련된다.

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

(0, 0)이 아닌 R^2의 원소 (a,b)에 대하여, 함수 $y \to a\log(y) + b\log(1-y)$는 $y = \frac{a}{a+b}$에서 최댓값($\in [0,1]$)을 갖는다. D의 목적함수는 $P(Y=y|x)$의 MLE(Maximum log-likelihood estimation)로 생각할 수 있다. 여기서 Y는 x가 p_{data}로부터 오면 1값을 가지고 p_g로부터 오면 0값을 가진다. 즉 다음과 같이 정리할 수 있다.

$$C(G) = \max_D V(G,D)$$
$$= E_{x \sim p_{data}}[\log D^*_G(x)] + E_{x \sim p_x}[\log(1 - D^*_G(G(z)))]$$
$$= E_{x \sim p_{data}}[\log D^*_G(x)] + E_{x \sim p_g}[\log(1 - D^*_G(x))] \quad // \ G(z)를 \ x로 \ 치환$$
$$= E_{x \sim p_{data}}[\log \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}] + E_{x \sim p_g}[\log \frac{p_g(x)}{p_{data}(x) + p_g(x)}] \quad // D에 \ MLE \ 적용$$

Theorem 1. The global minimum of the virtual training criterion C(G) is achieved iff $p_g = p_{data}$ at that point, C(G) achieves the value $-\log 4$

proof)

$p_g = p_{data}$이고 $D^*_G(x) = \frac{1}{2}$일 때, $C(G) = \log\frac{1}{2} + \log\frac{1}{2} = -\log 4$임을 확인할 수 있다. $p_g = p_{data}$일 때,

$C(G) = \frac{1}{2}$이 최적의 값임을 나타내기 위해서 다음과 같이 나타낼 수 있다.

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

그리고 $C(G)$로부터 이 표현식을 빼면 다음을 얻을 수 있다.

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

KL은 두 확률분포의 차이를 계산하는데 사용되는 함수로 쿨백-라이블러 발산을 나타낸다.
위 식에서 모델의 분포와 데이터 생성 프로세스 간의 JSD(Jensen-Shannon divergence)을 확인할 수 있다.

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

두 분포간의 JSD는 항상 양수이기 때문에(두 분포가 일치할때는 0) $C^* = -\log(4)$는 $C(G)$의 global minimum이 된다.

Convergence of Algorithm1

proposition 2.

만약 G와 D가 충분히 학습되었고, 판별자가 주어진 생성자에 대해서 최적의 값에 도달하였으며, p_g가 업데이트되어 기준을 개선한다면 p_g는 p_{data}로 수렴한다.

Proof)

$V(G, D) = U(p_g, D)$를 p_g의 함수라고 생각해보자. ($U(p_g, D)$는 p_g에서 convex하다. Convex한 함수 최댓값의 하방 도함수(subderivative)에는 최댓값이 도달하는 지점에서의 함수의 도함수가 포함된다.
따라서, $f(x) = f_\alpha(x)$이고, 모든 α에 대해 $f_\alpha(x)$가 x에 대해 convex 하다면, $\beta = \arg\sup_{\alpha \in A} f_\alpha(x)$일 경우 $\partial f_\beta(x) \in \partial f$이다. 이는 corresponding G가 주어졌을 때 최적의 D에서 p_g에 대한 경사 하강 업데이트를 진행하는 것과 동등하다. 따라서, $\sup_D U(p_g, D)$는 p_g에 대해 convex하며, 유일한 전역 최적값이 존재하므로, p_g가 p_x로 수렴한다.

Advantages and disadvantages

GAN 프레임워크는 이전 모델링 프레임워크와 비교하여 장단점이 있다.

단점)

1. $p_g(x)$의 명시적인 표현이 없다.
2. D가 훈련 중에 G와 잘 동기화되어야 한다.

장점)

1. Markov Chains가 필요하지 않다.
2. 그래디언트를 계산하기 위해 역전파만 사용된다.
3. 다양한 함수를 모델에 통합할 수 있다.
4. 학습 중 Inference가 필요하지 않다.