

GANs NIPS2014, Ian Goodfellow review

ToBigs 19기 오유진

1. Why study generative modeling?

- A. 고차원 probability distribution을 찾는 것 보다 학습 데이터의 분포를 학습하여 유사한 sample을 생성하는 것을 목표로 함 (training & sampling)
- B. 이는 강화학습에 여러가지 방법으로 사용될 수 있음
- C. Missing data 역시 학습하여 model generation을 도울 수 있음
- D. Multi-modal 학습에 용이하게 사용됨
 - i. 전통적인 방법인 MSE가 하나의 정답을 갖는 것과 달리 GAN은 하나의 input이 다수의 output을 가질 수 있음
- E. 활용 예시 : Single Image Super-resolution(고화질 변환), Introspective Adversarial Networks(대화형 사진 편집), Create art, Image-to-Image translation

2. How do generative models work? How do GANS compare to others?

- A. MLE: Maximum Likelihood Estimation
주어진 training 데이터를 가장 잘 설명하는 모수(model)를 찾는 것.

MLE는 모수(θ)를 추정하기 위한 추정량(θ^*)이다.

모수를 추정하는 방법은 다양한 방법이 있는데, 그 중 MLE는 가능도(Likelihood)를 최대화(Maximum) 하는 값으로 추정하고, 그 값을 최대가능도 추정량(MLE)이라고 하는 것이다.

가능도 함수(Likelihood Function)는 확률밀도함수를 다르게 바라본 함수이다.

확률밀도함수 $f(x)$ 는 고정된 모수에 대하여 x 의 함수이다.

확률 밀도함수를 구하는 문제는 모수를 주고, 확률값을 계산하도록 한다.

반면, 가능도함수는 주어진 자료(x)가 얻어질 가능성을 추정량에 대한 함수로 나타낸 것이다. 주어진 자료를 가장 잘 설명하는 모수의 추정량을 찾아 최대가능도 추정량(MLE)으로 삼는다.

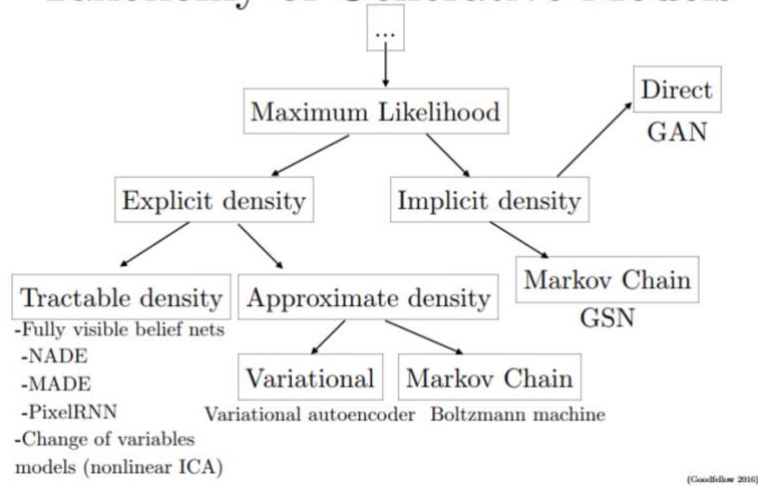
최대가능도 추정량은 가능도함수의 최대값을 가지는 값을 추정량으로 하는 방법이다. 가능도 함수의 y 축은 가능도($L(\theta^*)$), x 축은 추정량(θ^*)이다.

이때, 로그함수가 단조증가함수이므로. 가능도 함수를 최대화하는 θ 는 \log 가능도 함수를 최대화하는 θ 와 같다.

서로 다른 모수에 대한 추정량의 집합으로 각각 다른 모델을 만들 수 있다. 여러 모델 중 가능도함수를 최대화하는 추정량(θ^*)을 가진 모델을 찾는 것이 목표이다.

B. 생성 모델의 가계도

Taxonomy of Generative Models



C. How Gans compare to others?

- i. Latent code: 잠재 코드를 사용하여 나중에 생성되는 것을 설명
- ii. Asymptotically consistent(Unlike variational methods)
- iii. No Markov chains needed (Unlike GSNs)
- iv. Often regarded as producing the best samples
 1. No good way to quantify this

3. How do GANs work?

GAN : Generative Adversarial Networks

두 모델이 가지는 Adversarial(적대적) 관계를 이용하여 서로 발전하는 생성 모델

A. Adversarial Relation

- i. generative model
 - 분별모델을 상대로 완벽한 속임수를 수행
 - 랜덤 노이즈를 multilayer perceptron 에 넣어 모방 데이터를 만든다.
- ii. discriminative model
 - 실제 데이터와 생성 모델이 만들어낸 데이터를 구별

- multilayer perceptron 에 데이터를 넣어 모방 데이터인지 진짜 데이터인지 구분한다.

➔ 경쟁은 generative model 이 실제 데이터와 구별하기 힘들 정도로 모방 데이터를 잘 만들 때까지 계속된다.

B. The Training Process

- minibatch of x 와 minibatch of z 가 동시에 실행
 - minibatch: 학습을 진행하면서 여러 번 시행하는데, 한 개의 batch 사이즈가 너무 크면 연산 시간이 너무 길어 이러한 문제를 해결하기 위해 batchsize 를 더욱 줄여서 시행하는 방법
- x (data set)는 Discriminator 의 cost function 을 줄이는 방향으로 파라미터를 업데이트
- z (latent variable)는 Generator 의 cost function 을 줄이는 방향으로 파라미터를 업데이트
- 이때, 비용 함수를 줄이는 최적화 알고리즘은 여러 가지가 있지만 여기서는 Adam 이라는 optimizer 알고리즘을 사용

C. Cost functions

GANs framework 에는 여러 가지 cost function 방법이 있다

- The discriminator's cost

$$J^{(D)}(\theta^{(D)}, \theta^{(G)}) = -\frac{1}{2} \mathbb{E}_{x \sim p_{\text{data}}} \log D(x) - \frac{1}{2} \mathbb{E}_z \log (1 - D(G(z))).$$

- discriminator 의 loss function 은 binary cross entropy loss function 을 사용한다. (위 공식) 위 공식은 Discriminator 의 cost function 함수인데, 두 군데의 minibatch 데이터를 입력받는다.
 - 한 곳은 dataset 에 존재하는 real 이미지 ($x \sim p(\text{data})$)
 - 또 다른 한 곳은 generator 가 생성한 fake 이미지 (z)
- $\log D(x)$ 부분에서 $D(x)$ 는 1로 최대화할 수 있도록 해야 한다.
 - $1 \rightarrow \text{real image}$
- 반면에 $\log (1-D(G(z)))$ 이 부분에서 $D(G(z))$ 는 0으로 minimize 해야 한다.

- $0 \rightarrow$ fake image

ii. Minimax

- 위 공식은 discriminator 에 대해서만 정리한 것으로 실제로는 generator 와 discriminator 두 부분 모두 cost function 이 정의되어야 한다.
- GAN 은 optimizer 문제이기보다는 서로 상대방이 싸우는 게임과 가깝다. 결국 generator 와 discriminator 의 합은 0 이라고 볼 수 있다.

iii. Heuristic, non-saturating game

- Minimax 를 적용하면 이론처럼 실제로는 잘 작동하지 않는다.
- 왜냐하면 Discriminator 의 정확도가 거의 완벽에 가까워진다는 말은 더 이상 generator 가 잘 작동할 수 없다는 의미와도 같기 때문이다.
- Minimax 에서는 Generator 가 discriminator 가 real 이미지를 찾을 수 있는 확률을 최소화하는 방향으로 진행했지만, 반대로 discriminator 가 fake 이미지를 찾을 수 있는 확률을 최대화하는 방향으로 변경한다.

iv. Comparison of cost functions

- 특히 GAN 의 $J(G)$ 는 training data 에 직접 접근하지 않고, training data 에 관한 모든 정보를 discriminator 로부터 전달받는다.

4. How to evaluate GANs?

GANs은 객관적인 평가 기준이 없다. 그렇다면 평가할 수 있는 방법에 대해 알아보자.

A. Manual Evaluation

- 직접 사람이 성능을 평가한다.
- AMT를 사용하여 각각 epoch마다 생성된 이미지를 평가한다.
- 평가자의 주관성이 개입될 여지가 있다.

B. Qualitative Evaluation

i. Rating and Preference Judgement

인간 심판은 생성된 이미지의 순위를 매긴다

ii. Rapid Scene Categorization

이미지 평가 시간을 매우 짧게 제한하여 평가한다.

iii. Nearest Neighbors

가장 유사하게 생성된 이미지를 찾는다. Euclidean distance를 이용한다.

C. Quantitative Evaluation

i. Average Log-Likelihood

생성자가 얼마나 훈련 이미지의 분포를 설명하는지 추정한다.

고차원 공간에서 평가하기 쉽지 않다.

ii. Inception Score

이미지의 Fidelity(질), Diversity(다양성)을 평가한다.

- 먼저, Inception V3을 이용하여 생성된 이미지를 분류하고, 생성 이미지가 학습 이미지와 닮은 정도에 대한 라벨 확률분포로 반환한다. 이미지 분류기를 통해 생성 이미지의 속성을 측정할 수 있다.
- 그 다음엔 생성 이미지들에 대한 라벨 확률분포를 결합한다. 결합한 분포를 marginal 분포라고 한다. 이 분포는 생성기의 다양성을 평가하는 지표이다. 최종적으로, 각 이미지가 각각의 Label에 맞게 분류되고, 동시에 집합적으로 다양성을 갖는지 평가한다. 이때, KL-Divergence를 이용하여 두 확률분포가 얼마나 다른지 확인할 수 있다.
- 가장 이상적인 경우는 높은 KL-Divergence를 가질 때이다. 이를 Inception Score라고 한다.
- 반면, Inception Score은 훈련데이터셋에 없는 생성 결과는 레이블이 잘 되지 않으며, 훈련데이터와 다른 레이블셋을 사용하여 생성하는 경우 다양성 분포를 추정하기 어려우며, 분류기의 이미지 품질 개념이 인간의 관점과 다르다는 점 등 다양한 한계가 있다.

iii. Frechet Inception Distance

실제 이미지와 생성된 이미지에 대해 두 그룹의 유사도를 측정한다.

유사할 수록(FID 점수가 낮을 수록) 성능이 좋다.

- IS를 개선하기 위해 고안된 방법이다. IS는 생성 이미지만으로 성능을 평가하는 반면, FID는 실제 이미지 모음 통계와 생성된 이미지 모음 통계를 비교한다.
- 이미지의 평균과 공분산을 계산하여 multivariate Gaussian 분포로 요약하고, FD 거리를 계산한다.
- FID는 pretrained된 Inception v3에서 출력 레이어를 제거하고 출력이 마지막 풀링 레이어의 활성화(activation)된 값을 사용한다. 이 출력 레이어에는 총 2,048개의 활성화(activation)이 있으므로, 각 이미지는 2,048개의 활성화 특징으로 예측이 된다.
- 먼저, 실제 이미지가 표현되는 방법을 참조하기 위해 생성하고자 하는 도메인의 실제 이미지 모음에 대한 2,048개의 특징 벡터를 구한다. 그리고 생성된 이미지에 대한 특징 벡터도 구한다.

$$d_2((m,C),(m_w,C_w))=\|m-m_w\|^2_2+\text{Tr}(C+C_w-2(CC_w)^{1/2}).$$

- 이렇게 구한 벡터들을 활용해 위의 식과 같은 논문의 방정식을 거쳐 FID 점수를 구한다. 위의 식에서 점수를 d^2 로 표현하며, 이는 곧 거리(distance)이며 제곱 단위이다.
- m 과 m_c 는 각각 실제 이미지 및 생성된 이미지의 기능별 평균(feature-wise mean)을 나타내며 이때 2,048개의 벡터에서 각 요소는 이미지에서 관찰되는 평균 특징이다.