**Generative Adversarial Nets**

한현수

**구현? 왜 G마지막에서 tanh activation을 사용하는지?**

**Abstract**

적대적 process(경쟁하는 process)를 통한 생성 모델을 제안.

* G = generative model: data distribution을 capture
* D = discriminative model: G와 training data를 추정하는 classifier

이를 학습하는 방법은 G는 D가 잘못 분류할 확률을 maximize하는것이다.   
(minimax two-player game)  
G가 training data와 유사하게 만들어 D의 추정 확률 분포를 0.5로 만든다.  
G와 D가 Multilayer perceptrons(MLP)로 구현된다면 이는 backpropagation으로 학습 가능하다.

**1. Introduction**

Deep learning은 고차원의 sensory input을 mapping하여 분류하는 모델을 사용하였다.   
ex) well-behaved gradient를 가지는 선형 활성화 함수를 기반으로 한 backpropagation, dropout사용

Deep generative model은 maximum likelihood estimation과 같은 확률 연산을 근사하는데 어려움이 있고, generative context에서 선형 유닛의 이점을 사용하기에 어려움이 있다.

G는 랜덤 노이즈를 MLP를 통과시켜 sample을 만들게 된다.

D는 model이 생성한 분포와 기존의 data분포를 결정함으로 학습한다.

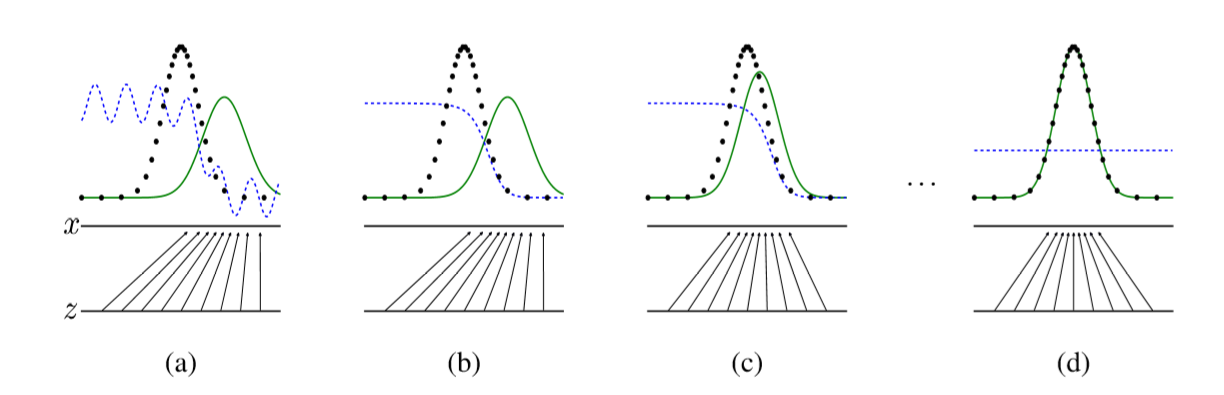
즉 D는 진짜 데이터인지를 구별, G는 가짜 데이터를 만들어서 학습하는 것이다.

**3. Adversarial nets**

****

이 수식을 보면 첫 번째 항은 real data를 D에 넣었을 때 나오는 결과에 log를 취한 값이고, 두번째 항은 fake data 즉, G(z)를 D에 넣었을 때에 1-output의 log를 취한 값이다.  
그래서 첫번째 항은 D가 real data를 참으로 판별할 때 값이 커지게 되고, 두번째 항은 G가 fake data를 참으로 판별하면 -inf에 가깝게, 거짓으로 판별하면 0에 가깝게 되는 것이다.  
이처럼 D에게는 V(D, G)를 최대화, G에게는 V(D, G)를 최소화하도록 하는 것이 목표가 된다.  
그래서 본 논문에서는 D와 G 두 player간의 최대화 최소화를 뜻하는 two-player minmax game으로 표현한다.  
학습 초기에 G의 성능이 좋지 않기 때문에 D가 잘 구별하여 log(1-D(G(z)))가 잘 변하지 않게 된다. 그러므로 log(1-D(G(z)))를 최소화하는 것이 아닌 log(D(G(z)))를 최대화할 수 있도록 학습하는 것이 좋다고 한다. 그 이유는 log(1-D(G(z)))의 gradient가 너무 작은 값이라 학습이 느리기 때문이라고 한다. (log(x) 그래프와 log(1-x)그래프를 생각하면 편하다.)

**4. Theoretical Results**



D의 분포(acc): 파란색 점선, real data: 검정색 점선, (G) fake data: 녹색 실선  
(a): 초기상태로 real data와 fake data의 분포가 전혀 달라 G의 성능이 좋지 않고, D 또한 학습이 제대로 되지 않은 상태이다.  
(b): G는 학습이 덜 되어 real data와 fake data의 분포가 다르고, D가 먼저 어느정도 학습되어 잘 구분을 하고 있다.  
(c): D가 어느정도 학습되면 G가 실제 데이터 분포와 유사하게 되는 방향으로 학습을 하게 된다.  
(d): 위의 반복으로 real data와 fake data의 분포가 유사해져 D가 구분을 못하고 확률을 0.5로 계산하게 된다.

이 과정으로 real data와 fake data를 구별할 수 없을 정도로 G가 fake data를 생성해내게 된다.

**Algorithm 1** : GAN의 학습에는 Minibatch stochastic gradient descent가 사용됨. k는 hyperparameter임. 이 논문에서는 k=1을 사용함

**For** number of training iterations **do**

**For** k steps **do**