# 연구실 실습일지	회차	7/15
	작성자 (학번)	김재희 2015038195
	작성일	2019.10.16

교과목명	컴퓨터공학연구실심화실습2	수업번호	24488
실습 일시	2019 년 10 월 16 일 시 부터 ~ 시 까지	실습진행시간	
실습 장소	한양대학교 1 공학관		
실습 내용	(see below)		

7주차 주제 : Generative Adversarial Network

7주차에는 2014년 6월 Ian J.Goodfellow 등이 발표한 gan 논문을 읽어보고, 정리를 해보았습니다.

1) Abstract

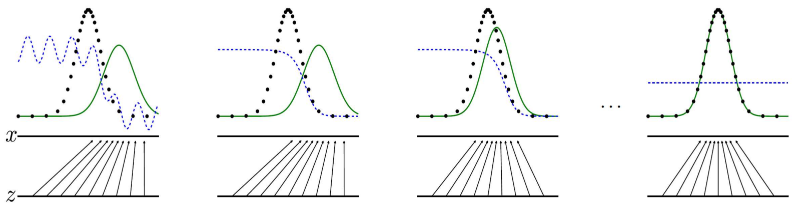
- generative model G : training data distribution을 흉내내려하는 모델
- discriminator model D : G가 아니라, training data에서 온 것인지 파별하는 모델
- G는 training data의 distribution을 학습하여, 임의의 noise를 training data와 같은 distribution으로 생성하고, D는 해당 input이 생성된 이미지인지, training data로부터 나온 이미지인지에 대한 확률이 1/2이 되도록 한다.

2) Introduction

- D는 training data distribution인지 G에서 온 것인지 판별하는 것을 학습한다.
- G는 위조지폐를 만드는 팀과 유사하고, D는 위조지폐를 감지하는 경찰과 유사하다.

3) Adversarial nets

- data x에 대한 G의 분포 Pg를 학습하기 위해 input noise Pg(z) 정의한다.
- G는 θg를 갖는 mulitilayer perceptron에 의해 표현되는 미분가능한 함수
- D(x)는 x가 Pg가 아닌 데이터에서 나온 확률
- $\min_G \max_D V(D,G) = E_{x \sim P_{data}(x)}[\log D(x)] + E_{z \sim P_z(z)}[\log(1 - D(G(z)))]$
- G는 V를 최소화하려하고, D는 V를 최대화하려 한다.
- E는 기댓값 , $x \sim P_{data}(x)$는 x가 training data distribution에서 왔을 때
- D입장에서 D(x)는 1에 가깝고, D(G(z))는 0에 가까워야 한다. 그러면 V의 최댓값은 0 이 된다.
- G가 완벽하게 생성하면 D(x)=1/2이다. G의 입장에서 V의 최솟값은 $-\infty$
- 학습시킬 때, inner loop에서 D를 최적화하면 overfitting 될 수 있다.
- 대신에, k step만큼 D를 최적화하고, G는 1step 만 최적화한다.
- 학습 초기에 G가 형편없기 때문에 log(1-D(G(z)))를 최소화하는 것보다, log(D(G(z)))를 최대화 하도록 하는 것이 더 학습이 잘 된다.



(a) (b) (c) (d)

- 검정 : Px 파랑 : D 초록 : Pg(G)

a) 초기상태

b) D가 data와 sample 구별하도록 학습 D = $\frac{P_{data}(x)}{P_{data}(x)+P_g(x)}$ 로 수렴

c) Pg 가 Pdata와 가까워진다.

d) Pg = Pdata가 되어 D(x) = 1/2

4) Theoretical Results

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.

- Proposition 1. Global Optimality of Pg = Pdata

G가 고정된 경우, D는 $D^*{}_G(X) = \dfrac{P_{data}(X)}{P_{data}(X) + P_g(X)}$ 이다.

proof)

어떤 G가 주어지면 D의 학습기준은 V(G,D)를 최대화하는 것

$$\begin{aligned} V(G,D) &= \int_x P_{data}(x) \log(D(x))dx + \int_z P_z(z) \log(1 - D(g(z)))\, dz \\ &= \int_x P_{data}(x) \log(D(x)) + P_g(x) \log(1-D(x))dx \qquad (3) \end{aligned}$$

$$\begin{aligned} C(G) &= \max_D V(G,D) \\ &= E_{x\sim P_{data}}[\log D^*_G(x)] + E_{z\sim P_z}[\log(1 - D^*_G(G(z)))] \\ &= E_{x\sim P_{data}}[\log D^*_G(x)] + E_{x\sim P_g}[\log(1 - D^*_G(x))] \\ &= E_{x\sim P_{data}}\left[\log \frac{P_{data}(x)}{P_{data}(x) + P_g(x)}\right] + E_{x\sim P_g}\left[\log \frac{P_g(x)}{P_{data}(x) + P_g(x)}\right] \qquad (4) \end{aligned}$$

- Theorem 1. C(G)의 global minimum은 Pg=Pdata인 경우에만 달성된다. 이 때, C(G) = - log4

  proof)

  Pg=Pdata, $D^*_G(X) = 1/2$. C(G)=log(1/2)+log(1/2) = - log4

$$E_{X \sim P_{data}}[-log2] \ + \ E_{x \sim P_g}[-log2] \ = \ -log4$$

  $C(G) = V(D^*_G G)$로부터 이 식을 빼면 다음과 같은 식을 얻는다.

$$C(G) = -log(4) + KL\left(P_{data} \| \frac{P_{data} + P_g}{2}\right) + KL\left(P_g \| \frac{P_{data} + P_g}{2}\right) \qquad (5)$$

$$C(G) = -log(4) + 2 \cdot JSD(P_{data} \| P_g) \qquad (6)$$

  C* = - log4 가 C(G)의 global minimum이고 유일한 해답은 Pg=Pdata 이다.

cf)

- Kullback-Leibler Divergence : 서로 다른 확률 분포의 차이르 추정하는 척도. 추정한 확률 분포와 실제 확률 분포 사이의 차이가 작다면 좋은 추정
- 정보량을 얼마나 잘 보존하는지 측정
- 원본데이터가 가지고 있는 정보량을 잘 보존할수록 원본데이터와 비슷한 모델
- item S_i의 정보량 $I_i = -\log(P_i)$
- 정보량 차이 $\Delta I_i = \log(P_i) - \log(Q_i)$
- P에 대하여 이러한 정보 손실량의 기댓값을 구한다.
- $D_{KL}(P \| Q) = E[\log(P_i) - \log(Q_i)] = \sum_i^N P_i \log\frac{P_i}{Q_i}$
- Symmetric 하지 않다. => $D_{KL}(P \| Q) \neq D_{KL}(Q \| P)$
- Kullback-Leibler Divergence을 Symmetric 하게 개량한 Jensen-Shannon Divergence
- $JSD(P \| Q) = \frac{1}{2} D_{KL}(P \| \frac{P+Q}{2}) + \frac{1}{2} D_{KL}(Q \| \frac{P+Q}{2})$

5) Advantages and disadvantages

- Advantages
  - 마르코프 체인이 필요없이, gradients를 얻기 위해 back-propagation만 사용
  - 특별히 추론이 필요없다.
  - 다양한 함수들이 모델에 접목될 수 있다.
  - 마르코프 체인을 썼을 때에 비해 훨씬 sharp한 결과를 얻을 수 있다.

- Disadvantages
  - Pg(x)가 명시적으로 존재하지 않는다.
  - D는 G와 균형을 잘 맞춰서 성능 향상되어야 한다.

6) Conclusions and future work

- conditional generative model로 발전시킬 수 있다.
- learned approximate inference는 x가 주어졌을 때 z를 예측하는 보조 네트워크를 학습함으로써 수행될 수 있다.
- parameters를 공유하는 조건부 모델을 학습함으로써 다른 조건부 모델을 대략 모델링 할 수 있다.
- Semi-supervised learning에도 활용 가능하다.
- G와 D를 조정하는 더 나은 방법이나 학습하는 동안 sample z에 대한 더 나은 distributions을 결정하는 등의 방법으로 속도를 높일 수 있다.

- 참고

https://arxiv.org/pdf/1406.2661.pdf
https://www.youtube.com/watch?v=odpjk7_tGY0&t=1607s
https://www.slideshare.net/NaverEngineering/1-gangenerative-adversarial-network

담당 교수명	문영식 교수님		
담당교수 의견		담당 교수 확인	(인)