E-Deep 2회차 발표 (1/14) 논문 리뷰

"Generative Adversarial Nets"

Abstract

이 논문에서는 2가지 모델을 동시에 학습시킨다 : Generative model인 G, Discriminative model인 D 이다.
Training 방식은 G는 D가 실수를 하도록, (G가 만들어낸 sample과 진짜 Data인 training data를 구분하지 못하도록,) 이루어진다. G와 D가 마치 minmax 게임을 하는것처럼 진행된다.
임의의 G, D에 대해 unique solution이 존재한다.

1. Introduction

G : Generative Model → Training Data의 분포를 파악해서 이미지들을 만들어냄.
D : Discriminative Model → G가 만든 sample과 Data를 구분.

이제껏, (~2014) 딥러닝은 Discriminative Model 계에서 큰 성공을 이루어왔다.
(High-dimensional data를 class Label로 매핑)
이 딥러닝 모델은 최적화과정에서 Dropout과 Backpropagation을 베이스로 한다. 이 최적화는 piecewise linear units을 사용하는데,
Deep Generative models은 piecewise linear units을 사용하기 힘들고, 다루기힘든 확률 계산으로 인해 Dropout과 Backpropagation의 영향이 적다.

이 논문에서 제안하는 Adversarial Nets 에서는, Generative model이 Discriminative model과 대립하게 된다.

비유> G : 위조 지폐 만드는 범죄자
D : 위조 지폐 탐지하는 경찰.
) 시간이 갈수록 G는 더 위조지폐를 잘 만들고,
D는 더 구분하는 능력 향상

⇒ 나중에는 G가 만든 위조지폐와 진짜 지폐가 구분하기 힘들 정도로 유사해진다!

(2. 생략)

3. Adversarial Nets

Data x 에 대해 generator의 분포 P_g 을 러닝하기 위해, input 노이즈 변수 $P_z(z)$ 에 대해 prior를 정의함다. ~~그러면~~,
그러고나서 Data space 로의 매핑을 $G(z;\theta_g)$ 로 나타낸다.
(G: Parameter θ_g 로 표현된 multilayer perceptron 으로 나타내진 미분가능한 함수).

또한, second multilayer perceptron $D(x;\theta_d)$ 도 정의한다.
(output: a scalar. i.e. input이 진짜 Data인지 아닌지).
($D(x)$ 는 데이터가 (x) P_g ~~가 아닌~~ 로부터 오지 않은 확률.)

Training ① D: trained to maximize probability
(데이터가 G에서 왔는지 training data에서 왔는지 올바른 라벨을 뽑는 확률)
② G: $\underbrace{\text{minimize}}_{\text{trained to}}$ $\log(1-D(G(z)))$ 즉, $D(G(z))$가 커지도록.
($G(z)$를 D가 data label로 표기하도록)

식으로 표현하면,

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

< Algorithm 1 >

k : a hyperparameter

for number of training iterations do

  for k steps do

  - sample minibatch of m noise samples $\{z^{(1)}, \cdots, z^{(m)}\}$ from noise prior $p_g(z)$ ← 만든 데이터이므로 noise sample로 표현
  - sample minibatch of m examples $\{x^{(1)}, \cdots, x^{(m)}\}$ from data generating distribution $p_{data}(x)$. [진짜 Data들
  - Update D (using stochastic gradient) "ascending" ← D에 대해 Maximize

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

  end for

- sample minibatch of m noise samples $\{z^{(1)}, \cdots, z^{(m)}\}$ from $p_g(z)$.
- Update G "descending" ← G에 대해 Minimize

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

end for

4. Theoretical Results

$\min_G \max_D V(G, D)$ 가 global optimum 을 가진다. ($p_g = p_{data}$)

4.1. Global optimality of $p_g = p_{data}$.

(먼저 G fixed 상태에서 D를 K steps 학습시키므로)

<Proposition 1> For G fixed, the optimal discriminator D is

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

pf) Given G, to maximize $V(G,D)$,

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

$\mathbb{E}_{x\sim p(x)}[q(x)] = \int p(x)q(x)\,dx$

$$\overset{\because}{=} \int_x p_{data}(x) \log D(x) + \int_z p_z(z) \log(1 - D(G(z)))$$

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

<u>Lemma</u>

For $\forall (a,b) \in \mathbb{R}^2 \setminus \{0,0\}$, the function $y \rightarrow a\log(y) + b\log(1-y)$ 는 $y = \frac{a}{a+b}$ 에서 maximum을 가진다. in $[0,1]$.

$$\left(\frac{d}{dy}(a\log y + b\log(1-y)) = \frac{a}{y} + \frac{-b}{1-y} = \frac{a - ay - by}{y(1-y)} = 0 \text{ when } y = \frac{a}{a+b} \right)$$

위에서 a는 $p_{data}(x)$, b는 $p_g(x)$ 다. $p_{data}(x)$ 와 $p_g(x)$ 모두 0이 되지않으므로

(즉, $Supp(p_{data}) \cup Supp(p_g)$ 안에서 정의됨)

(참고) $\mathrm{supp}(f) = \overline{\{x \mid f(x) \neq 0\}} \supset \{x \mid f(x) \neq 0\}$.

위의 Lemma에 의해서,

$$V(G,D) = \int_x \underbrace{p_{data}(x)\log D(x) + p_g(x)\log(1-D(x))}_{D(x)=\frac{p_{data}(x)}{p_{data}(x)+p_g(x)} \text{ 에서 Maximum을 가짐}}\, dx$$

G는 fixed / proposition 1 에서 D는 max값으로 D_G^*을 가진다.

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

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

thm 1

The global minimum of $C(G)$ is achieved

$$\Longleftrightarrow p_g = p_{data}$$

At that point, $C(G) = -\log 4$.

pf) ($\Leftarrow$) For $p_g = p_{data}$, $D_G^* = \frac{p_{data}}{p_{data}+p_g} = \frac{1}{2}$.

Hence, $C(G) = \mathbb{E}\left[\log\frac{1}{2}\right] + \mathbb{E}\left[\log\frac{1}{2}\right] = 2\times\log\frac{1}{2} = -\log 4$

($\Rightarrow$)

$C(G) = V(G, D_G^*)$

log2 + log2

$$C(G) - (-\log 4) = \mathbb{E}_{x\sim p_{data}}\left[\log\frac{p_{data}}{p_{data}+p_g}\right] + \mathbb{E}_{x\sim p_g}\left[\log\frac{p_g}{p_{data}+p_g}\right] - (-\log 4)$$

$$= \mathbb{E}_{x\sim p_{data}}\left[\log\frac{p_{data}\times 2}{p_{data}+p_g}\right] + \mathbb{E}_{x\sim p_g}\left[\log\frac{p_g\times 2}{p_{data}+p_g}\right]$$

$$C(G) - (-\log 4) = \mathbb{E}_{x\sim P_{data}}\left[\log \frac{2P_{data}}{P_{data}+P_g}\right] + \mathbb{E}_{x\sim P_g}\left[\log \frac{2P_g}{P_{data}+P_g}\right]$$

$$= \underbrace{KL\left(P_{data} \,\|\, \frac{P_{data}+P_g}{2}\right) + KL\left(P_g \,\|\, \frac{P_{data}+P_g}{2}\right)}_{= 2JSD(P_{data} \| P_g)}$$

$$KL(P \| q) = \mathbb{E}_{x\sim P(x)} \log \frac{P(x)}{q(x)}$$

$$\left(JSD(p \| q) = \frac{1}{2} KL(P \| q) + \frac{1}{2} (q \| M) \right)$$

$$\Rightarrow \quad C(G) = -\log 4 + 2JSD(P_{data} \| P_g)$$

JSD 는 항상 nonnegative 이고, 두 분포가 같을 때만 0 이다.

즉, $C(G) = -\log 4 + 2JSD(P_{data} \| P_g) \geq -\log 4$ ($P_{data} = P_g$일 때 등호성립)

즉, $C(G)$가 global minimum을 가졌다는 건 $P_{data} = P_g$ 를 뜻한다. ∎

< Proposition 2 >

If G and D have enough capacity, at each step of Algorithm 1, D is allowed to reach its optimum given G and P_g is updated so as to improve the criterion

$$\mathbb{E}_{x \sim P_{data}} [\log D_G^*(x)] + \mathbb{E}_{x \sim P_g} [\log (1 - D_G^*(x))]$$

$\Rightarrow P_g \rightarrow P_{data}$. 즉, Algorithm 1의 Iterative 방식으로 Generator G가 Data의 분포들에 수렴함.

Pf) Consider $V(G, D) = U(P_g, D)$

<u>Note</u> $U(P_g, D)$ is convex in P_g. (Global optimum이 존재)

if $f(x) = \sup_{\alpha \in A} f_\alpha(x)$ and $f_\alpha(x)$ is convex in x $\forall \alpha$,

then $\partial f_\beta(x) \in \partial f$ if $\beta = \arg\sup_{\alpha \in A} f_\alpha(x)$.

$\sup_D U(P_g, D)$ is convex in P_g with a unique global optima.
($-\log 4$ when $P_g = P_{data}$)

∎

실제 실험에서는 θ_g를 업데이트 시킴으로써 P_g를 optimize. (즉, G의 parameter를 Update)

& Multilayer perceptron을 사용하면 Critical points가 여러개 나온다. (Min, Max 포인트들 X)
그래도 성능은 잘 나온다!

5. Experiments

G는 ReLU, sigmoid 를 섞어서 사용한다. D는 maxout 사용.

또한, D를 학습시키는 데에 Dropout 이 사용된다.

6. Advantages and disadvantages

단점 : • No explicit representation of $P_g(x)$.

• D가 학습중에 G와 잘 동기화되어야함. 특히, D를 업데이트하지않고 G를 과하게 학습시키면 안됨.

장점 : • Markov Chain이 필요없다. Backprop 만 사용!

• 사용가능한 함수들 多

• No inference is needed.

• G가 Data로부터 바로 학습되는 것이 아니라 D에서 계산한 gradients 이용.

• sharp, degenerate 분포를 나타낼 수 있다. (MC 에서는 흐릿함).