23.07.13 _ 5주차

딥러닝 논문 요약 및 구현 스터디

발표자

김영동

Generative Adversarial Nets

Abstract

Abstract

We propose a new framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. In the space of arbitrary functions G and D, a unique solution exists, with G recovering the training data distribution and D equal to $\frac{1}{2}$ everywhere. In the case where G and D are defined by multilayer perceptrons, the entire system can be trained with backpropagation. There is no need for any Markov chains or unrolled approximate inference networks during either training or generation of samples. Experiments demonstrate the potential of the framework through qualitative and quantitative evaluation of the generated samples.

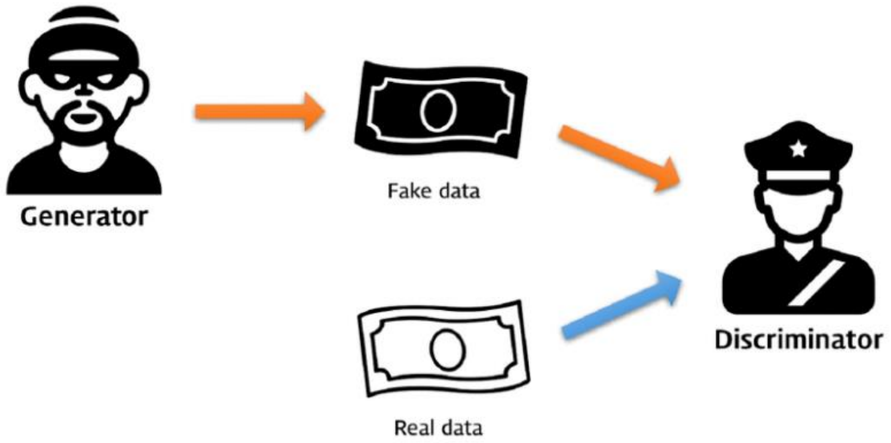
- **두가지 모델을 훈련시켜 Generative model을 추정하는 새로운 프레임워크 제안**
  - Generative model(생성자), G : 학습데이터의 분포를 모사함
  - Discriminative model(판별자), D : sample 데이터가 G로부터 나온 데이터가 아닌 실제 학습데이터로부터 나온 데이터일 확률을 추정
- **학습과정**
  - G는 D가 실수할 확률을 높이는 방향으로 학습(잘 구별하지 못하도록)
- **본 논문의 특징**
  - 본 논문에서 제안하는 프레임워크는 2인용 minimax 게임과 비슷.
  - 임의의 G, D의 공간에서, G와 D는 각각 유일한 답이 존재하며 모두 ½의 확률을 갖는다.
  - G, D가 다층퍼셉트론의 구조를 갖는다면 역전파를 이용한 학습이 가능하다.
  - 이는 데이터를 훈련하거나 생성할 때 다른 네트워크나 Markov chain이 필요없다.
  - 본 논문의 실험은 생성된 샘플에 대해 질적, 양적으로의 가능성을 보여줌.

Introduction

- 딥러닝은 다양한 분야(natural images, audio waveforms containing speech, and symbols in natural language corpora)에서 좋은 성능의 모델을 만들게 해줌.
- 이 중, discriminative model(많은 정보가 담긴 입력 데이터의 클래스 라벨을 매핑하는 모델)은 역전파와 드랍 아웃을 기반으로 높은 성능을 낼 수 있었다.
- 하지만, Deep generative model은 제어하기 힘든 수많은 probabilistic computations(MLE 등)을 추론하는 것에 대한 어려움, 생성 모델에서 부분적 선형 유닛의 이점을 이용하는 것의 어려움으로 인해 좋은 성능을 내지 못함. 즉, discriminative model을 만들 때와는 다른 방식의 접근이 필요하다.
- 따라서 본 논문에서는 위에서 언급한 어려움을 피해간 생성 모델을 제안한다.

Introduction

- **Adversarial nets framework**
  - Adversarial nets framework는 generative model, discriminative model을 모두 사용
    - Generative model : 판별자를 상대로 완벽한 속임수를 수행
    - Discriminative model : 실제 데이터와 생성자가 만들어낸 데이터를 구별
  - 위 컨셉의 과정을 경찰(판별자)와 위조지폐범(생성자) 사이의 경쟁으로 비유하면, 위조지폐범은 최대한 진짜 같은 화폐를 만드는 것이 목표이고, 경찰은 진짜 화폐와 가짜 화폐를 완벽히 판별하여 위조지폐범을 검거하는 것이 목표.



Related work

- **잠재 변수가 있는 무방향 그래픽 모델은 유방향 그래픽 모델의 대안**
  - RBM(restricted Boltzmann machines)
  - DBMs(deep Boltzmann machines)
  - 이러한 모델 내의 상호작용은 정규화되지 않은 포텐셜 함수의 곱으로 표현되며, 이는 무작위 변수의 모든 상태에 대한 글로벌 적분으로 정규화된다.
  - Partition 함수와 그 기울기는 MCMC로 추정할 수 있지만, 모두 파악하기는 힘들다.
- **하이브리드 모델**
  - DBNs는 하나의 무향 계층과 여러개의 유향 계층을 포함하는 하이브리드 모델
  - 빠르게 근사하는 레이어별 훈련 기준이 존재하지만, 무향 및 유향 모델 모두에서 계산의 어려움이 있다.
- **Log-likelihood를 근사화하거나 제한하지 않는 대체 기준**
  - Score matching
  - NCE(Noise-Contrastive Estimation)
  - 위 방법 다 학습된 확률 밀도를 정규화 상수까지 분석적으로 지정해야 한다.
  - 여러 레이어의 잠재 변수가 있는 생성 모델(ex, DBNs, DBMs)에서는 정규화되지않은 확률 밀도를 도출하는 것조차 불가능하다.
  - Denoising AE, contractive AE와 같은 일부 모델에는 RBM에 적용되는 score matching과 매우 유사한 학습 규칙이 있다.
  - NCE는 fixed noise distribution을 사용하기 때문에, 모델이 관측 변수의 작은 부분 집합에 대해 올바른 분포를 학습한 후 학습 속도가 급격히 느려진다.

Related work

- **확률 분포를 명시적으로 정의하는 것이 아니라 원하는 분포에서 표본을 추출하도록 생성 모델을 훈련시키는 방법**
  - 이 방법은 모델을 역전파로 훈련하도록 설계할 수 있다는 장점이 있다.
  - GSN(generative stochastic network)
  - 위 모델은 매개 변수화된 Markov 체인을 정의하는 것으로 볼 수 있다.
  - GSN과 비교해서 GAN은 샘플링을 위해 Markov 체인을 필요로하지 않는다.

Adversarial nets

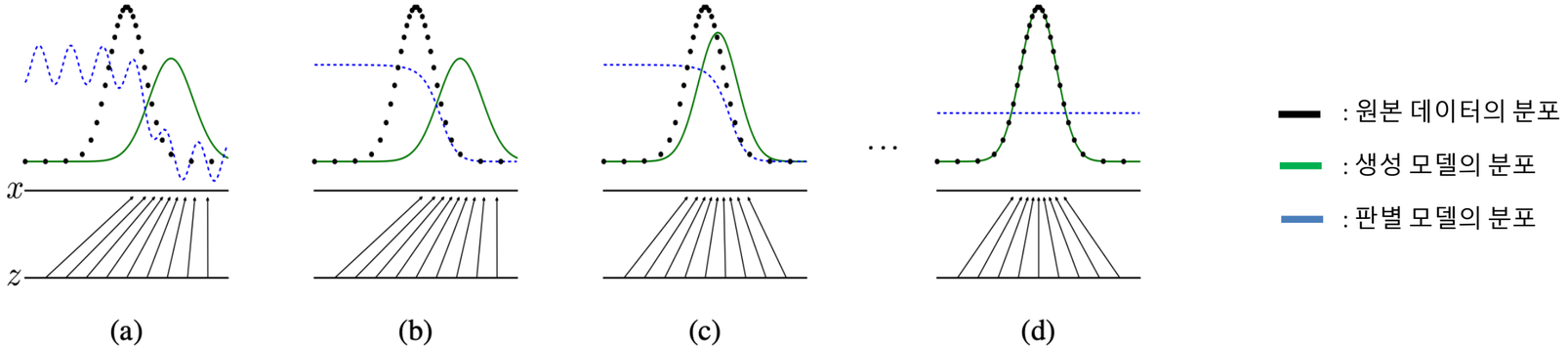
- **모델 G의 학습과정**
  - Generator의 분포 p_g를 x에 대해 학습시키기 위해 input noise 변수에 대한 사전분포인 $p_z(z)$를 정의
  - 노이즈변수의 데이터 공간의 매핑 $G(z;\theta_g)$을 만듬 (G: 미분가능한 다층 퍼셉트론)
  - 입력된 샘플이 p_g가 아닌 실제 데이터 분포에서 얻어졌을 확률을 계산하는 다층 퍼셉트론 $D(x;\theta_d)$를 정의
- **Objective function**
  - D는 실제 데이터와 생성된 데이터에 대해 적절한 label을 할당하도록 하는 확률을 최대화, $\log(1-D(G(z)))$를 최소화

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

  - 첫째 항 : 실제 데이터 x를 판별자에 넣었을 때 나오는 결과에 log를 취해 얻는 기댓값
  - 두번째 항 : 가짜 데이터 z를 생성자에 넣어 나온 결과를 판별자에 넣었을 때의 결과에 log(1-결과)를 취해 얻는 기댓값
  - 즉, **생성자**는 만드는 데이터가 그럴듯한 이미지로 만들기 위해 노력하고, **판별자**는 원본데이터에 대해 1을 뱉을 수 있게 노력함.

Adversarial nets

- **수렴과정**
  - 학습의 목표 : $p_g \rightarrow p_{data}, D(G(z)) \rightarrow 1/2$, 즉, 생성된 분포가 원본데이터의 분포와 유사하게 되어 판별자가 진짜와 가짜를 구분할 수 없어 항상 ½의 확률을 내뱉는 것



Theoretical Results

- **Global Optimality**

Proposition 1. *For G fixed, the optimal discriminator D is*

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

- **Proof:** For G fixed,

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

$$= \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x}))dx + \int_{z} p_{\boldsymbol{z}}(\boldsymbol{z}) \log(1 - D(g(\boldsymbol{z})))dz$$

$$= \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x})) + p_g(\boldsymbol{x}) \log(1 - D(\boldsymbol{x}))dx$$

z에서 x로 매핑되는 과정으로 볼 수 있음

$y = alog(y) + blog(1-y)$는 $\frac{a}{a+b}$에서 극댓값을 가짐, $y \in [0,1]$

따라서, V는 $D_G(x) = \frac{p_{data}(x)}{p_{data}(x)+p_g(x)}$일 때 최댓값을 가짐.

Theoretical Results

- **Global Optimality**

Global Optimality of $p_g = p_{\text{data}}$

- **Proof:** For G fixed,

$$
\begin{aligned}
C(G) &= \max_D V(G,D) \\
&= \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[\log(1 - D_G^*(G(\boldsymbol{z})))] \\
&= \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim p_g}[\log(1 - D_G^*(\boldsymbol{x}))] \\
&= \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(\boldsymbol{x})}{P_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] + \mathbb{E}_{\boldsymbol{x}\sim p_g}\left[\log \frac{p_g(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] \\
&= \mathbb{E}_{x\sim p_{data}}\left[log\frac{2*p_{data}(x)}{p_{data}(x)+p_g(x)}\right] + \mathbb{E}_{x\sim p_g}\left[log\frac{2*p_g(x)}{p_{data}(x)+p_g(x)}\right] - \log(4) \\
&= KL(p_{data}||\frac{p_{data}(x)+p_g(x)}{2}) + KL(p_g||\frac{p_{data}(x)+p_g(x)}{2}) - \log(4) \\
&= -\log(4) + 2\cdot JSD\left(p_{\text{data}}||p_g\right)
\end{aligned}
$$

$KL(p_{data}||p_g) = \int p_{data}(x)\log\left(\frac{p_{data}(x)}{p_g(x)}\right)dx$

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

JSD는 distance matric이기 때문에 최소값은 0. 즉, p_{data}와 p_g가 같을 때 C(G)의 최솟값은 –log(4)

따라서 p_{data}와 p_g가 같을 때가 global optimum

Theoretical Results

- **Algorithm 1**

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.

Experiments

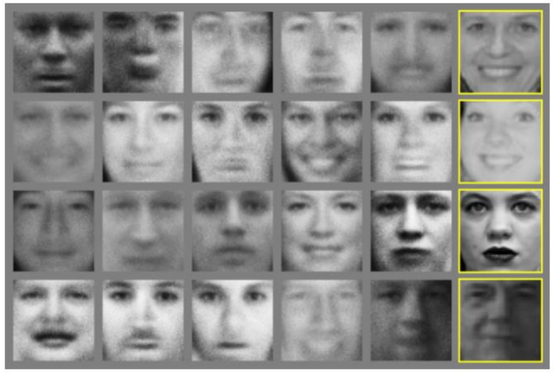
- MNIST, Toronto Face Database(TFD), CIFAR-10에 대해 학습 진행
- G는 ReLU, sigmoid 혼합하여 사용, D는 maxout activation 사용
- p_g에 있는 테스트 데이터들의 확률은 G로 생성된 데이터들에 Gaussian Parzen window를 맞추는 방식으로 측정

Model	MNIST	TFD
DBN [3]	138 ± 2	1909 ± 66
Stacked CAE [3]	121 ± 1.6	$\mathbf{2110 \pm 50}$
Deep GSN [6]	214 ± 1.1	1890 ± 29
Adversarial nets	$\mathbf{225 \pm 2}$	$\mathbf{2057 \pm 26}$

- 위 방식을 사용해 확률을 측정하는 것은 분산이 크고 이미지 등의 고차원 데이터에서 좋은 효과를 내지 못하지만 본 논문에서 사용가능한 방식 중에서는 가장 좋은 방식이었다고 한다.

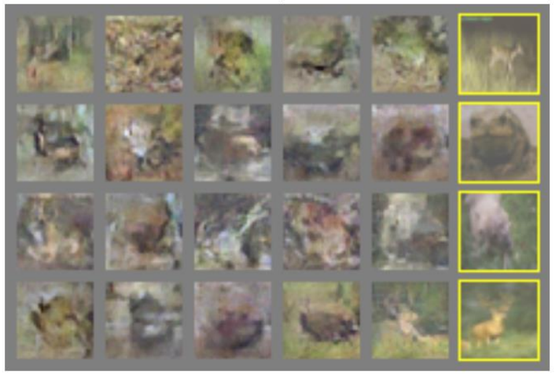
Experiments

a)



b)

c)



d)

Advantages and Disadvantages

- adversarial nets framework는 이전에 존재하던 것들과 비교했을 때 장점과 단점 모두를 가지고 있다.
- 단점
  - 생성된 데이터의 분포를 확실하게 표현하는 것이 없다는 것
  - 학습할 때 D는 G와 잘 동기화 되어야한다.
  - 즉, G가 D의 학습 없이 너무 많이 학습되어서는 안된다는 것.
  - 만약 D의 가중치를 조정하는 과정 없이 G만 계속해서 가중치를 조정할 경우, G는 p_{data}의 분포를 따르는 데 충분한 다양성을 지닐 수 없게 되는 'Helvetica scenario'에 빠지게 되어 원하는 G를 얻지 못하게 됨.
- 장점
- 역전파로 기울기를 구할 수 있기 때문에 Markov chain이 필요하지 않다.
- 학습 과정에 inference를 할 필요가 없다는 것.
- adversarial nets framework와 다양한 함수를 합칠 수 있다는 것.