

개발지식 (이론) 인공지능 개인정리

## PGGAN 개인정리(논문 리뷰 Progressive Growing of GANs for Improved Quality, Stability, an

약 1년 전 by Prowd Loner

- 논문은 제목에 나온 Progressive Growing of GANs for Improved Quality, Stability, and Variation를 리뷰합니다. GAN 모델 중에서도 유명한 논문이죠.
  - 제 정리글은 핵심만 간단히 정리할 것입니다.
- 한국어로 정말 잘 정리된 글로는 <https://sensibilityit.tistory.com/508>가 있습니다.
- 글 구성이나 설명을 인용하고 싶을 정도인데, 최근에는 논문을 보는 버릇을 길러야 겠다고 생각해서, 일단 한번 보고 이해한 후에 논문 구성을 따라가며 필요한 내용만 아래에 적는 것입니다.

### [ABSTRACT]

- 해당 논문은 GAN의 새로운 학습 방법을 제안한 것입니다.
- 키포인트로는 Generator와 Discriminator의 해상도를 점진적으로 늘려가는 방식으로, 결과 출력물의 해상도를 높이는 학습법이라고 합니다.
- 잠시 기존 GAN과 비교를 하자면,
- 해당 논문이 나온 시점에서 GAN은 고해상도 데이터 생성 학습이 어려운 상황이었다고 합니다.
- 해상도가 낮으면 GAN에서 핵심이 되는 Generator(이하 Gen)가 Discriminator(이하 Disc)를 속이기 쉽죠.
- 하지만 고화질이 될수록 Gen은 Disc를 속이기 힘들어지기에, 학습 속도도 매우 느리며, GAN의 특성상 통제 불능의 요소가 많을 것입니다.

이를 해결하기 위해 PRGAN이 한 것은,

먼저 낮은 해상도로 학습을 진행하고, 점진적으로 더 높은 해상도로 학습을 하길 반복하는 형식을 사용했고, 저는 이것이 학습의 방향을 정해준 작업이라 생각합니다.(우리도 공부할때 쉬운 문제부터 점진적으로 풀어나가다보면 보다 어려운 것들도 수월하게 풀수 있어지고, 오히려 이렇게 함으로써 전체 학습 속도가 더 빨라진다고 생각합니다.)

## [INTRODUCTION]

- 18년 기준 대표적인 Generative 모델은

autoregressive models (van den Oord et al., 2016b;c),

variational autoencoders (VAE) (Kingma & Welling, 2014),

generative adversarial networks (GAN) (Goodfellow et al., 2014).

이 있으며, 각각 뚜렷한 장단점을 가집니다.

autoregressive models(PixelCNN)은,

선명한 이미지를 만들수 있지만, 검증이 느리고,

직접적으로 픽셀의 조건부 분포를 모델링하기에 latent representation이 없으며, 그렇기에 적용가능성을 줄인다고

하는데, (do not have a latent representation as they directly model the conditional distribution over pixels, potentially limiting their applicability.) 솔직히 PixelCNN을 아직 읽어보지 않았기에 모르겠네요.

추후 리뷰하고, 이 글도 수정하겠습니다.

variational autoencoders (VAE)는,

학습이 쉽고 성능도 좋은 편이지만 데이터가 흐려지는 것을 막을수 없습니다. (확률분포에 따라 근사치를 구하는 형태라서 언제나 무난한 값을 선택하기에 생겨나는 문제)

GAN은,

서로 다른 역할을 지닌 모델들의 상호작용의 반복으로 매우 선명하고 좋은 출력결과를 만들수 있지만, 특유의 학습 불안정성이 내포하고, 낮은 해상도만 출력이 가능했습니다.

높은 해상도를 만들어내려고 하면, 필요 데이터와 학습 횟수가 어마어마하게 요구되고, 잘 학습되는지도 알수없죠.

- GAN이 현재로써 가장 주목받는 Generative 모델인데, 고해상도로 갈수록 Generator의 거짓을 Discriminator가 알아채기 쉬워지며, 학습시에도 고해상도의 큰 용량으로 인해 작은 규모의 미니배치를 사용해야 했다고 합니다. 즉 학습이 불안정해지죠.

- 일단 Intro는 기존의 문제점이나, 이를 어떻게 해결할 것인지에 대한 설명의 섹션 위치에 대해 알려줍니다.

## [PROGRESSIVE GROWING OF GANS]

- 논문의 방향은 저해상도로부터 시작하는 GAN의 학습 방법론을 제안한다고 합니다.

다음으로 조금더, 조금 더 해상도 높은 이미지를 만들도록 훈련한다고 합니다.

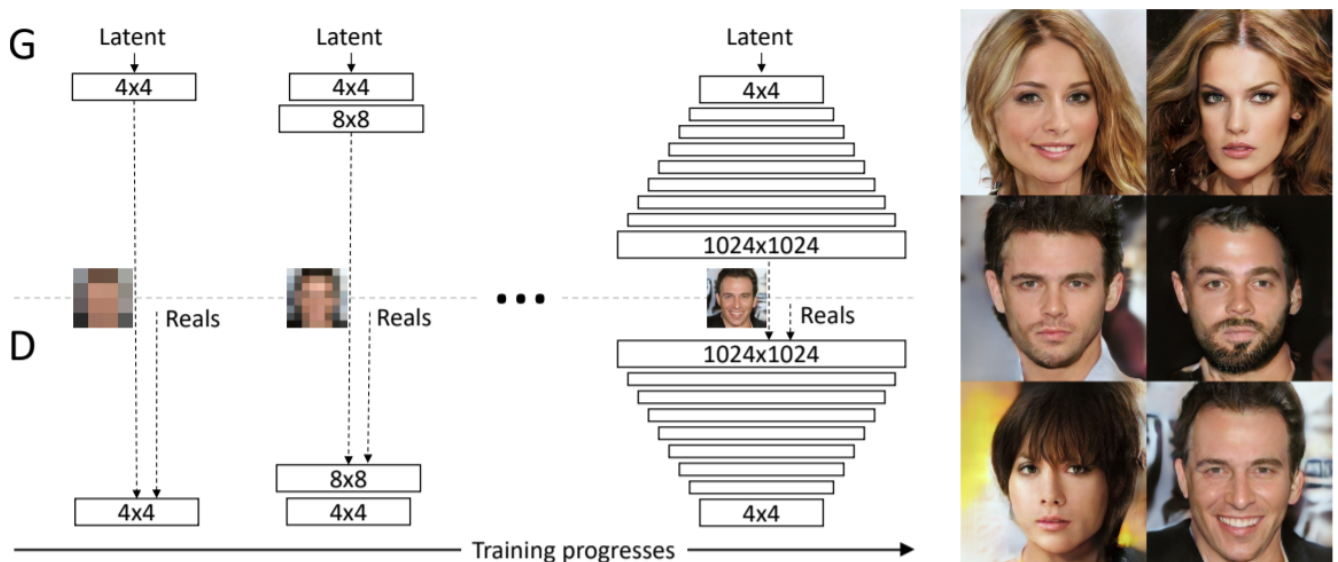
이것의 의미는, 낮은 해상도... 즉 이미지의 전체적인 형태에 대해 먼저 모델이 선행학습을 진행하고, 갈수록 보다 더 디테일한 부분에 집중할수 있도록 하는 것으로,

개인적으로 생각하기에는, 그림 그리는 분들의 구도 잡는 것이 떠오르네요.

그림도 한번에 모든 부분을 디테일하게 그리는 분은... 뭐 없다고는 말 못하겠지만, 아마 그분들도 머릿 속에서는 전체적 조화에서 디테일을 모두 연산한 후에 손으로 옮기는게 아닐까 조심스레 생각해보며, 대다수 사람들은 먼저 전체적인 공간에서의 이미지에서, 점차 더더욱 디테일한 부분을 그려나가는 것과 마찬가지로인 것 같습니다.

여기까지만 들어도, 이를통해 한번에 완벽하게 이미지를 창조해내는 것보다, 선행된 학습에서 만들어낸 것을 기반으로 학습 방향이 정해지기에 무척 좋은 학습법인 것 같네요.

- PGGAN 구조도



보신다면, Generator와 Discriminator가 마치 거울처럼 존재하는 것을 볼수 있습니다.

Auto Encoder가 딱 반대로 생겼죠.

보신다면 training progress별로, 처음에는 4x4의 저해상도 출력으로 대략적인 윤곽을 매우 쉽게 학습합니다.

그리고 그 다음으로, 8x8 레이어를 추가하여, 말 그대로 8x8공간에 대해 채워넣을 값들을 예측해내고, 점진적으로,

4, 8, 16, 32, 64, 128, 256, 512, 1024...

이런식으로 해상도를 올려나가며 학습시키는 것입니다.

매우 간단한데, 매우 획기적이네요.

Gen이 만든 저해상 이미지가 Disc에게 진짜와 같다고 인정받으면, Gen과 Disc의 레이어를 하나씩 더 추가시켜나가는 것으로, 학습 방향을 정할수 있습니다.

정리하자면,

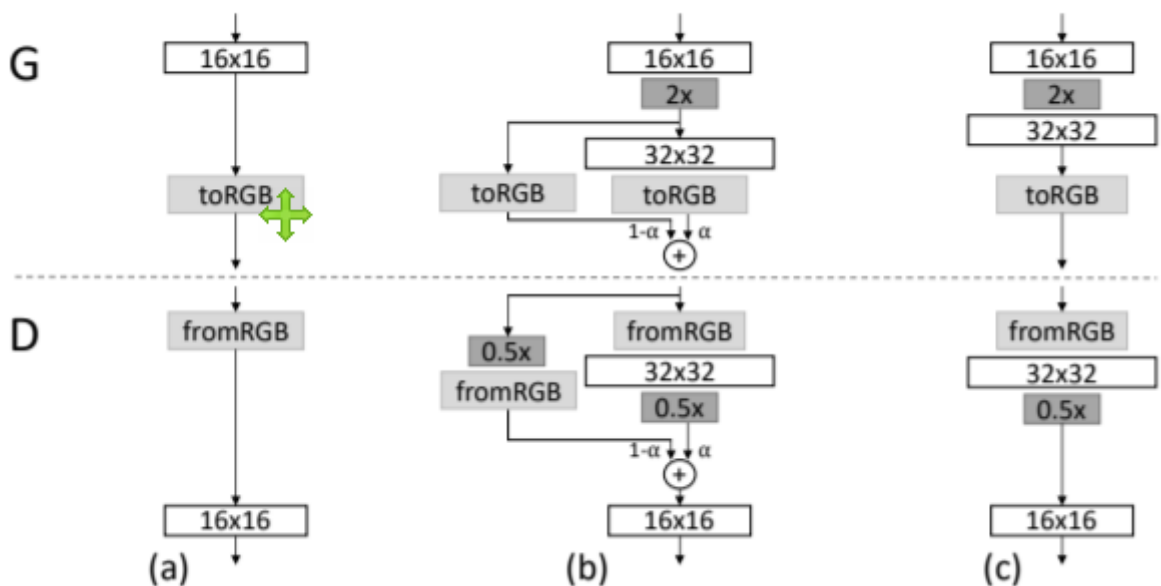
1. PGGAN은 점차적으로 해상도를 높여나가는 방식의 학습법을 사용하는데, 이는 전체적 스케일의 데이터를 학습하고, 추후 세부적인 퀄리티를 학습하는 것과 같다.
2. Gen과 Disc는 서로 거울상이며, 동시에 레이어가 추가되며 증가한다.
3. 모든 레이어는 학습시 동결시키지 않고 학습을 시킨다.
4. 새로운 레이어가 들어올 때, 스무스하게 fade in을 해준다.

라는 것이 해당 논문의 전체 내용으로, 즉 기존 GAN에서 높은 해상도를 효율적으로 학습시키는 방식을 설명합니다.

거의 다 설명한 것이나 마찬가지인데,

아래에서 디테일을 확인해봅시다.

[INCREASING VARIATION USING MINIBATCH STANDARD DEVIATION]



먼저 16x16 해상도로 RGB 이미지를 만들었다고 합시다.

곧바로 그 2배인 32x32 해상도를 학습시킬수 있으면 좋겠지만, 이렇게 될시 아직 학습이 전혀 안된 32x32 레이어의 간섭으로, 기껏 잘 학습된 저해상도 레이어까지 영향을 미칠수 있습니다.

그렇기에 스무스하게 레이어를 끼워넣는 방식이 필요한데,

이전 학습시킨 레이어의 출력 결과물을 사용하는 것입니다.

b로 갈때, 그냥 32x32레이어만 끼워넣는 것이 아니라, 16x16으로 만들어낸 이미지를 2배로 뺏튀기합니다. 그러니까, 저해상도 이미지의 스케일만 늘려서, 대략적인 형태를 지닌 이미지를 만들어내는 것이죠.

그리고 32x32에서 만들어낸 이미지 역시 존재할텐데, 잘 학습된 저해상 이미지와, 이 새로운 이미지를 서로 합칩니다. 그냥 합치는게 아니라, 각 픽셀 값에 1~0 사이의 비율을 나타내는 a를 곱하고, 저해상도 픽셀에는 그 나머지 비율을 뜻하는 1-a를 곱해줘서 서로 더해주면, 저해상도의 큰 그림과, 앞으로 학습시킬 고해상도의 디테일이 합쳐지겠죠?

Disc에서는 역시나 학습 안된 32x32레이어와 더불어, 저해상도로 학습된 16x16을 위해 위와 같은 방식으로 학습을 진행합니다.(a는 0에서 1로 점차적으로 증가합니다. 즉, 이전 레이어의 영향력이 갈수록 줄어드는 것.)

그렇게 하여 기존의 저해상도 레이어에서 학습된 방향을 일그러뜨리지 않고, 새로 추가된 레이어를 학습시킬수 있다고 하네요.

학습이 일정수준 이상이 되었으면 이전 레이어의 보조를 빼고, 그대로 동작을 시키며 파인튜닝을 한다고 합니다.

[그외 적용 기법]

- 아래로는 PGGAN에 적용한 기법들을 써두었네요.

GAN에 발생하는 신호의 증폭을 완화시키기 위해 정규화 레이어를 사용했는데,

Batch Normalization이 아니라 pixel normalization이라는 것을 사용했다고 합니다.

tion” (Krizhevsky et al., 2012), configured as  $b_{x,y} = a_{x,y} / \sqrt{\frac{1}{N} \sum_{j=0}^{N-1} (a_{x,y}^j)^2} + \epsilon$ , where  $\epsilon = 10^{-8}$ ,  $N$  is the number of feature maps, and  $a_{x,y}$  and  $b_{x,y}$  are the original and normalized feature vector in pixel  $(x, y)$ , respectively. We find it surprising that this heavy-handed constraint does not seem to harm the generator in any way, and indeed with most datasets it does not change the results much, but it prevents the escalation of signal magnitudes very effectively when needed.

해당 논문의 수식

- 새로운 레이어의 가중치 초기화는 He Initializer를 사용했다고 합니다.

- 그 밑으로 대부분 실제 사용 결과 및 성능 결과 페이퍼입니다.

[구현]

저해상도 학습에서 시작해서, Gen이 Disc를 속이면, 다음 해상도로 넘어가는 방식으로...

그리고 smooth fade in이 되도록 구현해주면 되겠네요.

- 설계

1. 이미지 데이터셋 준비
2. 이미지 데이터셋 해상도 변경(16,32,64,128 ... 1024)
3. 16x16의 GAN을 작성후 학습.
4. 학습이 완료되면 32x32레이어를 추가.

이때부터 기존 GAN과 다른 구조가 하나 생겨나는데,

바로 이전 레이어의 결과물을 출력하고, 그것에 비율을 매겨서, 이전 레이어 결과물 비율  $1-a$ 와, 이번 레이어 결과물 비율  $a$ 를 통해 점차  $a$ 가 증가하며 학습을 하는 것입니다.

$a$ 가 1이 되어 이번 레이어에게 출력에 대한 최종 영향력이 완전히 넘어가고, 학습이 잘 이뤄지면, 다음 레이어를 또 추가.

5. 원하는 해상도가 될때까지 위를 반복.

입니다.

<https://wiserloner.tistory.com/1160>를 기반으로 해서

위와 같은 설계를 구현할수 있겠네요.

위의 코드는 DCGAN에 대한 기본적인 모델인데, PGGAN에는 WGAN?인가 하는 것을 사용한다고 하니, PGGAN이 제대로 구현된 텐서플로 구조를 한번 찾아봐야 겠습니다.

// 이상입니다.

실제 만드는 것은 추후에 정리.

이번으로, GAN으로 고해상 이미지를 만드는 학습방식을 알아봤는데,

다음으로는 Style GAN으로 GAN의 결과물을 어떻게 조작할수 있을지를 알아볼 것입니다.

## 개발지식 (이론)/인공지능 개인정리 카테고리의 다른글

Batch Normalization 복습

GAN의 종류와 발전

PGGAN 개인정리(논문 리뷰 Progressive Growing of GANs for Improved Quality, Stability, an



DCGAN 개인정리

댓글 0

댓글을 남겨주세요

이전

175 / 1132

다음

[최근 글](#)[최근 댓글](#)

- 1 코틀린 URI / Path String 상호 변환
- 2 안드로이드 시스템 언어 변경 코드
- 3 안드로이드 나이트 모드 전환 코드 kotlin
- 4 안드로이드 kotlin 뷰 터치 영역 확대(T...
- 5 안드로이드 kotlin 키보드 바깥 화면 눌...



---

7 안드로이드 tabLayout 비활성 처리

---

8 지극히 주관적으로 설명하는 프로그래밍...

---

9 git sorce tree 잘못 push 한 commit 내...

---

10 개인정리 : 딥러닝 비지도 학습 기반 이...

---

홈으로

방명록

로그아웃

맨위로

COPYRIGHT LONER의 학습노트, ALL RIGHT RESERVED.