

Do you know GAN? (2/2)

GAN은 생성 모델로써 이미지를 만들어내는 생성기와 생성기로부터 만들어진 이미지와 실제 이미지를 구별하는 구별자가 서로 경쟁을 하는 가운데 서로의 성능이 향상되는 재미있는 구조를 갖고 있다. GAN이 각광을 받는 이유는 성능이 좋기도 하지만 이렇게 독특한 구조 때문이기도 하다. 그동안 있어 왔던 생성 모델들과는 궤를 달리하는 형태이고 이로부터 다양한 아이디어들이 나오면서 최근에는 상당히 높은 질의 이미지들을 만드는 수준에 이르렀다. 지난 글에서 GAN의 기본 원리와 배경 이론에 대해 살펴 보았다면, 이번 글에서는 GAN에 대한 기본적인 이해를 바탕으로 GAN이 기존 모델과 다른 점이 무엇인지 그리고 어떤 장단점이 있는지 조금 더 심화된 내용을 다루고자 한다.

글 | 유재준 jaejun2004@gmail.com

카이스트 바이오및뇌공학과 학부를 졸업하고 동 대학원 박사 과정을 밟고 있는 공돌이입니다. 바이오영상신호처리 연구실에서 전통적인 신호처리 이론을 이용한 영상 복원이나 대수적 위상수학을 이용한 뇌 네트워크 분석을 연구하였으나, 최근에는 기계학습에 빠져서 이를 이용한 의료 영상 복원에 적용해보고 있습니다. 처음 가본 NIPS 2016 학회에서 GAN을 접하고 매우 흥분하여 열심히 가지고 놀아보고 있지만 사실 당장 졸업에 필요한 연구 주제와는 전혀 관계가 없다는 것이 합정입니다. 딥러닝은 이제 갓 1년차 공부 중인 초짜 대학원생이며, 혼자 공부하던 것을 블로그에 정리하던 것이 취미였는데 많은 분들이 좋아해 주셔서 매우 신이 나 있는 상태입니다. 최근 졸업이 다가오면서 외부활동이 점차 줄어들고 있지만, 앞으로 더 많은 분들과 교류하고 배워 이 분야에서 내가 아는 것을 꾸준히 나누는 사람이 되는 게 꿈입니다.

Deep Convolutional GAN

초창기 GAN에 대해 얘기하려고 하면 빼놓을 수 없는 연구가 바로 DCGAN(deep convolutional GAN)* 이다. 지금은 GAN이 매우 뛰어난 결과들을 보여주고 있지만, 초기 결과는 아이디어의 참신성에 비해 그리 인상적이지 않았다. GAN 구조가 학습시키기 매우 어려웠다는 것도 여러 이유 중 하나였다. DCGAN이 나온 이후, GAN 결과들이 매우 급격하게 발전하기 시작했다. DCGAN 논문이 기여한 바를 정리해보면, 다음과 같다.

- 대부분의 상황에서 언제나 안정적으로 학습이 되는 구조를 제안하였다.
- word2vec과 같이 DCGAN으로 학습된 생성기는 벡터 산술 연산이 가능한 성질을 갖고 의미론적(semantic)으로 신규 샘플을 생성할 수 있다는 것을 보여 주었다.

DCGAN 논문은 GAN이 잘 학습되는 구조를 매우 세세한 가이드 라인으로 제시한 연구였다. 이 논문 이후에 나온 대부분의 GAN 연구들이 어떤 형태로든 DCGAN 구조를 바탕으로 하고 있다고 할 정도로 DCGAN은 매우 잘 확립된 구조이다. 일단 DCGAN 에서 제시한 가이드 라인대로 GAN 구조를 짜면 상당히 안정적으로 학습이 된다. 이런 구조를 발견하기 위해서 대학원생들이 얼마나 힘들었는지 논문의 한 구절에서 언뜻 느껴볼 수 있다.

"그러나 광범위한 모델 탐구 끝에 우리는 다양한 데이터셋을 대상으로 안정적인 학습을 할 수 있고, 더 높은 해상도와 더 깊은 생성 모델(deeper generative models)을 가능하게 하는 구조의 모음(a family of architectures)을 발견했다."

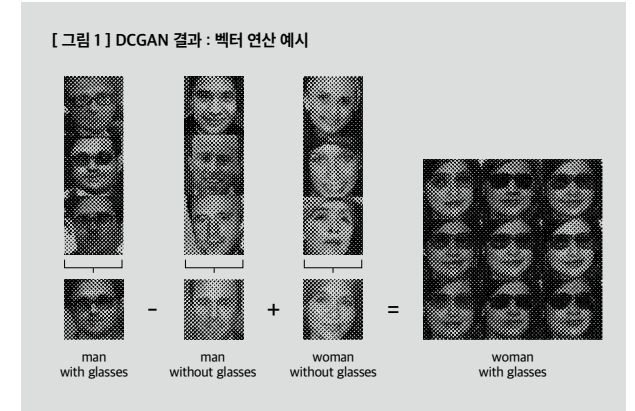
DCGAN은 이름에서 알 수 있듯이 convolution 구조를 GAN에 잘 결합한 것이다. CNN(convolutional neural network)이 지도 학습(supervised learning)에서 매우 큰 성공을 거둔 것에 비해 비지도 학습(unsupervised learning)에서는 상대적으로 잘 사용되지 못했다. 이러한 측면에서 DCGAN 논문은 지도 학습과 비지도 학습에서의 격차를 줄이는 데 큰 역할을 했다고 평가된다.

DCGAN 결과에서 가장 재미있는 부분은 아래와 같이 생성기의 입력으로 들어가는 z 은닉 공간(latent space)에서 벡터 산술 연산이 가능하다는 점이다. 가장 흔한 예시로 word2vec 연구가 있다. 다음 수식을 계산하고 답을 추정해보자.

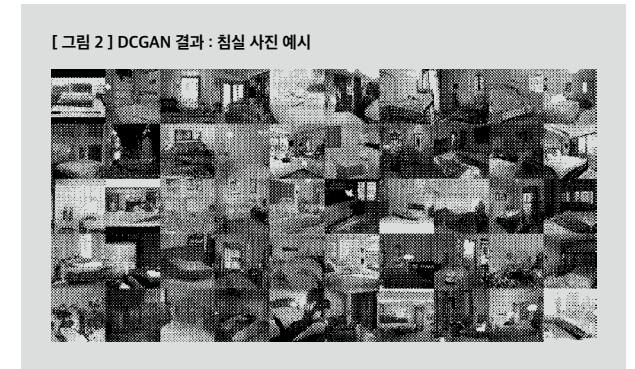
KING - MAN + WOMAN

사람은 생각보다 쉽게 위 구조 속에서 QUEEN이란 단어를 연상할 수 있지만 컴퓨터에게 이런 연산은 단어의 의미를 이해하고, 그에 맞는 새로운 단어를 찾는 등의 매우 고차원의 처리가 필요한 어려운 문제다. 기존의 word2vec 연구에서는 뉴럴 네트워크를 사용하여 말뭉치에서 실제로 단어 간의 관계를 학습하는 것을 보여 주었고, DCGAN은 이를 말뭉치가 아닌 이미지에서 하는 것이 가능하다는

것을 시사했다. 아래 [그림1]이 바로 그 결과이다.

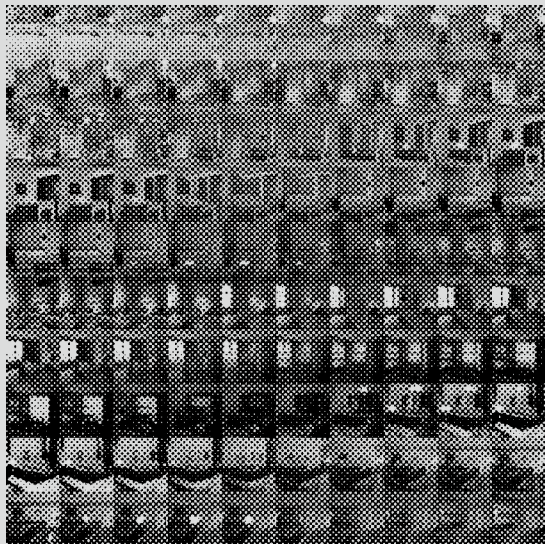


실제로 모두 DCGAN으로 생성된 결과들이다. 상당히 진짜같은 결과만으로도 놀라운 데, 이미지가 갖는 의미를 바탕으로 직관적인 벡터 산술이 가능하다는 것을 알 수 있다. 안경을 쓴 남자와 안경을 쓰지 않은 남자 그리고 안경을 쓰지 않은 여자를 생성하게 하는 입력 값들이 은닉 공간에 각각 벡터로 존재하고 있을 텐데, 각각의 벡터를 서로 빼고 더해 주면 최종적으로는 안경을 쓴 여자를 생성하는 입력 벡터를 찾을 수 있다는 것이다. 물론 z벡터 하나만 생성기에 입력해서서 깔끔한 결과가 나오지 않기에 세 개 정도를 추리고 그 세 개의 평균 z벡터를 사용해서 결과를 만든 것이기는 하다. 하지만 신기한 것은 매한가지다. 어떻게 보면 네트워크가 영상의 의미를 이해했다고 생각할 수 있다. 이 외에도 [그림2]와 같이 침실을 생성한 결과 그림들을 보면 작은 그림이긴 하지만 꽤나 그럴듯한 결과를 만들어 냈다는 것을 확인할 수 있다.



논문에서 "공간을 걷는다" 라고 표현했듯이 [그림3]처럼 은닉 공간에서 천천히 벡터의 값을 바꾸면, 생성기가 내보내는 이미지가 하나의 침실에서 다른 침실로 위화감 없이 부드럽게 변화하는 것을 볼 수 있다. 특히 벽이었던 부분이 자연스럽게 하나의 창으로 변하는 것을 보면 매우 놀랍다.

[그림 3] "은닉 공간에서 돌아다니기"



만약 생성기가 단순히 영상을 외워서 보여줄 뿐이라면 주어진 특정 입력에 대해 특정 이미지를 내보내는 일대일 대응 함수를 학습한 것으로 생각할 수 있다. 이럴 경우 은닉 공간에서 굳이 부드러운 변화가 있을 이유가 없다. 바로 옆의 z 벡터가 전혀 다른 샘플과 일대일로 연동될 수 있기 때문이다. 이렇듯 은닉 공간에서 벡터 연산이 가능하다는 것과 입력에 변화를 줬을 때 생성되는 결과가 부드럽게 변하는 것을 보는 분석이 중요한 이유는 우리가 학습시킨 GAN의 생성기가 일대일 대응 함수와 같이 매우 단순하고 의미 없는 함수(mapping)를 학습한 것이 아닌란 것을 시사하기 때문이다.

이렇게 수많은 이미지를 표현할 수 있는 정보들을 포괄하면서도 부드러운 변화에 대응할 수 있는 함수를 학습하기 위해서는 은닉 공간을 잘 정하는 것도 매우 중요한 일이다. GAN에서는 보통 z 은닉 공간은 고차원의 가우시안 분포를 많이 사용한다. 적절한 가정 하에서 충분히 복잡한 함수를 사용하여 대응시킬 수만 있다면 임의의 d 차원 분포는 정규 분포를 따르는 d개의 변수로 나타낼 수 있다는 것이 알려져 있기 때문이다².

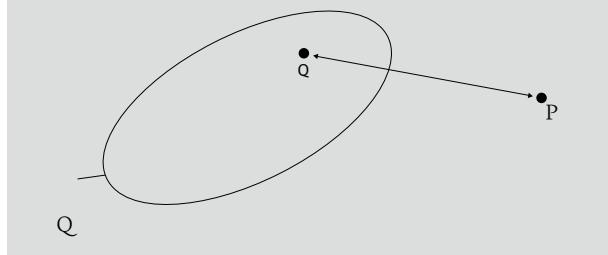
기존 생성 모델과 GAN의 차이점

그렇다면 GAN은 기존의 생성 모델들과 어떤 면이 다르기에 이렇게 비교적 뚜렷한 이미지를 만들 수 있는 것일까? GAN의 특징이자 기존 생성 모델들과 구별되는 가장 큰 차이점은 바로 GAN이 사실 샘플러(sampler)라는 것이다. 즉, 직접적으로 데이터의 분포를 학습하는 형태가 아니라 하나의 입력이 들어갔을 때 하나의 출력을 주는 샘플러를 학습한다는 독특한 특징을 지닌다.

조금 더 자세히 이해하기 위해 "확률 모델을 학습한다" 는 것에 대해 생각해보면, 모델을 추정한다는 것은 일반적으로 우리가 알고 싶은 P라는 진짜 데이터의 분포가 어딘가에 있을 때,

- (1) 이로부터 얻은 샘플들이 독립항등분포(independent and identically distributed, I.I.D.)의 성격을 띄고 있는 것으로 관측되었다고 가정한 후,
 - (2) Q라는 모수 모델 클래스(parametric model class)를 만들어,
 - (3) 그 안에서 P와 가장 가까운' 모델의 모수를 찾아내는 것을 말한다.
- 가장 간단한 예시로 데이터의 분포가 정규 분포를 따를 것이라 가정하여 가우시안 모델을 세우고, 현재 내가 갖고 있는 데이터를 가장 잘 설명할 수 있는 평균과 분산 값을 찾는 과정을 생각해볼 수 있다. 이를 위해서는 P와 Q의 '차이' 혹은 '거리'를 계산할 수 있어야 한다. 그러면 구한 두 분포 사이의 거리를 줄여나가는 방향으로 모델의 모수들을 조정할 수 있다. 이 과정을 적합(fitting)이라고 한다. GAN도 Jensen-Shannon divergence라는 측도를 사용하여 분포 간의 거리를 계산하고 이를 최소화하는 방향으로 생성기를 학습한다고 분석할 수 있다³.

[그림 4] 확률 모델 학습



보통 기존의 방식에서는 아래와 같은 Q에 대한 가정들을 사용한다.

- 쉬운 샘플링이 가능해야 한다.
- 샘플에 대한 모수의 경사(gradient)가 계산 가능해야 한다.
- 우도 함수가 다루기 쉬운 형태여야 한다.

이 중 가장 강력한 가정이 바로 우도 함수(likelihood function)가 계산 가능하다(tractable)는 것이다. 많은 경우 현실의 모델은 계산이 불가능한 형태의 수식으로 나타나는 것을 상기한다면, 기존의 모델들이 얼마나 강력한 가정을 사용하고 있는지 알 수 있다. GAN 형태의 모델들은 임의의 확률 변수 입력 값을 사용하여 비선형 변환 함수를 통과시키면 출력으로 샘플이 하나 튀어 나오는 구조다. 마치 버튼을 누르면 샘플이 튀어 나오는 자판기처럼 생각할 수 있다. GAN 모델들의 독특한 점은 바로 이 부분에서 나온다. GAN 모델들은 다른 확률 모델들과는 달리 우도 함수를 근사하려 하지 않고 샘플을 뽑아주는 함수를 학습했을 뿐이기 때문에 우도 함수 모델링이 필요 없는(likelihood-free) 모델이라고 할 수 있다.

물론 이 부분은 GAN의 특징일 뿐, 장점일 수도 있고 단점일 수도 있다. 데이터 분포에 대한 모델을 특정하지 않고 하나의 샘플을 뽑아서 보여주기 때문에 고정된 모델의 한계에 제약 받지 않고 또렷한 이미지를 보여줄 수 있기도 하지만, 다른 한편으로는 정작 이미지를 잘 뽑더라도 데이터의 분포에 대한 정보를 직접적으로 얻기는 어렵다. 때문에 분포를 알았을 때 시도해볼 수 있는 많은 분석들을 시도할 수 없다. 이 부분에 대해서는 이안 굿펠로우(Ian Goodfellow)의 NIPS 2016 tutorial 논문⁴이나 같은 시기, 같은 워크숍에서 발표된 세바스찬 노워진(Sebastian Nowozin)의 f-GAN⁵ 논문을 참고한다면 조금 더 심화된 내용을 확인할 수 있다.

이외에도 이후에 연구된 WGAN⁶에서는 기존의 GAN이 divergence를 측도로 사용 하기 때문에 생기는 여러 가지 문제를 지적하며 다른 방식의 측도를 제안하는 등, 점차 수식적인 분석과 함께 GAN의 가치 함수 자체를 근본적으로 수정하는 방향으로 연구가 발전되었다. 이를 바탕으로 보다 안정적인 학습과 결과를 보여 주었는데, EBGAN, LSGAN, BEGAN 등 이후 나온 많은 GAN들이 WGAN의 갈래로 분류 될 수 있다.

이렇게 보면 모든 연구가 끝나서 더 이상 할 것이 없는 것처럼 보이고, 점차 이론적인 문제로 깊게 들어가면서 공학자들이 개입할 수 있는 여지가 없는 것 같지만 아직은 풀어야 할 문제가 많다.

학습이 예전에 비해 수월해졌다는 것이지 정말 쉬워졌다는 것을 의미하진 않는다. 또한 네트워크의 학습이 안정적으로 이미지의 질을 보장하지 않는 경우 또한 많다. 수렴은 여전히 어렵고 이어 소개할 mode collapse나 모델 평가 등 아직도 풀어야 할 문제가 산적해 있다. 그런 의미에서 GAN 학습이 어려운 이유를 하나씩 알아 보자.

GAN 학습이 어려운 이유 I: Convergence

지난 글에서 소개했듯이 원 논문에서 GAN에 대한 이론적 근거를 증명해 주었지만 아쉽게도 실제 구현은 이론적 배경이 되는 가정과 거리가 있다. 때문에 GAN 가치 함수를 뉴럴 네트워크를 이용해 풀었을 때 이상적인 전역해로 수렴한다는 보장이 되지 않는다. 게다가 풀어야 하는 문제의 형태부터 이미 쉽게 문제를 풀 수 있는 볼록 함수 형태가 아닌 변수 두 개가 서로 엮여 있는 안장점 문제(saddle point problem)를 고려해야 하기 때문에 GAN은 학습이 매우 어렵기로 유명하다.

실제로도 많은 사람들이 간단한 예제에서도 문제를 풀기가 어려울 수 있는데 더 복잡한 문제에서 GAN 형태가 잘 풀릴 것인지에 대해 의문을 제기한 바 있다. 이 문제를 약간 더 직접적으로 느끼기 위해서 실제로 간단한 예제인 $V(x, y) = xy$ 가

어떻게 생겼는지 그려보면 [그림 5]와 같다.

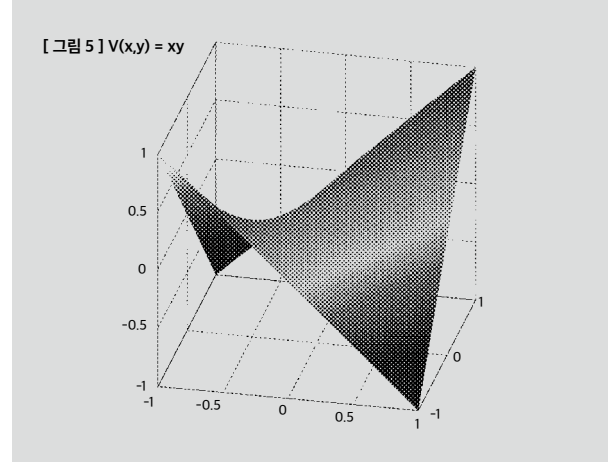
이 문제는 $x = 0, y = 0$ 에서 안장점을 갖는 매우 대표적인 예시다. 그리고 x와 y에 대해 최소최대 문제를 풀면 이 안장점이 평형점이라는 것도 쉽게 알 수 있다. 사실 안장점이 모두 평형점이 되는 것은 아니지만, 이 경우 하나의 변수에 대한 작은 변화가 다른 변수에 대해 가치 값을 줄일 수 없기 때문에 평형점이 되는 것이다. 만약 이 문제를 경사하강법(gradient descent)으로 풀면 결과가 평형점 주변에서 수렴하지 못하고 최초 시작점에 따라서 반지름이 정해지는 궤도(orbit) 위에서 영원히 헤매는 것을 확인할 수 있다. 심지어 학습률(learning rate)이 크면, 바깥 방향으로 발산하는 경우도 생길 수 있다.

이를 수식과 함께 확인해보면 조금 더 명확해진다. 학습율 γ 를 고정하고 $n \in \mathbb{N}$ 일 때, 각 변수에 대해 경사하강법으로 번갈아 계산하는 것은 [수식 1]과 같다.

$$\begin{aligned} \text{[수식 1]} \quad x_{n+1} &= x_n - \gamma y_n \\ y_{n+1} &= y_n + \gamma x_n \end{aligned}$$

[수식 1]은 [수식 2]처럼 다시 표현할 수 있고, [그림 5]처럼 $V(x,y) = xy$ 이므로

$$\text{[수식 2]} \quad \begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & -\gamma \\ \gamma & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix}$$



여기서 다시 [수식 3] 으로 바꾸고

$$\text{[수식 3]} \quad \begin{bmatrix} 1 & -\gamma \\ \gamma & 1 \end{bmatrix} = \frac{1}{\alpha} \begin{bmatrix} \alpha & -\alpha\gamma \\ \alpha\gamma & \alpha \end{bmatrix} = \frac{1}{\alpha} \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

α 를 [수식 4]와 같다고 해보겠다.

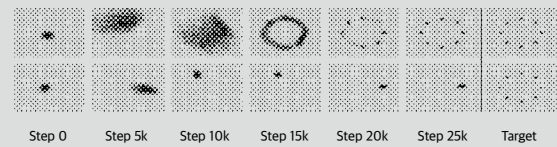
[수식 4]

$$\alpha = \sqrt{\frac{1}{1+\gamma^2}} \in (0, 1), \theta = \cos^{-1} \alpha \in \left(0, \frac{\pi}{2}\right)$$

고등학교에서 회전 행렬에 대해 배운 사람이라면 위의 행렬식이 매우 익숙할 것이다. $\gamma \approx 0$ 일 때, 즉 학습율이 충분히 작아서 $\alpha \approx 1$ 이면 경사하강법의 결과가 언제나 안정한 궤도(stable orbit)로 빠지고, $\alpha < 1$ 인 경우 (x_n, y_n) 이 나선형으로 발산하게 된다.

GAN 학습이 어려운 이유 II: Mode Collapse

앞서 소개한 문제도 그렇지만 GAN 학습의 어려움은 대부분 그 가치 함수의 형태에서 기인한다. 두 번째로 소개할 mode collapse 문제 역시도 GAN의 독특한 가치 함수와 그 문제 풀이 방식 때문에 발생하는 것으로 해석할 수 있다.

[그림 6] Mode Collapse 예시⁷

[그림 6]은 전형적인 mode collapse 문제의 예시다. 맨 오른쪽의 목표 분포를 보면 가우시안 혼합 모델로 총 여덟 개의 최빈값(mode)이 있는 것을 볼 수 있다. 아래 줄의 그림이 이 목표 분포를 근사하기 위해 GAN으로 여러번 반복하여 학습한 결과다.

GAN이 뽑은 샘플들을 보면 각각의 최빈값들을 각 단계마다 돌아가며 방문하는 것을 볼 수 있다. 즉, 원래라면 윗줄과 같이 전체 최빈값들을 보고 목표 분포가 여덟 개의 최빈값을 갖는 분포라는 것을 찾아내야 하지만, GAN은 그렇게 하지 못 하는 모습을 보여준다. 조금 더 직접적인 예를 들자면 숫자가 1부터 8까지 섞여 있는 이미지들로 데이터 분포가 있을 때, 우리는 GAN이 1부터 8까지 모든 숫자들을 만들어낼 수 있기를 바라는데 실제로는 1과 같이 가장 쉬운 한 가지 숫자만 만드는 모델을 학습한다는 것이다. 이런 현상을 하나의 최빈값에만 함몰되는 모델이나 함수를 학습한다고 하여 mode collapse 문제라고 부른다.

사실 우리가 매 단계마다 최적의 구별자 D^* 를 계산할 수 있다면 이런 문제가 생기지 않겠지만, 뉴럴 네트워크로 모델을 만들 경우 수렴이 될 때까지 계산을 매우 여러번 해야 하고 여러 가정이 깨지면서 수렴이 보장되지도 않는다. 따라서 현실적으로는 가치 함수를 각각의 변수에 대해 일정 횟수만큼 번갈아 푸는 방식을 택하는데 이런 방식 때문에 문제가 생기게 된다. 원래 풀고자 하는

GAN의 가치 문제는 [수식 5]와 같은 최소최대 문제이다.

[수식 5]

$$G^* = \min_G \max_D V(G, D).$$

그렇지만 실제 학습을 할 때는 G와 D에 대해 번갈아가며 풀어주기 때문에 뉴럴 넷의 입장에서는 이러한 최소최대 문제와 [수식 6]과 같은 최대최소 문제가 구별되지 않는다.

[수식 6]

$$G^* = \max_D \min_G V(G, D).$$

문제는 최대최소 문제에서 생긴다. [수식 6]의 안쪽부터 살펴보면 G에 대한 최소 문제가 먼저 있기 때문에 생성자의 입장에서는 현재 고정되어 있는(비최적, non-optimal) 구별자가 가장 헛갈려 할 수 있는 샘플 하나만 학습하면 된다. 즉, 가치 함수를 가장 최소화할 수 있는 최빈값 하나만 내보내면 된다. 이렇듯 GAN의 가치 함수 자체와 엮여 있는 문제이기 때문에 mode collapse 문제는 아직도 GAN에서 완전하게 해결되지는 않고 있다.

GAN 학습이 어려운 이유 III: Evaluation

이에 더해 모든 생성 모델이 갖는 고질적인 문제가 바로 평가의 객관성이다. 생성한 이미지의 질을 평가할 수 있는 객관적인 기준을 정하는 것이 매우 어렵기 때문에 새로운 모델이 예전의 모델에 비해 발전한 것인지 평가하는 것이 쉽지 않고 연구의 방향을 잡기도 어렵다.

현재 사용되는 방식을 몇 가지 살펴보자면 대표적인 것이 아마존 메카니컬 터크(Amazon Mechanical Turk)를 사용하여 사람이 직접 평가하도록 하는 방식이다. 그러나 이런 방법은 매우 주관적이고 일관된 평가가 어려우며, 위에서 설명한 mode collapse가 일어난 경우 전혀 모델의 문제점을 파악할 수 없다는 단점이 있다. Mode collapse가 일어난 모델의 경우 생성하는 이미지의 다양성이 부족할 뿐이지 단일 이미지 자체의 품질은 상당히 좋을 수 있기 때문이다.

두 번째로는 inception score라고 하여 구글의 인셉션 이미지 분류 모델에 생성된 이미지를 넣어 나오는 값을 바탕으로 평가를 하는 방식이 있다. 이 방법은 값이 일관되고 어느 정도 객관성을 보장할 수 있다는 장점이 있어 꽤 자주 사용되고 있다. 하지만 굳이 인셉션 모델을 사용해야하는 이유도 없고 어떤 모델의 경우 인셉션 모델에 특화(overfitting)되어 실제 이미지의 질과는 무관하게 점수가 좋게 나올 수 도 있다는 문제를 안고 있다. 이렇게 앞서 소개한 문제들 외에도 다양한 연구 거리가 남아 있겠지만 세 가지로

크게 정리해 보았다. GAN 연구가 활발하고 매일 하루가 멀다 하고 쏟아지는 만큼 더 이상 연구할 것이 없고 너무 늦었다고 생각할 수도 있으나 알고 보면 아직 가야할 길이 멀다.

'항GAN기'-믿거나 말거나

마지막으로 GAN에 대한 탄생 비화를 소개하면서 글을 마무리하겠다. 이안 굿펠로우가 최초로 만든 GAN은 multi-layer perceptron(MLP)을 이용한 매우 단순한 형태의 GAN이었다고 한다. 거짓말같이만 단 한 번의 시도만에 바로 성공했다는데... 물론 매우 간단한 문제에 대해 적용해봤을 것으로 추측되지만 GAN이 수렴시키기 어렵기로 악명 높다는 것을 생각해보면 솔직히 믿기지 않는 일하다.(마치 박혁거세 설화를 보는 느낌이랄까...)

이안 굿펠로우가 GAN에 대한 아이디어를 처음 떠올린 순간은 몬트리올의 'The 3 Brewers'라는 펍에서 친구들과 얘기를 하던 중이었다고 한다. 박사를 마치고 연구실을 떠나는 친구를 송별하는 자리였는데, 그렇게 모인 친구들 중 한 명이 모든 사진의 통계적 정보를 넣어서 스스로 사진을 만들어 낼 수 있는 기계에 대해 얘기를 꺼냈다. 즉석에서 친구들끼리 어떻게 하면 그런 기계를 현실적으로 만들 수 있을 지에 대해 논쟁이 벌어졌다.

존재하는 모든 사진에 대한 통계적인 정보를 얻는 것부터 일단 말이 되지 않으니 불가능한 프로젝트라고 생각하다가 순간 뉴럴 네트워크를 사용해 기계를 가르치면 더 진짜 같은 사진을 만들 수 있지 않을까 하는 생각이 들었다고 한다. 하지만 친구들은 그 아이디어에 대해 부정적이었고, 살짝 열이 받은 이안은 새벽에 술자리에서 돌아오자마자 앉은 자리에서 노트북으로 GAN을 코딩했다고 한다. 그리고 거짓말같이 단 만에 성공했다는데, 이후 인터뷰⁸에서도 "매우 매우 운이 좋았다. 만약 GAN이 한 번에 성공하지 않았다면, 그냥 포기했을 것이다" 라며 스스로도 운이 좋았다고 말한 바 있다.

이 인터뷰 내용이 사실이라면 여기서 우리는 여러가지 교훈을 얻을 수 있다(옛날 이야기에는 언제나 교훈이 있는 법). 문제를 설정하고 풀 때 직관력이 매우 중요하다는 것과 그 직관으로 얻은 아이디어를 바로 실험해보는 실행력이 중요하다는 점, 술자리에서 연구 얘기만 주구장창 하여도 진지하고 재미있게 들어줄 사람들이 있는 집단에 들어가야 한다는 것, 그리고 마지막으로 되는 놈은 뭘 해도 된다고 운도 조금은 좋아야 한다는 것이다. 희망적인 것은 어떤 문제에 대한 직관력은 그 분야와 연관된 깊은 수학적 지식에서 나올 수도 있지만 수많은 시행착오(a. k. a. 삽질)를 바탕으로 한 경험에서 길러진다는 점이다. 그리고 매우 다양하게 많은 시도를 하다보면 올바른 방법을 찾을 확률이

높으니 운이라는 것도 어느 정도 통계에 기대볼 수 있을 것 같다.

이렇게 열정적으로 문제를 풀다보면 비슷한 사람들이 모여 있는 집단(카카오..?)에 갈 수 있는 기회가 생기고, 더 재미있게 잘 연구할 수 있는 선순환이 이루어지지 않을까?

엄밀한 수학 지식을 바탕으로 차근차근 쌓아 올렸을 것 같은 매우 복잡한 이론들도 순간적인 직관에서 시작하는 경우가 매우 많은 것 같다. 그러니 수학이 어려운 공학자들이여 우리 모두 힘을 내자! 수학을 잘하는 수학자들과, 실험을 바탕으로 감이 살아있는 공학자들이 각자의 영역에서 연구를 하며 협업을 통해 문제를 발전시켜 나간다면 언젠가는 정말로 이미지뿐만 아니라 사람의 생각을 모사하는 공극의 생성 모델 또한 만들어 낼 수 있지 않을까 기대해본다.(필자의 꿈이기도 하다!)

^{*1} 논문 | Radford, A. (2016). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. doi : arXiv:1511.06434. ^{*2} 논문 | Doersch, C. (2016). Tutorial on Variational Autoencoders. doi : arXiv:1606.05908. ^{*3} 논문 | Goodfellow, I. et al. (2014). Generative Adversarial Nets. doi : arXiv:1406.2661. ^{*4} 논문 | Goodfellow, I. (2016). NIPS 2016 Tutorial: Generative Adversarial Networks. doi : arXiv:1701.00160. ^{*5} 논문 | Nowozin, S. et al. (2016). f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization. doi : arXiv:1606.00709. ^{*6} 논문 | Arjovsky, M. & Chintal, S. (2017). Wasserstein GAN. doi : arXiv:1701.07875. ^{*7} 논문 | Metz, L. et al. (2016). Unrolled Generative Adversarial Networks. doi : arXiv:1611.02163. ^{*8} 참고 | https://www.wired.com/2017/04/googles-dueling-neural-networks-spar-get-smarter-no-humans-required/