

[10차시 수업]

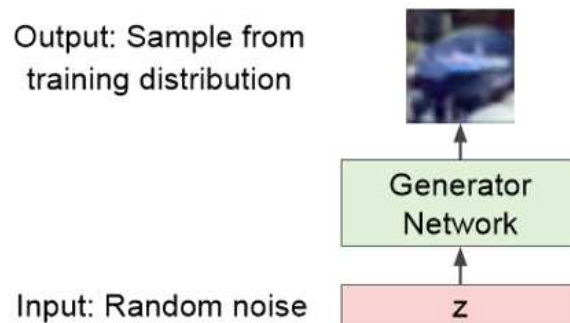
1. cGAN(Conditional GAN)

1) GAN은 Generator와 Discriminator의 상호 견제를 통하여, 기존 Generator model처럼 복잡한 확률 계산 없이도, 기존 어떤 Generator 보다 훨씬 실제와 비슷한 이미지를 만들어낼 수 있다.

GAN을 사용하면, 학습에 사용한 이미지와 흡사한 가짜 이미지를 만들 수 있지만, original GAN 모델에서는 어떤 이미지를 만들어낼지는 제어할 수가 없었다.

■ Latent Variable

Generator는 아래 그림과 같이 random noise(latent variable)이라고 불리는 ‘z’를 입력으로 받아 학습 영상의 분포를 반영한 실제와 비슷한 가짜 영상을 출력으로 내보내게 된다.

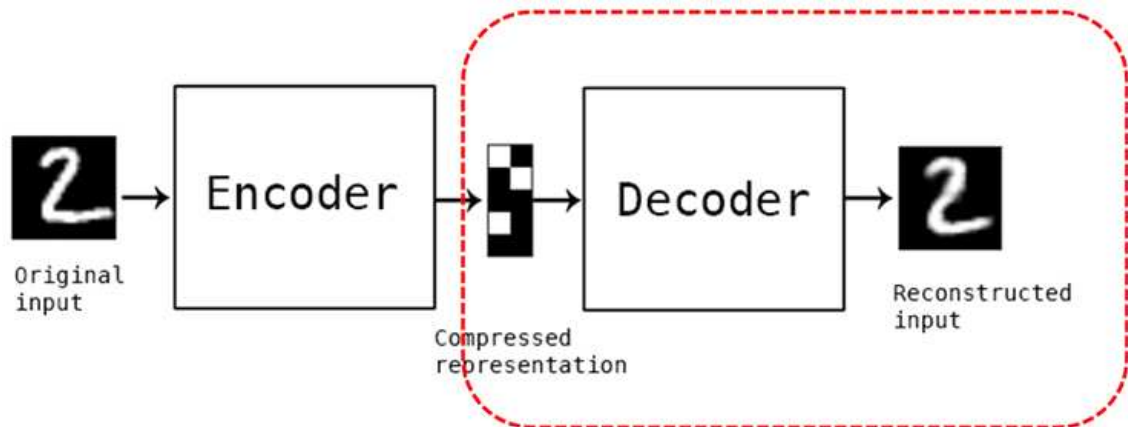


- Latent variable(내재 변수 혹은 잠재 변수)은 직접적으로 관측되는 변수가 아니라, 관측이 가능한 다른 변수들로부터 추론이 가능한 변수

구체적인 예를 들면 “건강”이라는 추상적인 변수는 직접적으로 측정할 수 없는 “내재 변수”이지만 몸이나 정신 상태를 표현하는 말로 자주 사용이 된다. 직접 계측할 방법은 없지만, 혈압, 맥박, 혈당, 체온, 체중, 허리둘레 등 “(많은) 관측이 가능한 외부 변수”로부터 추론이 가능하며, 이런 외부 변수로부터 추상적인 건강이라는 내재 변수를 도출하려면 수학적인 혹은 경험적인 모델이 필요하다.

그럼 건강이라는 내재 변수는 왜 사용할까? 좀 전에 살펴본 각각의 변수를 떼어서 해석하면 몸 상태를 파악하기가 어렵지만, 이것들을 건강이라는 내재 변수(추상적 개념)로 엮어내면 해석이 쉬워지기 때문이다.

- Autoencoder에서는 입력 데이터를 받은 후 encoding 과정을 거쳐 압축된 결과(compressed representation)를 만들어 내는데, 이것이 바로 내재 변수가 된다. 이미지는 색의 삼원소를 갖는 픽셀들의 배열로 구성이 되지만, 이것들을 encoder를 통해서 압축시키면 결과적으로 “z”가 얻어지며, 이렇게 얻어진 “z”를 다시 decoder 망을 통과시키면 복원된 데이터가 나오게 된다. 여기서 뒷부분만을 보면, Generator의 과정과 매우 유사하다는 것을 확인할 수 있다.



GAN에서는 “z”를 noise로 표현하고, latent variable이라고도 한다. noise라고 쓰는 이유는 사전에 정의할 수 없음에 의미를 둔 표현이고, latent variable이라고 하는 것은 autoencoder처럼 이것으로부터 출력 영상을 만들 수 있기 때문에 이 표현도 쓴다. 이 latent variable에 뭔가 이미지 생성에 관련된 condition을 부가하고, 이것을 통해서 이미지 생성을 가이드 할 수 있다면, 이제는 원하는 방향으로 이미지를 생성할 수 있게 되며, 다양한 분야에 응용이 가능해진다. 몬트리올대 팀은 이 점에 착안하여 cGAN이라는 망을 고안하였고, 이후 많은 사람들이 더 다양한 시도를 해볼 수 있는 단초를 마련하였다.

■ Original GAN

GAN은 2 player min-max game을 통하여 학습한다는 것.

즉, 학습 과정에서 상호 협조적이지 않는 상대(Generator와 Discriminator)가 서로 최적의 상태(Nash Equilibrium)에 도달하려고 노력하는 방식이며, 이 목표를 달성하기 위한 objective function으로 아래와 같은 식을 사용한다.

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

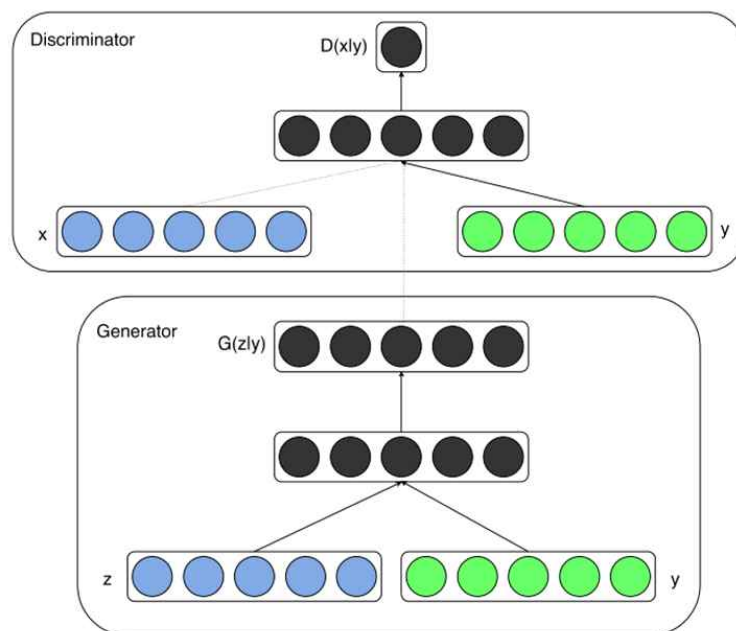
↑ Expectation
 ↑ x is sampled from real data
 ↑ Probability of D(real)
 ↑ z is sampled from N(0, 1)
 ↑ Probability of D(fake)
 ↑ fake

■ Conditional GAN(cGAN)

Original GAN에 어떤 condition을 가하는 방법으로 Generator와 Discriminator에 특정 condition을 나타내는 정보 y 를 가해주면 된다.

여기서 y 의 형태는 특별하게 정해진 것이 아니기 때문에 다양한 형태를 가질 수가 있다. 예를 들어, 숫자 필기체를 인식하는 MNIST에서 원하는 숫자를 generation 하고 싶다면, 숫자의 class에 해당하는 label(논문에서는 one-hot encoding을 하였음)을 추가로 넣어주면 된다.

만약 MNIST를 one-hot encoding을 하면 10-bit가 필요하기 때문에 y 는 10비트가 된다. 그렇지만 꼭 이렇게 class의 label 뿐만 아니라, 다양한 형태(multi-modal)도 가능하다. 아래 그림처럼 Generator의 기존 latent variable에 y 를 더해주면 되고, Discriminator 역시 마찬가지이다.



- cGAN은 Original GAN에 부가 조건 y 가 들어가는 점을 제외하면 동일함.
- cGAN을 풀어주기 위한 objective function을 구한다면, y 가 조건부로 들어간 것을 제외하면 동일하기 때문에, objective function 역시 아래처럼 거의 동일하다.

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

■ cGAN 결과

논문에서는 MNIST data에 대하여 조건 y 를 one-hot encoding 시킨 class label을 사용했으며, 이 실험을 통하여 아래 그림처럼, 원하는 숫자가 condition y 대로 생성되는 것을 확인할 수 있었다.



- Conditional GAN은 생성 조건을 제어해줄 수 있는 condition y 를 추가하는 간단한 동작만으로, 원하는 데이터를 생성할 수 있다.