

# GAN Based Anomaly Detection From Scratch : 밑바닥부터 시작하는 GAN 기반 이상치 탐지

## PT

- 1. Machine Learning
- 2. Generative Model
- 3. Anomaly Detection

2

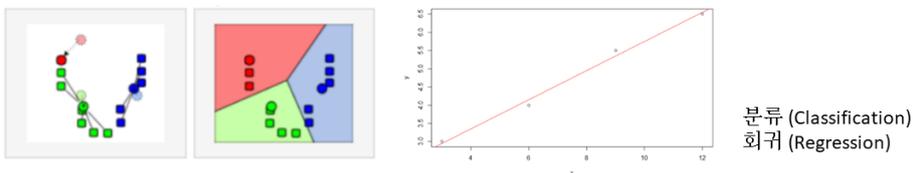
GAN Based Anomaly Detection에 대해 발표하려고 한다.

GAN에 대한 내용이므로 다들 머신러닝을 알 거라고 생각하지만 그래도 개념을 안 배운 분들도 있을 수 있다고 생각해서 간단하게나마 머신러닝을 리뷰하고 넘어가려고 한다.

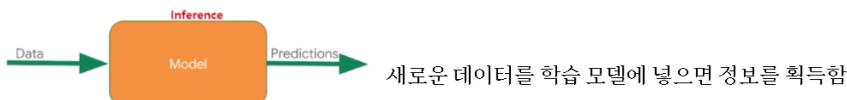
이후 Generative Model이 무엇인지 설명하고 우리가 하려는 Task인 Anomaly Detection에 대해 소개를 할 것이다.

### Machine Learning

- Machine Learning
  - 컴퓨터가 데이터를 가지고 학습하고, 학습한 것을 바탕으로 새로운 데이터가 왔을 때 예측하는 기술
  - 학습: 데이터의 패턴 찾기
  - 데이터의 패턴: 데이터가 어떻게 분포되었는지, 데이터가 어떤 형태로 위치하는지 등



분류 (Classification)  
회귀 (Regression)



새로운 데이터를 학습 모델에 넣으면 정보를 획득함

3

첫 번째 문장이 필자가 생각하는 머신러닝의 정의인 데, 진짜 간단하게 말하면 컴퓨터가 데이터

를 가지고 학습을 한다는 것이 '머신러닝'이다. 이 때 학습이란 데이터를 가지고 패턴을 찾는 것을 의미하고 패턴이란 데이터가 어떻게 분포되었는지, 데이터 어떤 형태로 위치하는지 등을 말한다.

그래서 학습이라는 것은 '데이터의 특성을 파악한다는 것'이고 이러한 학습은 학습 모델이라는 곳에서 진행이 된다. 그래서 분류를 위한 모델이라면 데이터 분류를 할 것이고, 회귀를 위한 모델이라고 하면 실수를 예측하기 위한 데이터의 계형을 그려낼 것이다.

이러한 학습이 끝나면 컴퓨터는 다시 학습 모델에 결과물을 저장해 놓는 데, 바로 데이터의 패턴을 저장한다고 보면 된다.

재밌는 사실은 이 모델에 새로운 데이터를 집어넣는다면 우리가 원하는 정보에 대한 예측치를 얻을 수 있다. 예를 들어 과일 데이터에 대한 분류 학습 모델에 '사과'를 집어 넣으면 '빨간 과일'이라고 분류(예측)한다. 임금 데이터에 대한 회귀 학습 모델에 '실적 수'를 집어 넣으면 '₩ 550,000'이라고 예측한다.

그래서 머신러닝을 다시 정리해서 말하면 '컴퓨터가 데이터를 가지고 학습하고 새로운 데이터가 왔을 때 예측하는 기술'이라고 말할 수 있다.

---

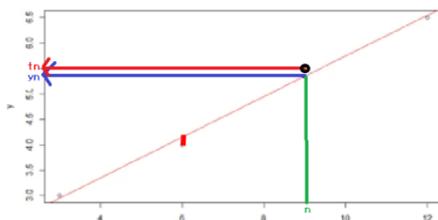
## Machine Learning

- Learning Process

- 모델 선택
- 목적 함수 (Loss Function) 정의
- 최적화 (Optimization)

- 가장 많이 사용되는 평균제곱오차(MSE)

- $J(w_0, w_1) = \frac{1}{N} \sum_{n=0}^{N-1} (y_n - t_n)^2$
- $y_n = y(x_n) = w_0 x_n + w_1$



< Loss function : MSE >  
 (yn = 예측 값, tn = 실제 값)  
 (yn - tn)의 절댓값이 작아야 좋음!

다음은 기계학습 방법론 3단계를 소개하려고 한다.

첫 번째로 무슨 학습 모델을 사용할지 선택을 한다. 아래 예시는 '선형 회귀 모델'이라고 보는데, 데이터의 계형에 맞는 선형 함수를 찾아내서 나중에 입력으로 어떤 수가 왔을 때 함수의 출력 값을 반환하는 것을 목표로 한다.

두 번째로 목적 함수(Loss)를 정의한다. 이걸 모델을 평가하기 위한 함수라고 볼 수 있는데, 가장 많이 사용되는 Loss는 MSE라고 해서 '예측 값과 실제 값의 차이 제곱'에 대한 평균이라고 보면

된다. 직관적으로 봐도 차이가 적을수록 좋은 모델인 것을 유추할 수 있을텐데, 실제로 Loss는 값이 작을수록 좋은 모델이라고 평가한다.

마지막으로 최적화 작업을 한다. 이걸 예측값과 실제 값의 차이가 적게 되도록 Loss인 J함수를 최소화하고, 그 때의 parameter 값( $w_0, w_1$  등)을 찾는 것을 의미한다.

만약 최적화 작업을 잘해가지고 Loss가 최소치가 됐다면, 즉 최적의 선형 회귀 모델이 완성됐다면 그 모델( $y$ )로 예측을 할 수 있을 것이다.

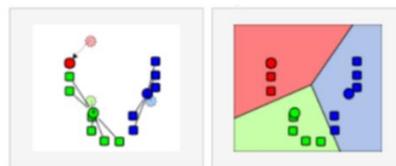
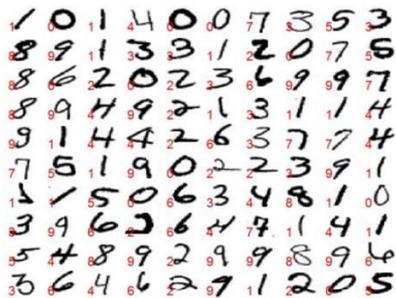
이 내용을 쉽게 말하면, 처음부터 최적화된 선을 찾는 것은 힘들기 때문에 처음에는 무작위로 선을 그리고 이후에는 정답과 예측 값의 차이가 적어지는 방향으로 선을 서서히 옮긴다고 생각하면 된다.

---

## Machine Learning

### • Supervised & Unsupervised

- 지도학습: 학습 데이터가 목표 값을 가지고 있을 때 학습시키기
- 비지도학습: 학습 데이터가 목표 값이 없을 때 학습시키기
- 목표 값: 예측 값이 최종적으로 되고자 하는 값 (정답)



5

다음은 지도 학습과 비지도 학습에 대해 간단하게 이야기하겠다. 지도 학습은 말 그대로 컴퓨터가 학습을 할 때 지도할 데이터가 있다는 것이다. 그래서 학습 데이터가 목표 값이 있는 경우를 의미하는 데, 예를 들어 아래 손 글씨 데이터의 왼쪽 하단에 위치한 숫자들이 바로 목표 값이라고 볼 수 있다. 만약 학습 모델이 손 글씨 데이터를 (0~9)로 분류하는 모델이라고 했을 때, 최적화된 모델을 찾기 위해서 정답 데이터를 이용할 수 있다.

**TMI)** 이전 장에서 소개한 모델은 정말 간단한 2차원 모델에 해당이 되는데 이런 이미지 데이터를 처리하는 경우, 픽셀 하나하나가 입력이 된다. 이미지의 입력이 28x28이므로 784차원 모델을 학습한다고 생각하면 된다. 출력도 10개가 나와서 만약 (1,0,0,...)이면 0, (0,1,0,...)이면 1 이런 방식으로 분류하는 것이다.

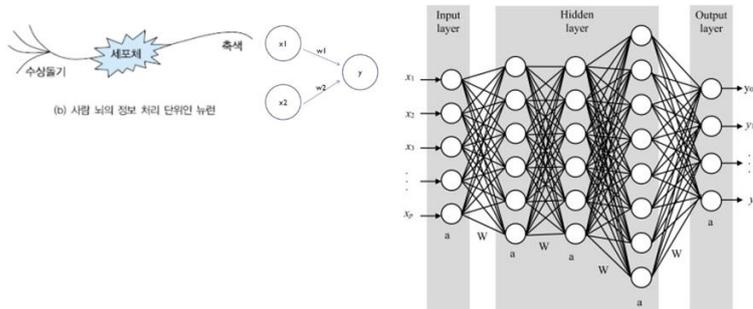
비지도 학습은 반대로 학습 데이터에 목표 값이 없는 경우이다. 따라서 앞장에서 소개한 Loss Function도 정의하기 힘들고 사실상 사람이 직접 평가하는 경우가 많다. 최적화를 할 때도 정답과

의 차이를 이용하는 것이 아니라 거리 공식(ex. 유클리디안, 맨해튼, 해밍 등) 같은 특정 수학 공식에 의존해서 처리하는 경우가 많다. 오른쪽 그림의 예시 같은 경우, 거리 공식을 기반으로 군집화를 하는 모델을 나타낸 것이다.

## Machine Learning

- Deep Learning

- 머신러닝의 학습 전략 중 ANN을 발전시킨 학문
- ANN: Artificial Neural Network (인공신경망)
- DNN: Deep Neural Network



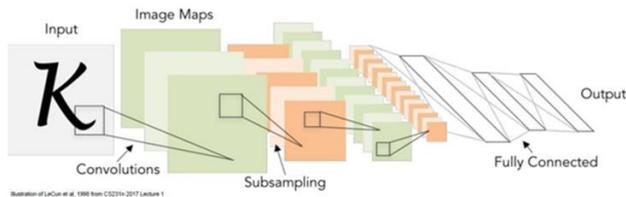
6

다음은 딥 러닝에 대한 설명이다. 앞에서 단순히 (분류를 하기 위한 모델), (회귀를 하기 위한 모델) 이런 식으로 크게 설명했지만 사실 분류와 회귀를 하는 여러 가지 전략이 있다. 대표적으로 KNN, SVM, ANN 등이 있는데 이들 중 ANN을 발전시킨 게 딥 러닝이다. ANN은 뇌의 정보 처리 방법을 모방한 학습 방법이고, 사람의 뉴런을 모방한 모양을 가지고 있다. 다수의 신호를 받아서 하나 혹은 여러 신호를 출력을 하게 되는 '퍼셉트론'이란 객체들로 구성돼 있는 모델이 ANN이라고 생각하면 된다. 이러한 ANN의 층이 깊은 모델이 바로 DNN이다. DNN이 곧 딥 러닝을 의미한다.

---

## Machine Learning

- CNN
  - Convolutional Neural Network
  - 이미지 자체를 입력으로 줄 수 있는 딥러닝 모델
  - 영상 처리에서 좋은 성능을 보이는 모델



7

마지막으로 CNN에 대한 소개이다. 앞에서 소개한 DNN이 딥러닝이라고 했는데 반대로 딥러닝이 DNN이라고 말할 수는 없다. 따라서 DNN은 딥러닝의 포함 관계라고 보면 되고 CNN도 마찬가지이다. CNN의 가장 큰 특징은 이미지 자체를 입력으로 줄 수 있다는 것이다. 따라서 픽셀 별로 다 나눠가지고 모델의 입력으로 주는 DNN과는 달리 이미지의 위치 정보도 활용할 수 있다. 그래서 영상 처리에서 좋은 성능을 보이는 모델이라고 생각하면 된다. CNN에 대해 자세한 내용을 소개할 수는 없지만 그냥 예시만 봤을 때, 이미지가 입력으로 들어오면 네트워크를 거치면서 Feature를 담은 Feature Map들이 형성된다. Feature Map들은 이미지의 특징 정보를 담고 있는 2차원 행렬이라고 생각하면 된다. 이러한 Feature Map들이 네트워크를 다시 거치면서 크기가 작아지기도 하고 개수가 많아지기도 한다. 하지만 결국에는 벡터 형태로 바뀌면서 분류와 같은 문제를 풀 수 있는 것이다.

ex) (1,0,0,0,...)=A, (0,1,0,0,...)=B, (0,0,1,0,...)=C 이런 식으로 말이다.

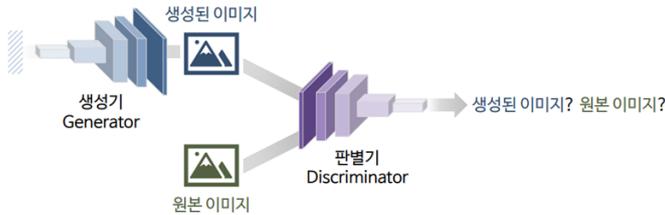
---

## Generative Model

- GAN

- 생성자 신경망과 판별자 신경망이 서로 적대적으로 경쟁하면서, 훈련을 통하여 자신의 작업을 점점 정교하게 수행하는 모델
- 생성기는 가짜 이미지를 실제 이미지처럼 만드는 방법을 학습한다 (비지도 학습)
- 판별기는 이미지를 더 잘 구별하도록 학습한다 (지도 학습)

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



8

이제 본격적으로 GAN을 소개하겠다. GAN은 Generative Adversarial Network이고 한국말로 생산적 적대 신경망이다. 따라서 두 적대적인 신경망이 존재하는 데 바로 생성기와 판별기이다. 생성기는 가짜 이미지를 실제 이미지처럼 만드는 방법을 학습한다. 즉 가짜 이미지 혹은 그냥 랜덤화된 행렬을 입력으로 받아 원본 이미지처럼 이미지를 생성하는 역할을 한다.

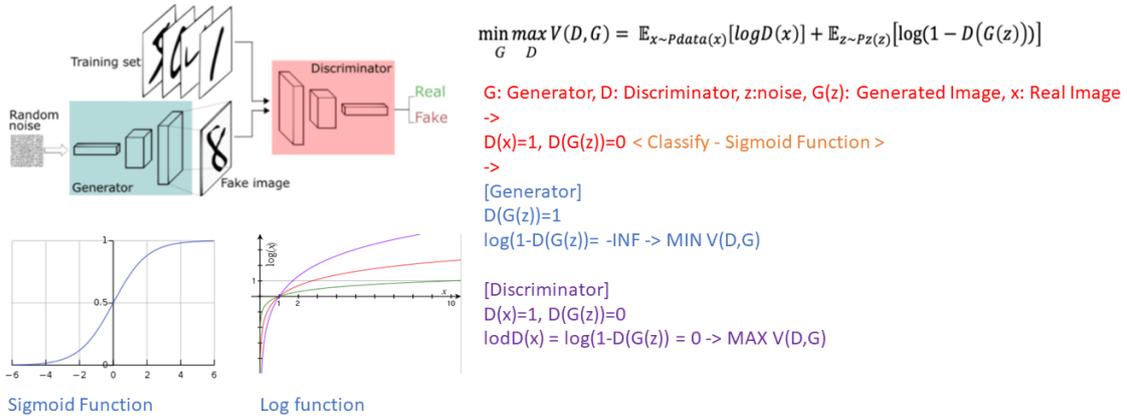
판별기는 이미지를 구별하는 학습을 진행한다. 즉 생성된 이미지와 원본 이미지를 입력으로 받아 생성된 이미지가 진짜인지 가짜인지를 판단하는 분류 학습을 진행한다.

따라서 생성기는 이미지를 생성하는 비지도 학습, 판별기는 생성된 이미지를 판별(분류)하는 지도 학습을 하는 구조로 이루어져 있다.

**TMI)** 원본 이미지가 정답이라고 생각할 수 있는 데, 일반적으로 이미지 자체는 정답 데이터로 인정하지 않으므로 생성기는 비지도 학습을 한다고 말한다.

## Generative Model

- GAN Loss



9

다음은 GAN 모델의 Loss Function을 소개하겠다. 수식에 보이는 G는 생성기, D는 판별기, z는 생성기의 입력에 해당하는 Random noise이다. 따라서 G(z)는 생성기를 통과한 noise이므로 생성된 이미지라고 볼 수 있고 마지막으로 x는 실제 이미지를 의미한다.

D(x)=1 이런 의미는 판별기에 원본을 넣었을 때 1이 된다는 의미인데 1은 원본, 0은 가짜라고 생각하자. 그래서 초기에 학습하기 전에는 D(x)=1이 될 거고 D(G(z))는 0이 될 것이다. 판별기는 분류기라고 앞서 소개했는데 분류를 할 때는 활성화 함수라는 것이 필요하다. 이 때 sigmoid라는 function을 사용해서 만약에 값이 0.5보다 작으면 0, 크면 1 이런 식으로 분류할 수 있다.

생성기(G)는 D(G(z))가 1이 되기를 원한다. 즉 판별기에 생성된 이미지를 넣었을 때 원본이라고 판단되기를 원한다.

판별기(D)는 D(x)=1로, D(G(z))는 0으로 판별하는 것을 원한다. 즉 원본 이미지와 생성된 이미지를 잘 구분해내는 것을 원한다.

이것을 수식에 대입해서 말하면 생성기(G)는 GAN Loss가 MIN이 되기를 원한다.

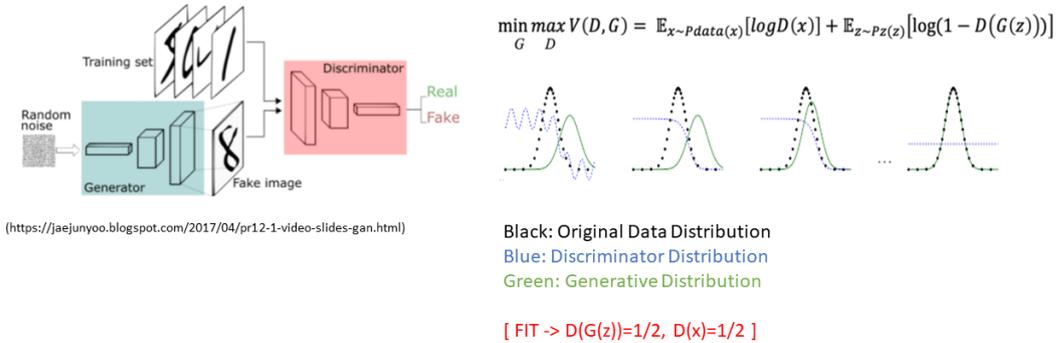
반대로 판별기(D)는 GAN Loss가 MAX가 되기를 원한다.

왜 -Inf가 MIN인데 0이 MAX냐라고 의문을 가진다면, GAN의 판별자는 0과 1사이의 값만 출력하고 log function에서 0은 최댓값이기 때문이라고 답변을 할 수 있다.

학습 이미지들을 잠깐 보면 여러 이미지들이 사용되는 것을 볼 수가 있는데, GAN같은 경우 같은 입력(noise vector)이 오더라도 원본에 해당하는 여러 이미지들 중 하나를 랜덤하게 생성하도록 학습이 된다. 그래서 Loss에 E라는 표현은 Expectation(기댓값)으로, 여러 생성된 이미지들에 대한 평균값으로 계산을 한다는 것이다. 하지만 Loss를 이해하는 데 혼동을 줄 수 있으므로 그냥 '학습 이미지가 여러 개면 그것들 중 하나를 생성하구나' 라고만 이해하고 넘어가자.

## Generative Model

- GAN Loss



10

그림으로 부가 설명을 하겠다.

검은 점선은 실제 데이터의 분포이고 한 점을 이미지라고 생각하면 된다.

파란 점선은 판별기의 output에 대한 분포이고 한 점을 이미지에 대응되는 분류 결과라고 생각하면 된다. (점이 위쪽에 있으면 1(원본), 아래쪽에 있으면 0(가짜)이라고 생각하자)

초록선은 생성기의 output에 대한 분포이고 마찬가지로 한 점을 이미지라고 생각하면 된다. (원본의 분포와 닮아지려는 성질이 있다 = 생성기에서 나오는 이미지들이 원본의 이미지들과 닮아지려는 성질이 있다)

처음에는 판별기가 먼저 훈련을 시작하는 데, 왼쪽에서 두 번째 그림을 보면 원본 이미지들(검은 점선)에 대해서는 1이고 생성된 이미지들(초록 선)에 대해서는 0이라고 나타내게 된다. 따라서 판별기가 훈련이 잘 됐다고 볼 수 있다.

다음은 생성기를 훈련을 시작하는 데, 왼쪽에서 세 번째 그림을 보면 생성된 이미지들(초록 선)이 원본 이미지들(검은 점선)과 닮아지는 것을 분포로 볼 수 있다. 따라서 생성기의 잘 됐다고 볼 수 있다.

이후에도 판별기와 생성기의 훈련을 번갈아서 하게 되는데 결국에는 마지막 그림처럼 판별기가 원본과 생성 이미지를 전혀 구별하지 못 하는 1/2인 상태가 된다.

즉 생성된 이미지가 원본인지 생성된 것인지 구분을 못 하는 상태가 될 때 학습을 종료한다.

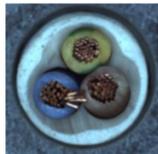
---

## Anomaly Detection

- Anomaly Detection

- Anomaly는 정상의 범주에 벗어나 있는 모든 것을 말함
- Anomaly Detection: 불량 검출, 이상 감지 등 비 정상 여부를 탐지하는 기술 (one class classification)

### [케이블의 단면영상]



→ normal  
**abnormal**

Anomaly Detection

11

다음은 Anomaly Detection에 대한 설명이다. Anomaly란 정상의 범주에 벗어나 있는 모든 것들을 말한다. 따라서 Anomaly Detection이란 불량 검출, 이상 감지 등 비 정상 여부를 탐지하는 기술을 말한다. 아래 그림은 케이블의 단면 영상인 데 살짝 일부가 튀어 나와 있으므로 불량 케이블에 해당된다. 따라서 Anomaly Detection을 하면 abnormal이라는 결과를 출력할 수 있다.

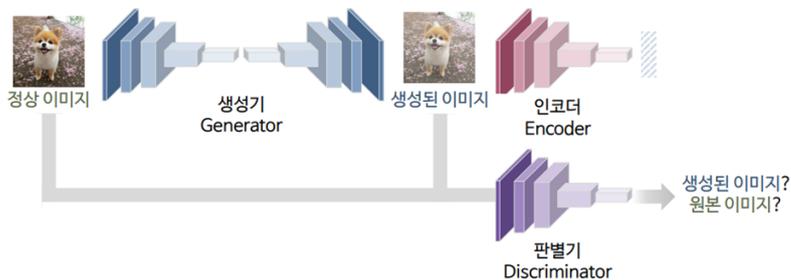
---

## Anomaly Detection

- GANomaly

- GAN을 기반으로 Anomaly Detection을 하는 모델
- 이미지 생성을 위한 학습과 잠재 공간에 대한 학습을 한 번에 진행하기 위하여 제안된 모델

$$\mathcal{L} = w_{adv}\mathcal{L}_{adv} + w_{con}\mathcal{L}_{con} + w_{enc}\mathcal{L}_{enc}$$



12

사실 Anomaly Detection을 구현할 수 있는 모델들은 여러 가지가 있는데, 대표적인 게 GAN Based Anomaly Detection이다. 이 방식이 꽤나 준수한 성능을 보이고 논문들도 계속 나오고 있다. 그래서 2018년 논문인 GANomaly를 가져 왔는데, 현재 최고 수준의 성능을 내는 모델은 아니지

만 초기 GAN모델인 AnoGAN보다 정확도가 높으면서 직관적이라는 장점을 가진다.

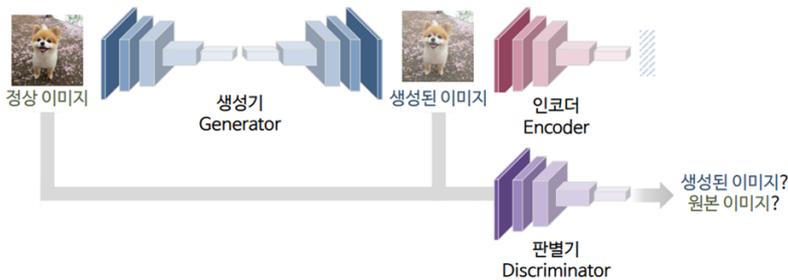
이 모델은 '이미지 생성을 위한 학습' 그리고 '잠재 공간에 대한 학습'을 한 번에 진행하기 위해 제안된 모델인데 이전의 모델들은 이러한 과정을 두 나눠서 진행했다. 따라서 이 모델이 더욱 효율적인 좋은 모델이라고 볼 수 있다.

## Anomaly Detection

- GANomaly

- 1-1. 정상 이미지를 훈련 이미지로 사용해서 정상 이미지를 잘 생성하도록 학습한다
- 1-2. 인코더의 Latent Vector(z2)가 생성기의 Latent Vector(z1)와 유사해지도록 학습한다
2. 테스트 이미지가 모델의 입력으로 오면 Forward 이후 z1과 z2의 차이를 계산한다
3. z1과 z2의 차이가 적으면 정상, z1과 z2의 차이가 크면 비정상으로 간주한다

$$\mathcal{L} = w_{adv}\mathcal{L}_{adv} + w_{con}\mathcal{L}_{con} + w_{enc}\mathcal{L}_{enc}$$



13

GANomaly를 이용한 Anomaly Detection 절차는 위와 같다.

먼저 정상 이미지를 훈련 이미지로 사용해서 정상 이미지를 잘 생성하도록 학습한다. 그리고 동시에 인코더의 잠재 벡터가 생성기의 잠재 벡터와 유사해지도록 학습한다. 여기서 잠재 벡터라는 것은 이미지의 특징을 담고 있는 압축된 벡터라고 생각하면 된다.

학습이 끝나고 테스트 이미지가 모델의 입력으로 들어오면 Forward 이후 두 잠재 벡터의 차이를 계산한다. 여기서 Forward는 모델을 끝까지 통과시킨다는 의미이다.

만약 잠재 벡터의 차이가 적다면 들어온 테스트 이미지가 정상이라고 간주한다. 그렇지 않으면 비 정상이라고 간주한다.

이것이 가능한 이유는 인코더의 잠재 벡터가 정상 이미지(원본과 아주 유사한 생성된 이미지)에 대한 특징을 가지고 있기 때문이다. 즉 입력으로 들어온 비 정상 데이터의 잠재 벡터와는 차이가 발생할 수 밖에 없다.

Ganomaly의 Loss는 모델 위의 수식과 같은 데 세 Loss가 선형 결합된 구조를 지닌다.

특징으로는 기존의 GAN Loss를 사용하지 않는다는 점이다.

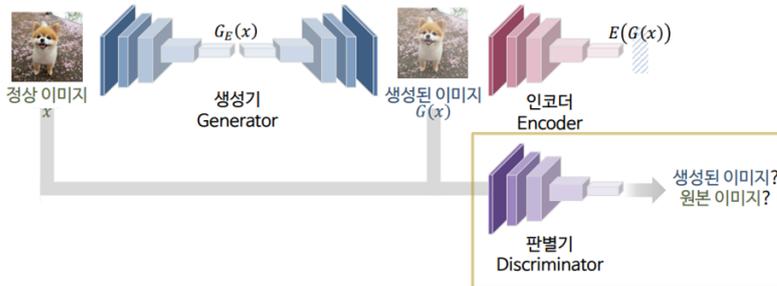
각각의 Loss에 대해서는 그림으로 다시 설명하겠다.

## Anomaly Detection

- Adversarial Loss

- 판별기(f)에서 생성된 이미지(G(x))와 원본 이미지(x)를 잘 구분하지 못하도록 학습한다
- 생성된 이미지와 원본 이미지 모두 원본 이미지라고 인식하게 학습한다
- **판별기를 속이기 위해 원본과 유사한 이미지를 생성하는 용도**

$$\mathcal{L}_{adv} = \mathbb{E}_{x \sim p_X} \|f(x) - \mathbb{E}_{x \sim p_X} f(G(x))\|_2$$



14

첫 번째로 Adversarial Loss이다.

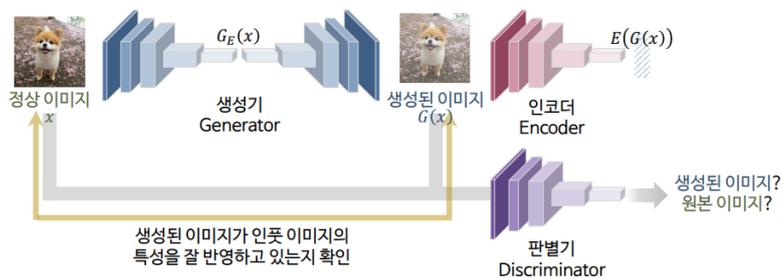
수식에 해당하는 f는 판별기에 해당한다. f(x)와 f(G(x))의 차이를 최소화하는 것을 목표로 한다는 의미이다. 즉 판별기가 생성된 이미지와 원본 이미지 모두를 원본 이미지라고 인식하게 학습하겠다는 것이다. 이 과정에서 원본과 유사한 이미지가 생성된다.

## Anomaly Detection

- Contextual Loss

- 생성된 이미지가 원본 이미지에 더욱 유사하는 방향으로 학습한다.
- G(x) 자체가 x가 되는 방향으로 학습한다.
- **원본과 아주 유사한 이미지를 생성하는 용도**

$$\mathcal{L}_{con} = \mathbb{E}_{x \sim p_X} \|x - G(x)\|_1$$



15

두 번째로 Contextual Loss이다.

Adversarial Loss를 이용해도 원본 이미지를 생성할 수 있다고는 하나 더욱 유사하게 만들고 싶어서 만들어진 Loss이다.

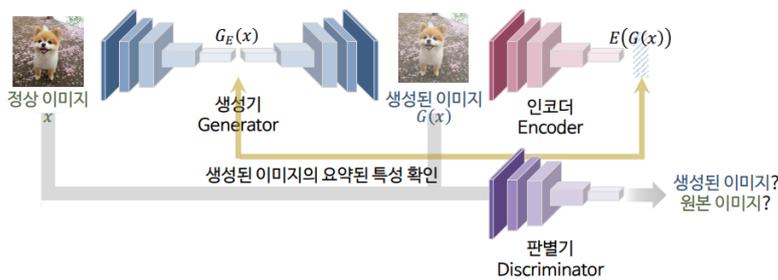
따라서 이번에는  $G(x)$  자체가  $x$ 가 되는 방향으로 학습한다. 여기서 Point는  $G(x)$ , 즉 생성된 이미지가 학습이 된다는 것이다. 따라서 생성된 이미지가 인풋 이미지(정상 이미지)의 특성을 잘 반영하는지 확인하는 과정을 차이를 통해서 구분하는 것이다.

## Anomaly Detection

- Encoder Loss

- 생성기에서 Encoding된  $z$ 와 인코더에서 Encoding된  $z$ 가 같도록 학습한다
- 유사한 Latent Vector 를 생성하는 용도

$$\mathcal{L}_{enc} = \mathbb{E}_{x \sim p_X} \|G_E(x) - E(G(x))\|_2$$



17

마지막으로 Encoder Loss이다.

$G_E(x)$ 는 생성기의 중반부에 해당하는 부분으로 입력 이미지의 특성을 담고 있는 잠재 벡터에 해당한다.  $E(G(x))$ 는 수식 그대로  $G(x)$ 를 Encoder에 입력시켰다는 것으로  $G(x)$ 에 대한 잠재 벡터에 해당한다. Loss를 보면 두 잠재 벡터의 차이가 적어지는 방향으로 학습이 되는 것을 알 수 있다.

따라서 유사한 잠재 벡터를 생성할 수가 있다. 여기서 Point는  $G_E(x)$ 를 학습시키는 것이 목적이 아니라  $E(G(x))$ 를 학습시키는 것이 목적이다.

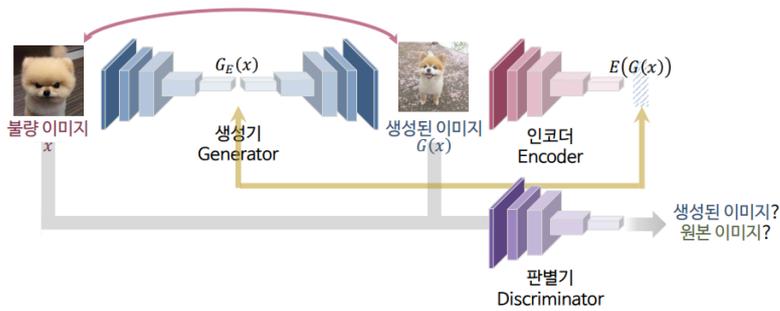
생성된 이미지  $G(x)$ 가 정상 이미지와 유사하므로 CNN 네트워크를 거친  $E(G(x))$ 는 정상 이미지에 대한 특징을 어느 정도 찾겠지만 약간의 부족한 부분을  $G_E(x)$ 로부터 취득해서 메꾼다고 생각해도 된다.

## Anomaly Detection

- Test Process

- 생성기의 Latent Vector( $G_E(x)$ )와 인코더의 Latent Vector( $E(G(x))$ )의 차이를 계산함
- 인코더의 Latent Vector는 확실히 정상 데이터의 Latent Vector에 해당함
- Latent Vector의 차이가 크다는 것은 비정상 이미지가 입력으로 왔다는 의미

$$A(\hat{x}) = \|G_E(\hat{x}) - E(G(\hat{x}))\|_1$$



18

다음은 테스트 과정에 대해서 이야기하겠다.

처음 설명한 것과 마찬가지로 생성기의 잠재 벡터와 인코더의 잠재 벡터의 차이를 계산해서 Anomaly 여부를 탐지한다.

불량 이미지  $x$ 가 오히려 생성된 이미지  $G(x)$ 는 정상 이미지에 해당한다. 또한  $E(G(x))$ 도 정상 이미지의 잠재 벡터에 해당한다.

결국 비정상 이미지( $x$ )에 대한 잠재 벡터( $G_E(x)$ )와  $E(G(x))$ 는 차이가 클 수밖에 없다.

이 차이의 정도를 개발자가 정해서 Anomaly 여부를 탐지하는 것이다. 이것을 Thresholding이라고 하는데, threshold를 초과하면 Anomaly, 초과하지 않으면 Normal이라고 분류하는 것이다.

## Anomaly Detection

- Experiments

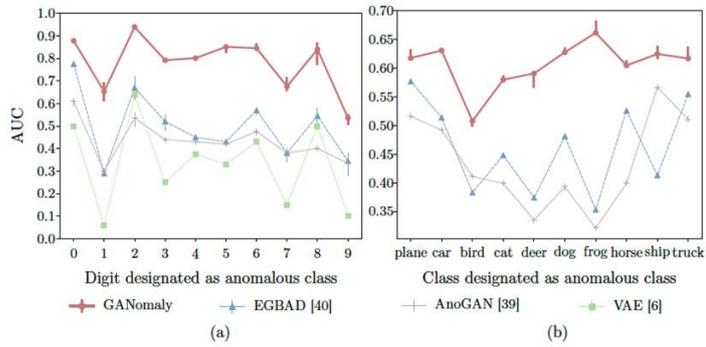


Fig. 4. Results for MNIST (a) and CIFAR (b) datasets. Variations due to the use of 3 different random seeds are depicted via error bars. All but GANomaly results in (a) were obtained from [40].

마지막으로 MNIST, CIFAR10이라는 이미지 데이터로 실험한 결과이다. 이전 모델들인 AnoGAN 따위보다 훨씬 높은 성능을 보인다.