

확률분포활용-GAN

아래 GAN 관련 먼저 쓰~으 들으세요. 1.5배속 괜찮음.

<https://www.youtube.com/watch?v=AVvIDmhHgC4>

- GAN(생성망)의 원리나 공식은 어렵지 않으나 하드웨어 자원이 많이 필요함.
- GAN에서 제공하는 모델을 일부만 튜닝하여 사용하는 방법(간단함) 대하여서는 11월7일(화) 오전 9시에 강의함.

아래의 사이트의 내용을 확실히 학습합니다. *****

<https://datascienceschool.net/02%20mathematics/06.00%206%EC%9E%A5%20%ED%94%BC%EC%A7%80%EC%97%A0%ED%8C%8C%EC%9D%B4%28pgmpy%29%EB%A1%9C%20%EA%B3%B5%EB%B6%80%ED%95%98%EB%8A%94%20%ED%99%95%EB%A5%A0%EB%A1%A0.html>

6장 피지엠패이(pgmpy)로 공부하는 확률론

- 6.1 집합 → [개념 및 파이썬코드] 확률
- 6.2 확률의 수학적 정의와 의미 → [개념 및 파이썬코드] 표본공간 / 확률표본
- 6.3 확률의 성질
- 6.4 확률분포함수 → [개념 및 파이썬코드] 확률질량함수 / 누적분포함수 / 확률밀도함수(중요)
- 6.5 결합확률과 조건부확률 → [개념 및 파이썬코드]
- 6.6 베이즈 정리 → [개념 및 파이썬코드] 베이지안 이라고도 함,
단 베이즈 정리는 간단하게 개념만 잡음/
별도로 다시 자세하게 학습해야함.

GAN (생성형AI)

생성망

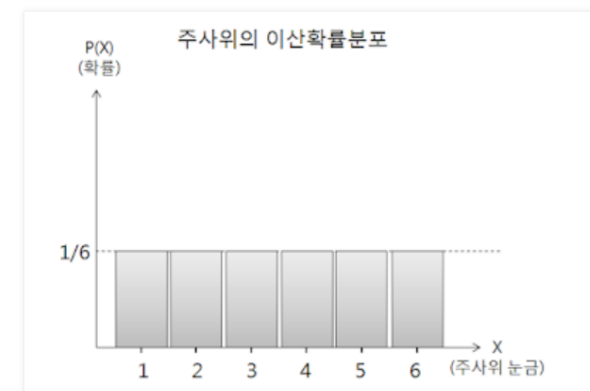
Generative model(생성모델)은 데이터를 생성(모방)하는 것
 $p(x,y)$ = 결합확률 (joint probability) 를 **베이즈정리**로 $p(y|x)$ 를 구하는 것.
데이터들의 분포를 모델링

판별망

Discriminative model(분류모델)은 데이터를 분류하는 것.
 $p(y|x)$ = 조건부확률 (conditional probability) 를 직접 구하는 것.
클래스간의 경계를 학습

GAN에서의 확률분포를 이해하려면, '확률분포'라는 말의 의미를 먼저 이해해야 합니다. 이제 확률분포라는 개념부터 차근차근 짚고 넘어가도록 하겠습니다.

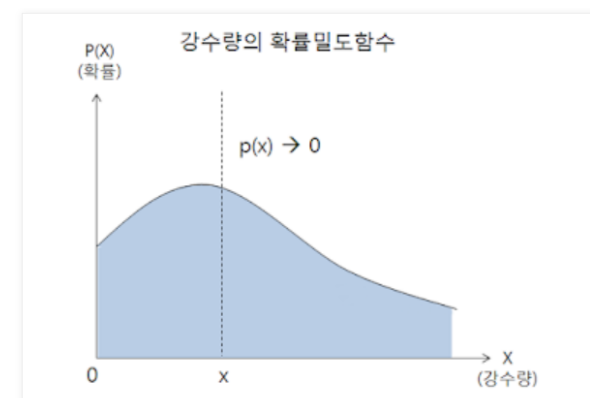
우리가 흔히 생각하는 확률은 어떤 일이 일어날 가능성을 말합니다. 확률분포란 어떤 확률변수 X 와 X 에 대응하는 확률의 쌍이라고 생각할 수 있습니다. 가장 쉬운 예로 주사위를 던지는 일을 생각해봅시다. 여기서의 확률 변수 X 는 주사위를 던져서 나오는 수이고, 가능한 수는 1, 2, 3, 4, 5, 6 입니다. 그리고 각 수마다 해당 수가 나올 확률이 존재합니다. 모두가 알다시피 그 확률은 $1/6$ 로 동일합니다. 이를 좌표계의 형태로 나타내면 아래 그림과 같습니다.



즉, 확률변수 X 가 가질 수 있는 값마다 대응하는 확률이 존재합니다. 이처럼 확률변수 X 가 셀 수 있는 값일때, X 의 확률 분포 $P(X)$ 를 '이산확률분포'라고 합니다. 주사위 눈금의 수는 말그대로 "1, 2, 3, 4, 5, 6"이라고 셀 수 있고, 각각에 대한 확률도 명확히 말할 수 있습니다. $1/6$ 이라고 말이죠.

반면에 확률변수 X 가 셀 수 없는 값이라면 어떨까요? 서울의 연평균 강수량을 예로 들어 봅시다. 강수량이라는 확률변수는 0이상의 실수(real number)로 정의할 수 있습니다. 즉, 셀 수 없다는 말입니다. 0과 1 사이의 모든 숫자를 하나씩 셀 수 있을까요? 0.1 간격으로 나누어 세면 되지 않을까요? 하지만 똑같은 문제가 발생합니다. 0과 0.1 사이에도 무한히 많은 실수가 존재하기 때문입니다.

강수량이라는 확률변수 X 를 셀 수 없을 뿐 아니라, 확률변수 각각에 대한 확률도 명확히 정의하기 어렵습니다. 어떤 해의 강수량이 정확히 125.4533242 일 확률은 얼마일까요? 그저 '0에 가깝다'고 대답할 수 밖에 없습니다. 이처럼 확률변수 X 가 셀 수 없는 연속적인 값일 경우의 확률분포를 특별히 '확률밀도함수'라고 합니다. 특정 확률변수 X 에 대한 확률을 명확히 정의하기 어렵기 때문에 '밀도'라는 개념을 채용하는 것입니다.



확률분포란 임의의 확률변수 X 와 그에 해당하는 확률의 쌍을 말하는 것으로 $P(X)$ 라고 표기할 수 있습니다. 즉, 수학적으로는 특정 변수 x 에 대응하는 확률 $p(x)$ 를 나타내므로 함수의 형태로 생각할 수 있으며, 시각적으로 상상해보자면 위의 그림과 같은 좌표계 상의 그래프로 표현할 수 있습니다.

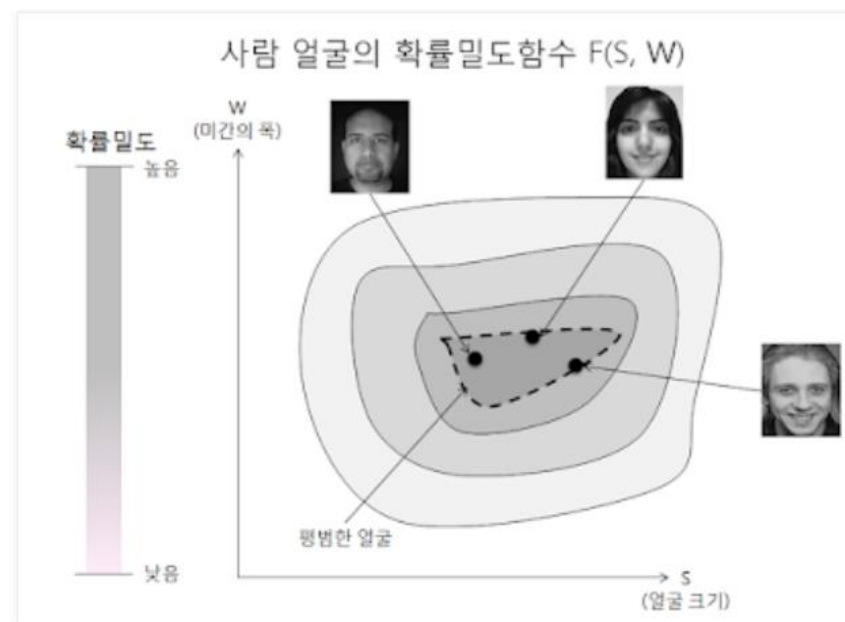
사람얼굴의 확률분포

이제 GAN에서 말하는 데이터의 확률분포를 살펴볼 차례입니다. 여기서 데이터가 '사람 얼굴 사진'이라고 가정하겠습니다. 그렇다면 사람 얼굴의 확률분포란 무엇일까요? 사람의 얼굴을 어떻게 확률변수라는 수치로 표현할 수 있을까요? 또 그에 대한 확률은 어떻게 정의할까요?

앞에서 CNN을 이용한 이미지 분류를 설명할 때, 이미지를 특징 공간 상의 한 점으로 나타낼 수 있다고 했습니다. 그리고 그 특징 공간의 각 좌표축(기저 벡터)는 어떤 특징을 나타내는 것입니다. 사람의 얼굴을 예로 들자면 얼굴 크기, 눈, 코, 입의 상대적 크기, 이마 넓이, 미간의 폭, 쌍꺼풀의 정도 등을 특징으로 선택할 수 있겠죠.

이렇게 생각하면 사람의 얼굴도 변수로 수치화할 수 있습니다. 설명을 간단히 하기 위해 얼굴 크기와 미간의 폭이라는 두 가지 특징만을 고려하겠습니다. 여기서 얼굴 크기를 S , 미간의 폭을 W 라고 하면 어떤 사람의 얼굴 F 는 S 와 W 의 순서쌍 $F(S, W)$ 로 표기할 수 있습니다. 맵소사! 사람 얼굴을 숫자로 바꾸는데 성공했습니다.

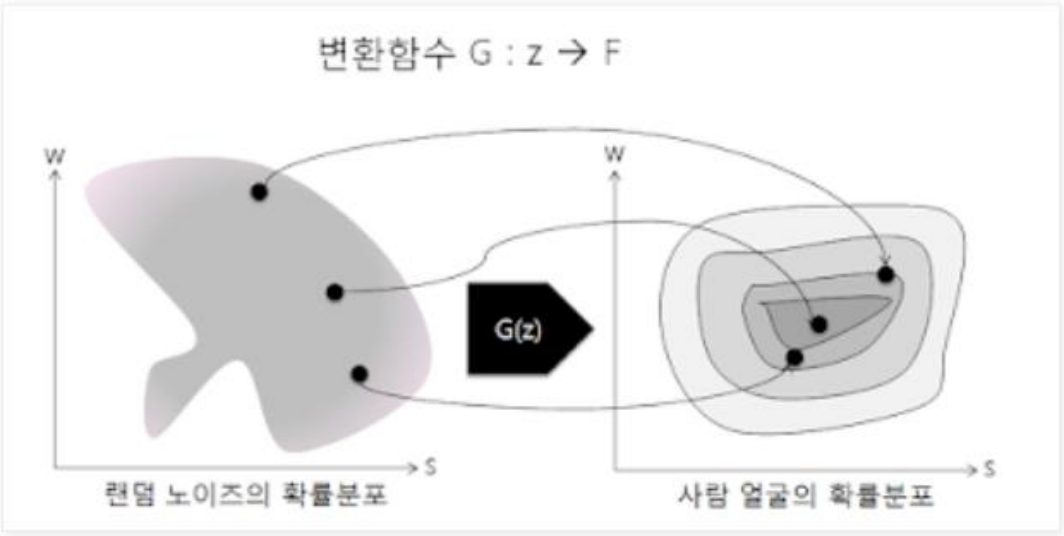
이제 사람 얼굴을 확률변수로 수치화하는데 성공했으니 그에 대한 확률도 정의할 수 있습니다. 그 확률이란 바로 비슷한 얼굴이 존재할 확률을 말합니다. 세상에 나와 얼굴이 정확히 똑같은 사람이 존재할 확률은 0에 가까우므로, 여기서도 밀도의 개념을 빌려와야 합니다. 즉, 사람 얼굴의 확률밀도는 (S, W) 좌표계 상에서 특정 좌표 근처에 얼마나 많은 얼굴이 존재하느냐는 것입니다. 결국 '평범한 얼굴'일수록 그 근처의 확률 밀도는 커질 것입니다. 이를 그림으로 나타내면 아래와 같습니다.



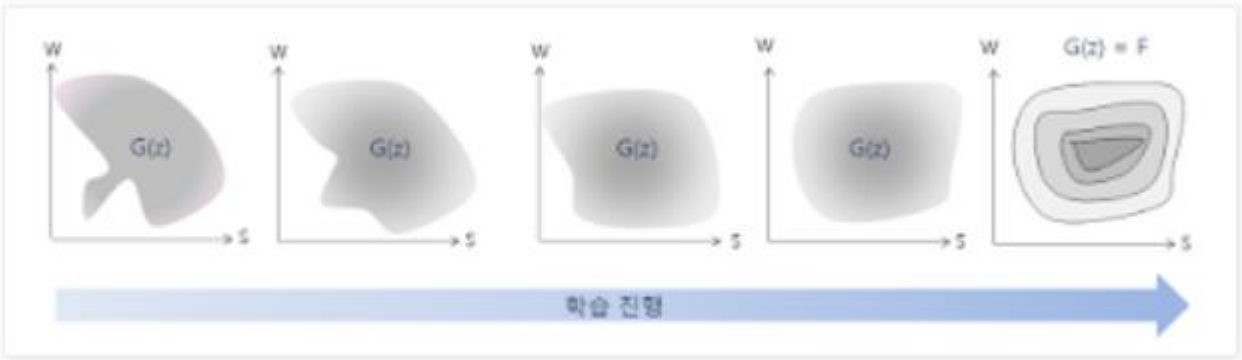
Gan과 확률분포

이제 GAN의 판별자 네트워크 입장에서 확률분포를 생각해 보겠습니다. 판별자의 역할은 주어진 얼굴 $f=(s, w)$ 가 학습 데이터세트에 포함된 진짜 얼굴인지, 생성자가 만들어낸 가짜 얼굴인지를 가려내는 것입니다. 결국 판단의 근거는 주어진 얼굴이 얼마나 그럴싸하냐는 것 입니다. 다른말로 하자면 주어진 얼굴과 비슷한 얼굴이 얼마나 존재하냐는 것이죠. 결국 판별자는 주어진 얼굴이 아래 그림에서 진한 색으로 표시한 영역에 가까울 수록 진짜일 확률이 높다고 판단합니다.

그렇다면 생성자는 판별자를 속이기 위해 어떻게 해야 할까요? 판별자가 비밀로하는 확률분포를 알아내야 합니다. 한 마디로 입력으로 주어지는 랜덤 노이즈의 확률 분포를 사람 얼굴의 확률분포로 변환하는 함수를 학습해야 합니다.



학습 초기에는 생성자가 만들어낸 얼굴이 판별자가 진짜라고 판단하는 영역에서 멀리 떨어져 있습니다. 하지만 수없이 많은 학습을 반복하면서, 생성자가 만들어내는 얼굴의 확률분포는 판별자가 알고있는 진짜 얼굴의 확률분포와 비슷해집니다. 학습이 완료된 시점에서는 생성자가 만들어낸 얼굴의 확률분포와 판별자가 알고있는 얼굴의 확률분포가 일치하게 되고, 판별자로서는 도저히 진짜와 가짜를 구분할 수 없게됩니다. 결국 앞에서 설명한대로 판별자가 정답을 맞힐 확률은 0.5가 됩니다.



GAN을 활용한 이미지 생성 – 아래내용은 대강 읽어봅니다.

<https://hyunsooworld.tistory.com/entry/%EC%B5%9C%EB%8C%80%ED%95%9C-%EC%89%BD%EA%B2%8C-%EC%84%A4%EB%AA%85%ED%95%9C-GAN>

GAN의 공식에 대한 풀이 영상입니다. 1시간:10분부터 봅니다.

**꼭 공식을 이해하지 않아도 텐서플로우나, 파이토치와 같은 딥러닝 프레임워크에서 구현 가능하니
공식에 대한 수학식이 이해되지 않으면 다음 슬라이드에서 프로그램으로 구현하여서 실행하여만 봅니다.**

<https://www.youtube.com/watch?v=uQT464Ms6y8>

텐서플로우로 구현하는 GAN

딥러닝프레임워크

구글 검색자주하여야 함. 시대에 따라 좋은 프레임 워크는 달라짐

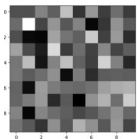
파이토치 2.0 vs 텐서플로

이제 **PyTorch 2.0**과 **TensorFlow**를 비교하여 서로 어떻게 비교되는지 살펴보겠습니다.

- **성능:** PyTorch 2.0은 향상된 성능을 제공하여 이전 버전보다 더 빠르고 효율적입니다.하지만 텐서플로우는 속도와 메모리 효율성 측면에서 여전히 PyTorch 2.0을 능가합니다.
- **사용 편의성:** PyTorch는 사용 편의성과 직관적인 API로 유명하여 개발자들 사이에서 인기 있는 선택입니다.반면 TensorFlow는 학습 곡선이 더 가파르지만 더 많은 유연성과 사용자 지정 옵션을 제공합니다.
- **커뮤니티 지원:** TensorFlow는 PyTorch보다 더 크고 활발한 커뮤니티를 가지고 있습니다. 즉, 개발자가 사용할 수 있는 리소스가 더 많습니다.그러나 PyTorch의 커뮤니티는 빠르게 성장하고 있으며 GitHub 및 Reddit과 같은 플랫폼에서 강력한 입지를 확보하고 있습니다.
- **다른 라이브러리와의 통합:** PyTorch 2.0과 TensorFlow는 모두 고유한 강점과 약점을 가진 강력한 딥 러닝 프레임 워크입니다.PyTorch 2.0은 유연성, 사용 편의성 및 NumPy 및 Hugging Face Transformer와 같은 다른 라이브러리와의 호환성으로 잘 알려져 있습니다.또한 고품질 튜토리얼과 문서를 제작하는 것으로 알려진 활발한 커뮤니티도 있습니다.

텐서플로우로 구현하는 GAN 기초 (속도 느림)

x데이터는 노이즈값



y데이터는 생성하고자하는 자료



코드 해석 잘 안되면 다음 9-10 슬라이드 확인

아래코드를 실행하여 봅니다. (텐서플로우에서 작업내용입니다.) /

활성화 함수 relu, Leakyrelu를 사용하는 이유 도 미리 알고 작업합니다. <https://balmstory.tistory.com/52>

<https://teddylee777.github.io/tensorflow/vanilla-gan/>

이모듈 추가해야함.

```
1 import matplotlib.pyplot as plt
2 NOISE_DIM=10
3 tmp=tf.random.normal([NOISE_DIM,NOISE_DIM]) #
4 plt.imshow(tmp,cmap='gray'),tmp
```



```
from tensorflow.keras.layers import Dense, LeakyReLU, Dropout, Input
from tensorflow.keras.models import Sequential, Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.optimizers.legacy import *
from tensorflow.keras.initializers import RandomNormal
import numpy as np
import matplotlib.pyplot as plt
```

결국 이코드는 무작위로 생성된 노이즈 샘플이 (Noise_dim=10해서 가로세로 10픽셀값의 무작위생성) 원본값 숫자 이미지와 같지 않다고 판별하여 계속 같은 이미지(확률밀도)를 만들게 하는 생성망(GAN) 코드임


```
x_train = x_train.reshape(-1, 784)
x_train.shape
```

(60000, 784)

Generator

생성자

generator를 정의합니다.

```
generator = Sequential([
    Dense(256, input_dim=NOISE_DIM), 1
    LeakyReLU(0.2),
    Dense(512),
    LeakyReLU(0.2),
    Dense(1024),
    LeakyReLU(0.2),
    Dense(28*28, activation='tanh'), 2
])
```

- (1) Input한 노이즈이미지를
- (2) 28*28=784 자료로 만들고

Discriminator

판별자

discriminator를 정의합니다.

```
discriminator = Sequential([
    Dense(1024, input_shape=(784,)), 3
    LeakyReLU(0.2),
    Dropout(0.3),
    Dense(512),
    LeakyReLU(0.2),
    Dropout(0.3),
    Dense(256),
    LeakyReLU(0.2),
    Dropout(0.3),
    Dense(1, activation='sigmoid') 4
])
```

- (3) 판별자에 생성자에서 생성된 784 이미지를 넣어서
- (4) x데이터와의 같음여부를 판별함.

Gan

1~4 과정을 진행

generator와 discriminator를 연결합니다.

```
# discriminator는 학습을 하지 않도록 하며, g
discriminator.trainable = False
gan_input = Input(shape=(NOISE_DIM,))
x = generator(inputs=gan_input)
output = discriminator(x)
```

학습

```
# discriminator와 gan 모델의 loss 측정을 위한 list 입니다.
d_losses = []
g_losses = []

for epoch in range(1, EPOCHS + 1):
    # 각 배치별 학습
    for real_images in get_batches(x_train, BATCH_SIZE):
        # 랜덤노이즈 생성
        input_noise = np.random.uniform(-1, 1, size=[BATCH_SIZE, NOISE_DIM])

        # 가짜 이미지 데이터 생성
        generated_images = generator.predict(input_noise)

        # Gan에 학습할 X데이터 정의
        x_dis = np.concatenate([real_images, generated_images])
```

Gan에 학습할 Y 데이터 정의 -> # Discriminator 훈련
-> # Gan 훈련 -> # Discriminator의 판별 학습을 방지합니다

```
1 generator = Sequential([
2     Dense(256, input_dim=NOISE_DIM),
3     LeakyReLU(0.2),
4     Dense(512),
5     LeakyReLU(0.2),
6     Dense(1024),
7     LeakyReLU(0.2),
8     Dense(28*23, activation='tanh'),
9 ])
```

****생성자 마지막에 28*23으로 넣으면**

```
1 discriminator = Sequential([
2     Dense(1024, input_shape=(784,)), kernel_
3     LeakyReLU(0.2),
4     Dropout(0.3),
5     Dense(512),
6     LeakyReLU(0.2),
7     Dropout(0.3),
```

판별자의 784와 맞지않게됨

```
1 ## GAN
2 # generator(생성자)와 discriminator(판별자)를 연결합니다.
3
4 discriminator.trainable = False
5 gan_input = Input(shape=(NOISE_DIM,))
6 x = generator(inputs=gan_input)
7 #x=generator()
8 output = discriminator(x)
9
10 # GAN모델정의
11 gan = Model(gan_input, output)
12
13 # gan.컴파일
14 gan.compile(loss='binary_crossentropy', optimizer=adam)
```

**생성자를 받아서 판별자에 넣었을때
판별자의 784와 불일치**

ValueError: Exception encountered when calling layer "sequential_17" (type Sequential).

Input 0 of layer "dense_58" is incompatible with the layer: expected axis -1 of input shape to have value 784, but received input with shape (None, 644)

Call arguments received by layer "sequential_17" (type Sequential):

- inputs=tf.Tensor(shape=(None, 644), dtype=float32)
- training=None
- mask=None

에러

그외 텐서플로우로 구현하는.

가나다라마-GAN Model 한글 손 글씨 이미지 생성

<https://www.youtube.com/watch?v=z-LIpUX-lpc>

<https://github.com/MijeongJeon/Korean...>

**반드시 작업해야 하는 자료는 아님.
아이디어 차원에서 볼것
코드 에러 생기면 수정하면서 하세요.**

파이토치로 구현하는 손글씨 생성GAN 텐서플로우와 거의 똑같음.

<https://kingnamji.tistory.com/11>

파이토치, 흑백사진->컬로로 GAN

<https://pseudo-lab.github.io/Tutorial-Book/chapters/GAN/intro.html>

생성 모델링(GENERATIVE MODELING)

컬러 이미지 생성 모델 구축

1. GAN 소개
2. 데이터 탐색
3. GAN
4. pix2pix
5. 참고 문헌

생성모델 정리하기

<https://wikidocs.net/139125>

대강읽기(이런GAN이 있구나 정도만 알아도 됨)

텐서플로우 허브에 각 모델에 대한 자료 있음.

https://www.tensorflow.org/tutorials/generative/style_transfer?hl=ko

GAN의 정량적 평가

<https://velog.io/@tobigs-gm1/evaluationandbias>

<https://velog.io/@viriditass/GAN%EC%9D%80-%EC%95%8C%EA%B2%A0%EB%8A%94%EB%8D%B0-%EA%B7%B8%EB%9E%98%EC%84%9C-%EC%96%B4%EB%96%A4-GAN%EC%9D%B4-%EB%8D%94-%EC%A2%8B%EC%9D%80%EA%B1%B4%EB%8D%B0-How-to-evaluate-GAN>

문서 하단에 넘파이를 이용한 FID 공식있음

*FID(FID(Frechet Inception Distance)점수는 실제 이미지와 생성된 이미지에 대해 computer vision 특징에 대한 통계 측면에서 두 그룹이 얼마나 유사한지 즉, 벡터 사이의 거리를 계산하는 메트릭이다.)

7. Generative model(생성 모델)

(1) 생성 모델 아이디어

(2) 생성 모델을 위한 사전지식

1) 생성모델의 역사(★작성중)

2) Variational Inference & Wake sleep algorithm(★작성중)

(3) 생성 모델들

1) VAE(Variational Auto-Encoder)

2) Generative Adversarial Networks (GANs)

3) DCGAN(Deep Convolution GAN)

4) SRGAN(Super Resolution GAN)

5) Cycle GAN

6) Disco GAN

7) Improved Techniques for Training GANs

(4) 생성 모델의 평가

GAN의 저작권 이슈

저작권 이슈에 대해 이해 하기

Gan은 노이즈데이터에서 확률밀도로 생성하는 가짜 데이터이기 때문에 저작권이슈에 자유롭긴 하지만 점점더 많은 이슈가 있기 때문에

 AI타임스
<https://www.aitimes.com/news/articleView>

미 저작권청 "AI가 만든 그림은 저작권 대상 아니다"
2023. 3. 16. — 생성 인공지능(AI)으로 만든 이미지는 앞으로도 미국에서 저작권 보호를 받을 수 없게 됐다. 전체 창작물 가운데 인간 고유의 창의성과 아이디어가 ...

 지디넷코리아
<https://zdnet.co.kr/view>

美 저작권청, 사람의 창의성 입증된 AI작품 저작권 인정
2023. 3. 19. — 이번 가이드라인은 AI를 사용해 생성된 자료를 포함하는 저작물을 심사 ... 최근 저작권청은 이미지생성AI 미드저니를 사용해 만든 그래픽노블의 저작권을 ...

 webzine-copyright.or.kr
http://webzine-copyright.or.kr/copyright_2305


인공지능(AI)이 생성한 콘텐츠도 저작권 등록이 가능할까?
고도화된 AI 기술은 방대한 양의 기존 저작물로 학습 또는 훈련을 한 후 이를 바탕으로 새로운 콘텐츠를 생성하는 데 이용된다. 이러한 생성형 AI를 이용하여 만든 콘텐츠 ...

[저작권동향] 공지능(Generative AI) 산업 현황 보고서

<https://www.copyright.or.kr/information-materials/trend/the-copyright/viewPress.do?brdctsno=51810>

AI 활용 가이드 라인

[https://blog.naver.com/muhayuιν/223134096650](https://blog.naver.com/muhayuिन/223134096650)

 연합뉴스
<https://www.yna.co.kr/최신기사>

"우리 정보 공짜수집·이용 안 돼"...늘어나는 챗GPT 차단 사이트
2023. 8. 28. — 오픈AI는 GPT봇을 공개하면서 정보 수집·이용을 원하지 않는 사이트 소유자에게 GPT봇의 크롤링을 차단하는 방법을 안내했다. GPT봇 공개 후 첫 주인 지난 ...