

22 - 1 딥러닝 및 응용 과제 2

GAN

조교 김수형, 조환희

ksh970404@hanyang.ac.kr

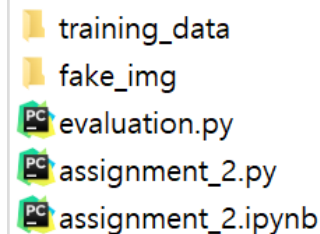
straw88@hanyang.ac.kr

과제 개요

- 과제목표 : GAN 네트워크를 완성하여 이미지 생성 task 수행
- 세부사항

1. 실습 때 배운 코드만 사용하여 GAN 네트워크를 완성하여 이미지 생성하는 것이 목표 (Generator는 제공됨, 수정불가)

- Generator 코드 제공됨, 수정 불가
- Transformer 등 사용불가능



2. 제공된 dataset (celeb A)으로 이미지 생성

3. Discriminator와 다양한 방법들의 변화를 주며, 생성된 fake 이미지와 FID score의 변화를 보고서로 서술한다.

제공된 데이터셋

- **training_data**
 - celebA dataset
 - <https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html>
 - RGB image



- 과제에서는 crop하여 input image **64x64** 사용

제공된 코드

- assignment_2.ipynb
 - 모델 학습과 fake 이미지 생성 및 저장 코드

```
import torch.nn as nn
import torch.utils.data
import torchvision.datasets as datasets
import torchvision.transforms as transforms
import numpy as np
from PIL import Image

from tqdm import tqdm

torch.manual_seed(0)
torch.cuda.manual_seed(0)
torch.cuda.manual_seed_all(0)

class Generator(nn.Module):
    def __init__(self, ):
        super(Generator, self).__init__()
        self.main = nn.Sequential(
            nn.ConvTranspose2d(128, 64 * 8, 4, 1, 0, bias=False),
            nn.BatchNorm2d(64 * 8),
            nn.ReLU(),
            nn.ConvTranspose2d(64 * 8, 64 * 4, 4, 2, 1, bias=False),
            nn.BatchNorm2d(64 * 4),
            nn.ReLU(),
            nn.ConvTranspose2d(64 * 4, 64 * 2, 4, 2, 1, bias=False),
            nn.BatchNorm2d(64 * 2),
            nn.ReLU(),
            nn.ConvTranspose2d(64 * 2, 64, 4, 2, 1, bias=False),
            nn.BatchNorm2d(64),
            nn.ReLU(),
            nn.ConvTranspose2d(64, 3, 4, 2, 1, bias=False),
            nn.Tanh()
        )

    def forward(self, input):
        # input data는 [batch size, 128, 1, 1]의 형태로 주어야합니다.
        return self.main(input)
```

```
class Discriminator(nn.Module):
    # 모델의 코드는 여기서 작성해주세요

    def __init__(self):
        super(Discriminator, self).__init__()

    def forward(self, input):
        return output

if __name__ == "__main__":
    # 학습코드는 모두 여기서 작성해주세요

    data_path = 'training_data/'

    dataset = datasets.ImageFolder(root=data_path,
                                   transform=transforms.ToTensor())

    device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")

    generator = Generator().to(device)
    discriminator = Discriminator().to(device)

    # FID score 측정에 사용할 fake 이미지를 생성하는 코드입니다.
    # generator의 학습을 완료한 뒤 마지막에 실행하여 fake 이미지를 저장하시기 바랍니다.
    test_noise = torch.randn(3000, 100, 1, 1, device=device)
    with torch.no_grad():
        test_fake = generator(test_noise).detach().cpu()

    for index, img in enumerate(test_fake):
        fake = np.transpose(img.detach().cpu().numpy(), [1, 2, 0])
        fake = (fake * 127.5 + 127.5).astype(np.uint8)
        im = Image.fromarray(fake)
        im.save("./fake_img/fake_sample{}.jpeg".format(index))
```

제공된 코드

- assignment_2.ipynb
 - 생성된 fake 이미지를 평가하는 코드

```
import os
import torch

from pytorch_fid.fid_score import *

os.environ['KMP_DUPLICATE_LIB_OK']='True'

real_img_path = 'training_data/celeba/'
fake_img_path = 'fake_img/'

device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")

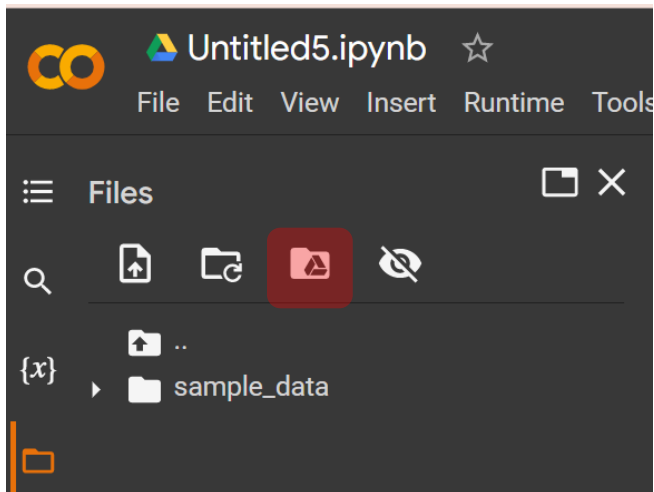
torch.manual_seed(0)
torch.cuda.manual_seed(0)
torch.cuda.manual_seed_all(0)

if __name__ == "__main__":
    fid = calculate_fid_given_paths(
        paths=[real_img_path, fake_img_path],
        batch_size=128,
        device=device,
        dims=2048
    )

    print("fid score : {}".format(fid))
```

Colab과 Google Drive 연동

- File – mount



Permit this notebook to access your Google Drive files?

Connecting to Google Drive will permit code executed in this notebook to modify files in your Google Drive until access is otherwise revoked.

No thanks

Connect to Google Drive




```
from google.colab import drive  
drive.mount('/content/drive')
```

```
path='drive/MyDrive/Colab Notebooks/data/faces/'
```

- 주의 : 제출시 **dataset path 로컬로 수정**
- 드라이브와 연동해야 동작되는 코드 제출 금지

Colab 폴더 다운로드

방법 1. zip 파일로 바꿔서 다운로드



```
!zip -r fake_img.zip fake_img
```

방법 2. google drive로 drag해서 이동 후 drive에서 받기

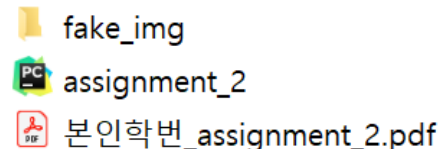
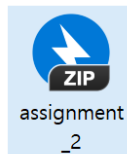
점수 산출

- 코드 (70%)
 - assignment_2.ipynb 혹은 assignment_2.py (파이썬 파일 **한 개 로만 작동** 가능 하도록) (10%)
 - 모델구현 및 실행 (40%)
 - **FID 70이하 (20%)**
- 보고서 (30%) **page 3장**
 - 코드 설명
 - 모델(코드)에 대한 **설명** 명시 (15%)
 - 실험결과
 - 성능향상을 위해 진행한 실험들의 **성능** 비교 (15%)

ex) layer 수, unit 수, dropout 추가, optimizer 변화, hyperparameter 조정 등

과제 조건

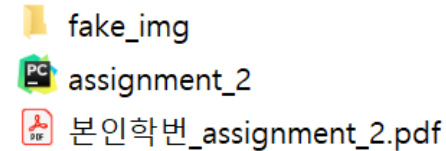
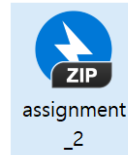
- 환경
 - 프로그래밍 언어 : **Python 3.7, pytorch 3.7~버전**
 - OS : Windows or Colab
 - 보고서 : **PDF page 3장** (word, notion, hwp, ppt, ...)
- 제출 사항
 - **assignment_2.zip**
 - 파이썬 파일 : **assignment_2.ipynb** 혹은 **assignment_2.py**
 - 결과 보고서 : **본인학번_assignment_2.pdf**
 - 생성한 fake 이미지가 저장된 폴더 : **fake_img**



주의 사항

- **파일명** 반드시 준수

- assignment_2.zip



- assignment_2.ipynb 혹은 assignment_2.py
- 본인학번_assignment_2.pdf
- fake_img

- 제출 기한 : **2022.06.17 (23:59)**
- 제출 : LMS
- 추가 제출 기한 **없음.**
- 점수 비중 : **코드 70% 보고서 40%**
- **코드 주의 사항**
 - 데이터 Local path로 변경
 - wandb 포함한 코드 제거

Thank you!
