HW3
Diffusion Generative Model

Yi-Lun Wu 吳易倫
yilun.ee08@nycu.edu.tw

Submission Deadline:

2022/12/06 08:59 a.m.

Submit to E3

Hard deadline, No extensions

Goals

- Implementing difussion generative models.
- Evaluating generative models in terms of FID.

Dataset MNIST

- There are 60000 handwritten digits in size 28x28.
- All images have been converted to RGB for ease of processing.

How to Evaluate Generative Models?

Calculate the divergence from real probability distribution to generative probability distribution.

- Pros:
  - The evaluation result is accurate.
- Cons:
  - Real probabilities is hard to estimate.
  - Not all generative models can produce probabilities for generated data.

Frechet Inception Distance (FID)

The equality $p_d(\cdot) = p_g(\cdot)$ holds except for a non-measurable set if and only if

$$\int p_d(x) f(x) dx = \int p_g(x) f(x) dx$$

for a basis $f(x)$, where $f(x)$ are polynomials of the data x.

For the practical reason, we only consider first two moments of probability distributions: mean and covariance.

The divergence of 2 probability distributions is estimated by Fréchet distance:

$$\|m_d - m_g\|_2^2 + \mathrm{Tr}\left(C_d + C_g - 2(C_d C_g)^{1/2}\right)$$

Calculating FID in Python

We use `pytorch_gan_metrics` to calculate FID in this homework:

```
pip install pytorch_gan_metrics
```

- Github Project Page
- Easy to integrate standard FID calculation into your training loop.
- Provide a standalone script to calculate the FID in the console.

See Github home page for other details.

Calculating FID in Console

Calculate FID using the script provided by `pytorch_gan_metrics`:

```
$ python -m pytorch_gan_metrics.calc_metrics \

      --path path/to/images \

      --stats path/to/mnist.npz

3.345186455974482 0.04813915263223895 93.22019768434347
```

- The last value of output is FID, i.e., 93.22.
- This requires ~3GB GPU RAM and ~3GB CPU RAM. The computation time on GTX-1080 is ~30seconds.
- `path/to/images`: A folder containing your generated images (recursively include subdirectory).
- `mnist.npz`: The precalculated mean and covariance of training data. This file can be downloaded from E3.

Grading Scheme

- FID is evaluated between 10000 generated images and training dataset.
- The FID you calculated is identical to the FID calculated by TA.

FID	**Points**
< 40	100
40 ~ 100	linearly between 0 and 100
> 100	0

Spec

- Implement a difussion-based generative model.
- Python 3.6 or later version.
- You can google, but you need to write your own code.
- Pretrained weights and external data are forbidden, so you should train your model from scratch.

Python Package White List

Deeplearning Packages (equipped with autograd engine)			
TensorFlow (include Keras)	PyTorch	MXNet	JAX
Deeplearning Related Packages			
torchvision			
Machine Learning Packages			
sklearn	xgboost		
Others			
pandas	click	tqdm	tensorboardX
tensorboard	opencv-python		

Submission File (1) - Source Code

- Zip your source code into **StudentID.zip** and submit to E3, where StudentID must be replaced with your student ID.
- You must add a readme file (.pdf, .txt, .md are ok) to clearly state how to create python environment and how to run your code to get the similar result.
- Do not include dataset in **StudentID.zip**, please add the description about dataset path in readme.
- Do not include model weights.

Submission File (2) - Images

- The number of generated images: 10000.
- The dimension of generated images: 28x28 RGB.
- The format of generated images: png.
- The filename of generated images: 00001.png ~ 10000.png.
- Zip these images into a .zip file without any subdirectory in this .zip.
  - MacOS and Linux: You can use `unzip` to check the following messages in the console:

✅ Correct output

```
$ unzip -l student_id.zip
Archive:  student_id.zip
  Length      Date    Time    Name
---------  ---------- -----   ----
      919  2022-11-07 08:38   00001.png
      865  2022-11-07 08:38   00002.png
...
```

❌ Incorrect output

```
$ unzip -l student_id.zip
Archive:  student_id.zip
  Length      Date    Time    Name
---------  ---------- -----   ----
      919  2022-11-07 08:38   images/00001.png
      865  2022-11-07 08:38   images/00002.png
...
```

Bad

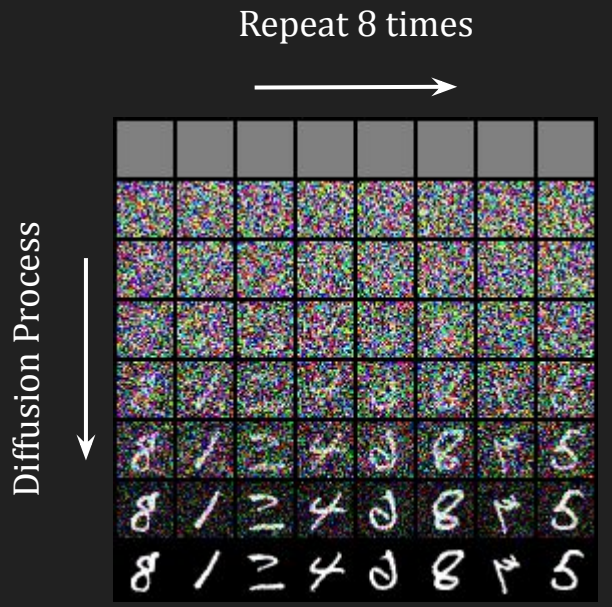
  - Windows: Please refer to the structure of mnist.zip (the training data of this homework).

Submission File (2) - Images

- For MacOS users, do not include `.DS_Store` and `__MACOSX` in your zip file.
- The name of zip file must be student_id.zip where student_id is your student ID.

Submission File (3) - Diffusion Process

- Start the diffusion process from noise 0 and repeat it 8 times to get 8 different results.
- Diffusion step must be divided into 7 equal parts. For example, if the total number of steps is 1000, the result should be recorded every 142 steps.
- Arrange these 64 images into a form similar to the image on the right and save it as a .png file. Upload this png file to E3.



Conclusion: There are 3 Files You Need to Upload to E3

1. Source code (.zip)
2. Generative images (.zip)
   - 10000 images
   - 00001.png ~ 10000.png
   - 28x28 RGB images
   - **NO subdirectory**
   - **NO .DS_Store and __MACOSX**
3. Diffusion process (.png)
   - start from noise 0
   - repeat 8 times
   - total steps are divided into 7 equal parts
   - save as .png

Appendix

FID: 106.7

FID: 53.23

FID: 39.01

FID: 20.8