

---

# Machine Learning HW11

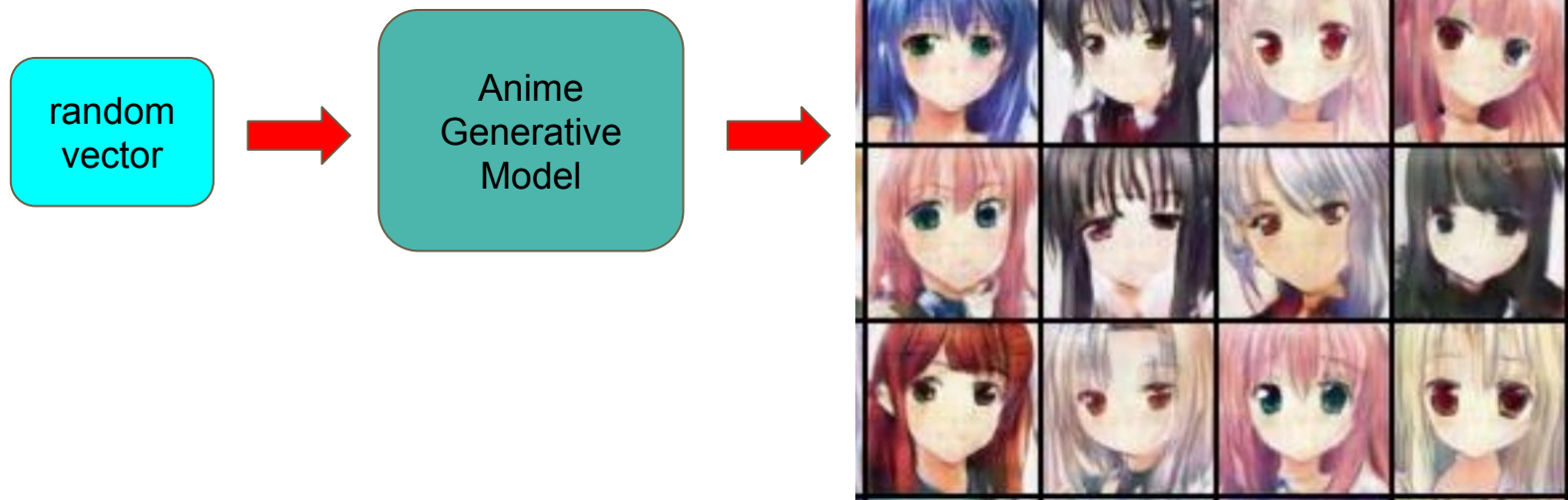
ML TAs

[ntu-ml-2020spring-ta@googlegroups.com](mailto:ntu-ml-2020spring-ta@googlegroups.com)

---

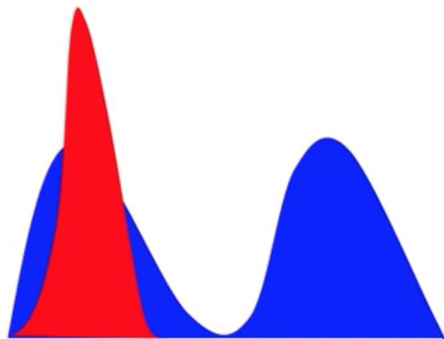
# Task Introduction

# Image Generation



# Mode Collapse

Generated  
Distribution



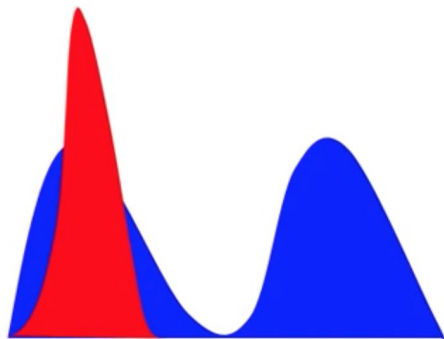
Data  
Distribution



<https://youtu.be/0CKeqXI5IY0?t=6099>

# Mode Collapse

Generated  
Distribution



Data  
Distribution



<https://youtu.be/0CKeqXI5IY0?t=6099>

# Dataset

# Data Collections

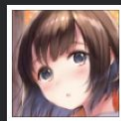


Crypko

## Crypko Beta

GAN × Smart Contract

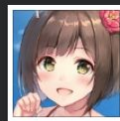
--- the next generation cryptocollectible game.



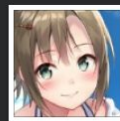
21.jpg



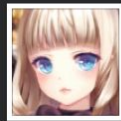
22.jpg



23.jpg



24.jpg



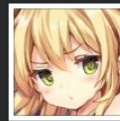
28.jpg



29.jpg



30.jpg



31.jpg



35.jpg



36.jpg



37.jpg



38.jpg

感謝 Arvin Liu(其他作業的助教) 蒐集 dataset  
這個網站目前已經不支援生成功能了

<https://crypko.ai/#>

# Report



# Report <sup>1/3</sup>

- (2.5%) 訓練一個 model。
  - (1%) 請描述你使用的 model(可以是 baseline model)。包含 generator 和 discriminator 的 model architecture、loss function、optimizer 參數、以及訓練 step 數(或是 epoch 數)。
  - (1.5%) 請畫出至少 16 張 model 生成的圖片。



1.b 範例 output

# Report 2/3

- (3.5%) 請選擇下列其中一種 model: WGAN, WGAN-GP, LSGAN, SNGAN (不要和 1. 使用的 model 一樣, 至少 architecture 或是 loss function 要不同)
  - (1%) 同 1.a, 請描述你選擇的 model, 包含 generator 和 discriminator 的 model architecture、loss function、optimizer 參數、及訓練 step 數(或是 epoch 數)。
  - (1.5%) 和 1.b 一樣, 就你選擇的 model, 畫出至少 16 張 model 生成的圖片。
  - (1%) 請簡單探討你在 1. 使用的 model 和 2. 使用的 model, 他們分別有何性質, 描述你觀察到的異同。

WGAN: 改 loss function

<https://github.com/eriklindernoren/PyTorch-GAN/tree/master/implementations/wgan>

WGAN-GP: 改 loss function

[https://github.com/eriklindernoren/PyTorch-GAN/tree/master/implementations/wgan\\_gp](https://github.com/eriklindernoren/PyTorch-GAN/tree/master/implementations/wgan_gp)

LSGAN: 改 loss function

<https://github.com/eriklindernoren/PyTorch-GAN/blob/master/implementations/lsgan/lsgan.py>

SNGAN: 改 model architecture

- 在 discriminator 的每個 module 前套上 `torch.nn.utils.spectral_norm()`

# Report 3/3

- (4%) 請訓練一個會導致 mode collapse 的 model。
  - (1%) 同 1.a，請描述你選擇的 model，包含 generator 和 discriminator 的 model architecture、loss function、optimizer 參數、及訓練 step 數(或是 epoch 數)。
  - (1.5%) 請畫出至少16張 model 生成且具有 mode collapse 現象的圖片。
  - (1.5%) 在不改變 optimizer 和訓練 step 數的情況下，請嘗試使用一些方法來減緩 mode collapse。說明你嘗試了哪些方法，請至少舉出一種成功改善的方法，若有其它失敗的方法也可以記錄下來。

如果同學沒有想法的話，可以使用 baseline model 訓練多一點 epochs，或許就能觀察到 mode collapse 的現象。這邊提供幾種可能的改進的方法：使用 WGAN 系列、SNGAN、加 dropout layer、對 input image 加上 noise。

# Submission <sup>1/4</sup>

GitHub 上的 hw11-<account> 裡面要包含以下 (還有後面幾頁的) 檔案:

- report.pdf
- \*.py
  - 請上傳你所有會需要用到的 .py 檔
  - ex: train\_gan.py, train\_wgan.py, model.py
- checkpoints/p1\_g.pth
  - 你使用在 Report Problem 1 的 generator
  - 如果 model 太大, 同學可以使用後面的 init.sh 來下載, 請記得讓下載後的路徑是這個路徑
- checkpoints/p2\_g.pth
  - 你使用在 Report Problem 2 的 generator
  - 如果 model 太大, 同學可以使用後面的 init.sh 來下載, 請記得讓下載後的路徑是這個路徑

# Submission 2/4

- hw11\_p1.sh
  - 說明: 可以利用同學繳交上來的 checkpoints/p1\_g.pth **重現** Report Problem 1 的圖片
  - 用法: **bash hw11\_p1.sh <checkpoint> <out\_image>**
    - checkpoint: 用來讀取 model 的路徑, 請同學不要寫死
    - out\_image: 生成 image 後的存檔路徑
  - 範例:
    - 助教執行 `bash hw11_p1.sh checkpoints/p1_g.pth ~/image_rep/p1.png`
    - 產生的 `~/image_rep/p1.png` 須要跟同學 report problem 1 的圖片一樣
- hw11\_p2.sh
  - 用法同上, 能 **重現** 同學繳交上來的 report problem 2 的圖片

# Submission 3/4

- train\_p1.sh
  - 說明: 訓練一個品質和 Report Problem 1 圖片差不多的 generator。
  - 用法: **bash train\_p1.sh <face\_dir> <checkpoint>**
    - face\_dir: 助教這邊存放訓練資料 face/ 的路徑, 請同學不要寫死
    - checkpoint: 訓練完 model 之後要存檔的路徑, 請同學不要寫死
  - 範例:
    - 助教執行 `bash train_p1.sh ~/data/face ~/checkpoints/p1_g_rep.pth`
    - 產生的 `~/checkpoints/p1_g_rep.pth` 須要跟同學交上來的 `checkpoints/p1_g.pth` 產生圖片的結果差不多
- train\_p2.sh
  - 用法同上, 能訓練一個品質和 Report Problem 2 圖片差不多的 generator。

# Submission 4/4

- `init.sh`
  - 說明:助教在跑大家的 code 前, 會先跑一次這個 code。如果你的 model 太大, 你可以使用這個 code 來下載你的 model。注意:請將你下載的 model 路徑存成如前幾頁所述 (ex: checkpoints/p1\_g.pth)
  - 用法: **`bash init.sh`**

# Reproduce Regulation

- 請同學確保你上傳的程式所產生的結果，會跟你 report 上的結果一致。基本上，使用你生成 report 圖片的那個 generator 和那組 input vectors 就可以做到這件事。
- 若助教覺得你生成的結果有問題，會重跑同學的 training code。若生成品質和你的 report 差太多，則對應的 report problem 視為無法重現。
- 生成品質由助教判斷，基本上不會為難同學，但是如果真的和 report 差太多還是會請同學說明一下。
- init.sh 執行時間沒有上限，但如果超過 **15 分鐘**，助教可能會私底下找你討論。
- hw11\_p1.sh, hw11\_p2.sh 執行時間上限皆為 **10 分鐘**。
- train\_p1.sh, train\_p2.sh 執行時間上限皆為 **6 小時**。



# Grading Policy

- Report 1.b 和 2.b 的圖片必須要能夠重現, 否則該小題 0 分計算。
- 事後補救依期初公告處理。

# Links

- Colab: <https://reurl.cc/V6a3jQ>
- Report template: <https://reurl.cc/z8ybMk>
- 遲交表單: <https://bit.ly/39d2x2m>