

生成式對抗網路 (Generative Adversarial Network, GAN) (一) – 基本概念介紹

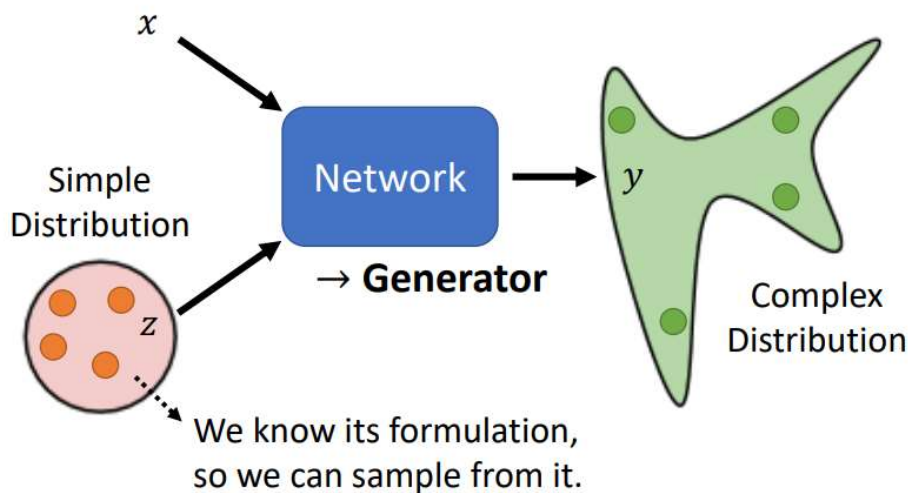
Create at 2022/06/22

- 生成式對抗網路 (Generative Adversarial Network, GAN) (一) – 基本概念介紹
 - Generation
 - Generative Adversarial Network (GAN)
 - Progressive GAN
- 上課資源：
 1. 生成式對抗網路 (Generative Adversarial Network, GAN) (一) – 基本概念介紹
(<https://www.youtube.com/watch?v=4OWp0wDu6Xw>).

Generation

把 **network** 當作一個 **generator** 來用

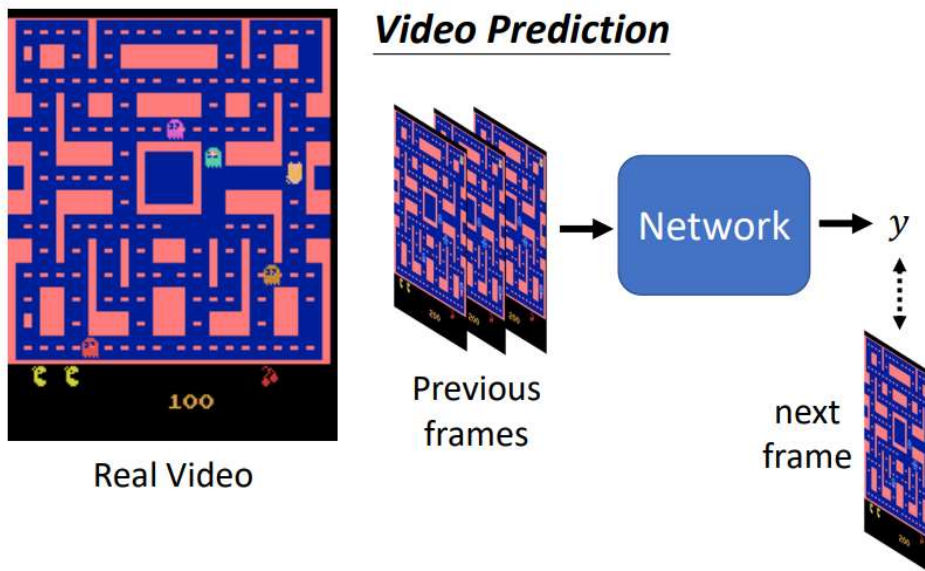
Network as Generator



- network 的輸入會加上一個 random 的 variable z
- z 是從某一個 distribution sample 出來的
 - distribution 的限制是必須夠簡單 (必須知道它的式子長什麼樣)
 - 可以是 function distribution、gaussian distribution、uniform distribution
- 所以現在 network 是同時看 x 、 z 得到輸出 y
- 每次用 network 的時候，會隨機生成一個 z ，所以 z 每次都不一樣
- 所以 network 的輸出不再是單一個固定的東西，而變成了一個複雜的 distribution
- 可以輸出一個 distribution 的 network，稱它為 generator

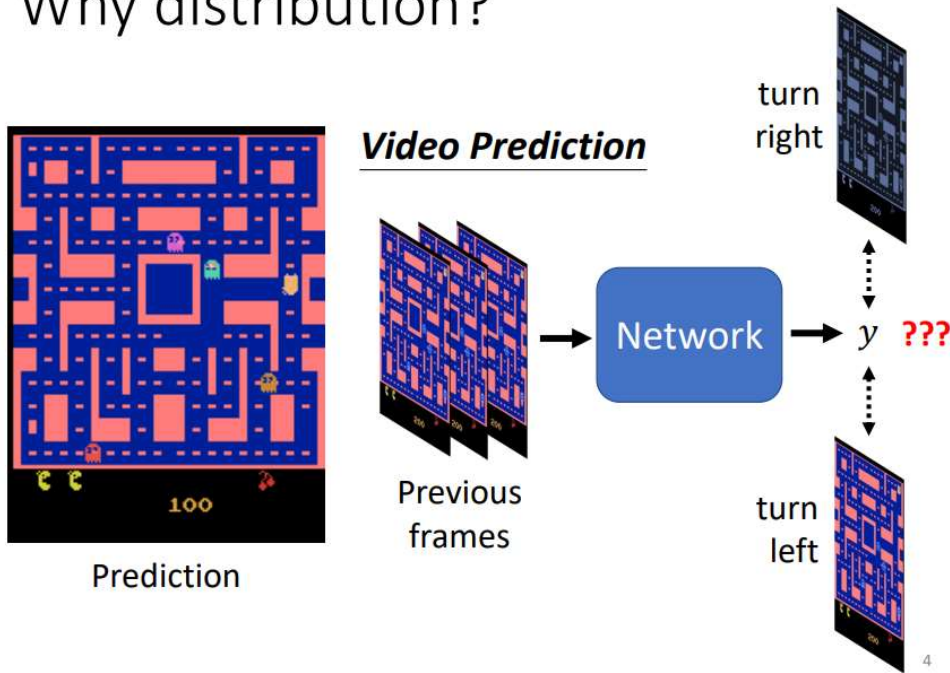
為什麼輸出需要一個分布？

Why distribution?



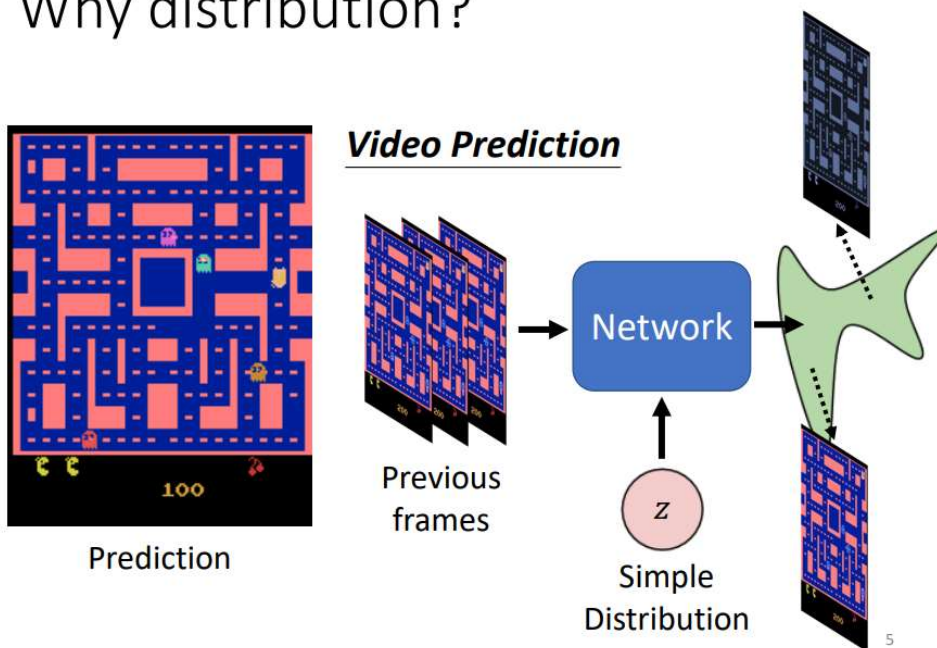
- https://github.com/dyelax/Adversarial_Video_Generation
(https://github.com/dyelax/Adversarial_Video_Generation)
- 給機器一段影片，預測接下來會發生什麼事情
- Video prediction
 - network
 - 輸入：過去的遊戲畫面
 - 輸出：下一秒的遊戲畫面

Why distribution?



- 如果用 supervise learning train 下去
 - 小精靈到轉角可能會分裂、消失
 - 因為對 network 而言，在訓練資料裡面同樣的輸入
 - 有時候同樣的轉角
 - 有時候會左轉、有時候會右轉
- 兩種可能性同時存在訓練資料裡面
 - 機器學到兩面討好

Why distribution?



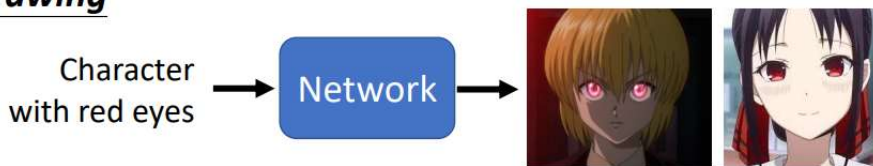
- 怎麼解決這個問題呢？
 - 讓機器的輸出是有機率分佈的，而不是單一的輸出
- 給 network 加上一個 input distribution z 的時候，輸出就變成一個 distribution 不再是固定的
- 希望訓練一個 network 可以知道輸出的分佈包含了向左轉跟向右轉的可能

Why distribution?

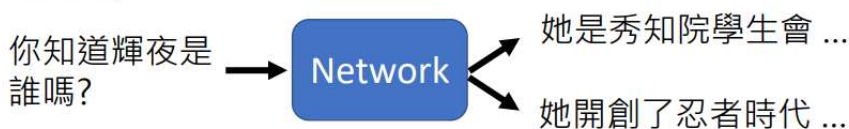
(The same input has different outputs.)

- Especially for the tasks needs “creativity”

Drawing



Chatbot



- 什麼時候會特別需要處理這個問題？
 - 當我們任務需要一點創造力的時候
 - 想要找一個 function，但是同樣的輸入有多種可能的輸出，而這些不同的輸出都是對的
 - 讓機器具有創造的能力

Generative Adversarial Network (GAN)

All Kinds of GAN ...

<https://github.com/hindupuravinash/the-gan-zoo>

GAN
ACGAN
BGAN
CGAN
DCGAN
EBGAN
fGAN
GoGAN
⋮

- SeUDA - Semantic-Aware Generative Adversarial Nets for Unsupervised Domain Adaptation
- SG-GAN - Semantic-aware Grad-GAN for Virtual-to-Real Urban Scene Adaption (github)
- SG-GAN - Sparsely Grouped Multi-task Generative Adversarial Networks for Facial Attribute Manipulation
- SGAN - Texture Synthesis with Spatial Generative Adversarial Networks
- SGAN - Stacked Generative Adversarial Networks (github)
- SGAN - Steganographic Generative Adversarial Networks
- SGAN - SGAN: An Alternative Training of Generative Adversarial Networks
- SGAN - CT Image Enhancement Using Stacked Generative Adversarial Networks and T1 Segmentation Improvement
- sGAN - Generative Adversarial Training for MRA Image Synthesis Using Multi-Contrast
- SiftingGAN - SiftingGAN: Generating and Sifting Labeled Samples to Improve the Remote Sensing Image Classification Baseline in vitro
- SiGAN - SiGAN: Siamese Generative Adversarial Network for Identity-Preserving Face Image Generation
- SimGAN - Learning from Simulated and Unsupervised Images through Adversarial Training
- SisGAN - Semantic Image Synthesis via Adversarial Learning

Mihaela Rosca, Balaji Lakshminarayanan, David Warde-Farley, Shakir Mohamed, "Variational Approaches for Auto-Encoding Generative Adversarial Networks", arXiv, 2017

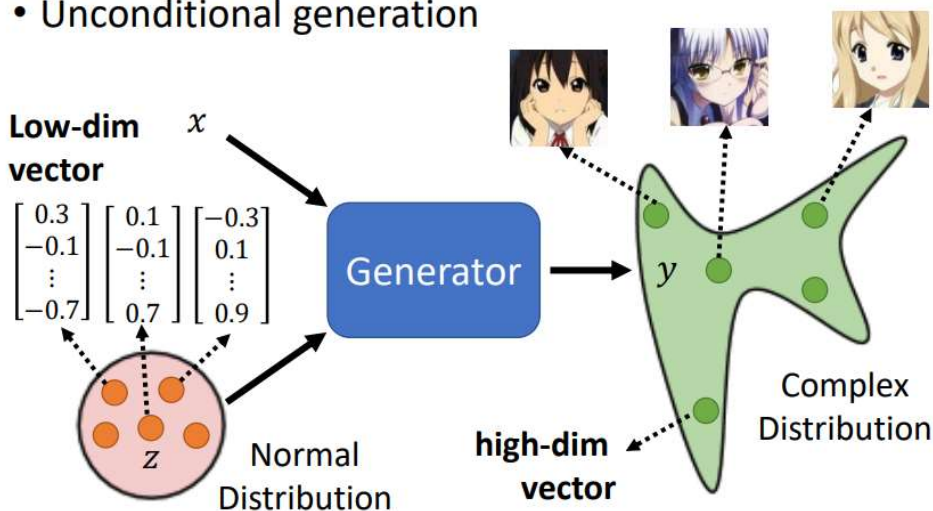
²We use the Greek α prefix for α -GAN, as AEGAN and most other Latin prefixes seem to have been taken
<https://deeephunt.in/the-gan-zoo-79597dc8c347>.

9

GAN 動物園

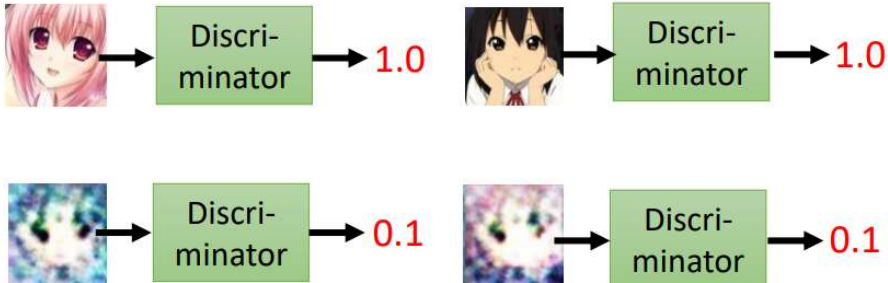
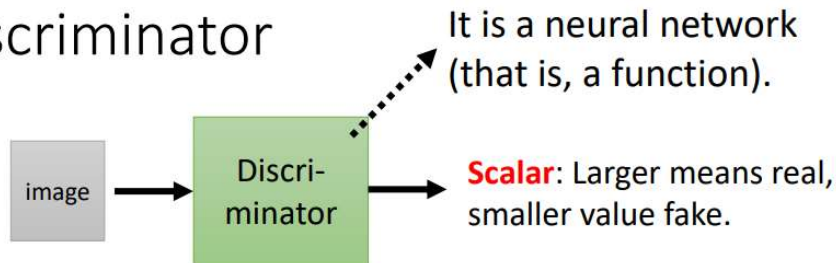
Anime Face Generation

• Unconditional generation



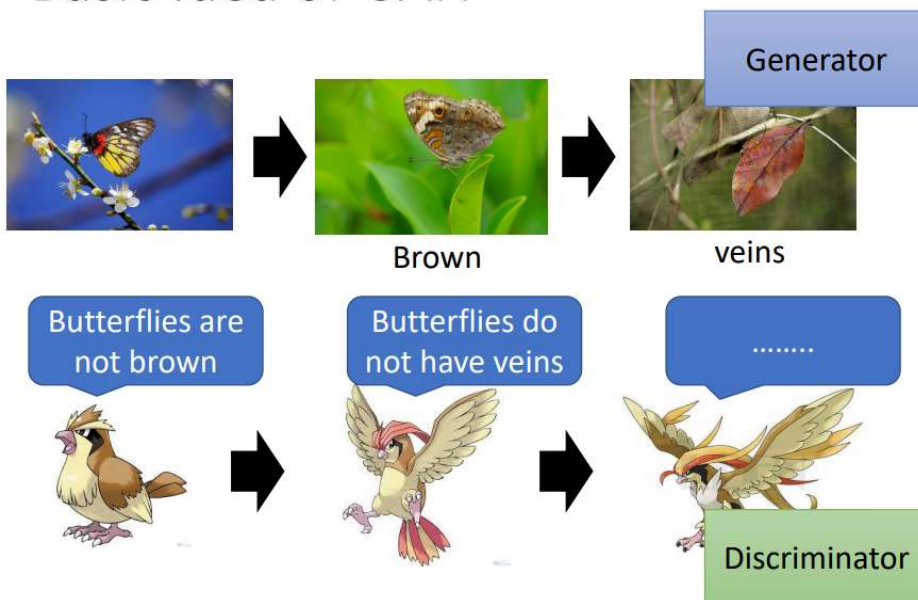
- 讓機器生成二次元人物的臉
- Unconditional generation
 - Generator
 - 輸入： z ，假設是 normal distribution sample 出來的向量，通常是 low-dimension 向量
 - 輸出： y
- 從 normal distribution sample 一個向量丟到 generator，generator 就給一個對應的輸出，一個二次元人物的臉

Discriminator



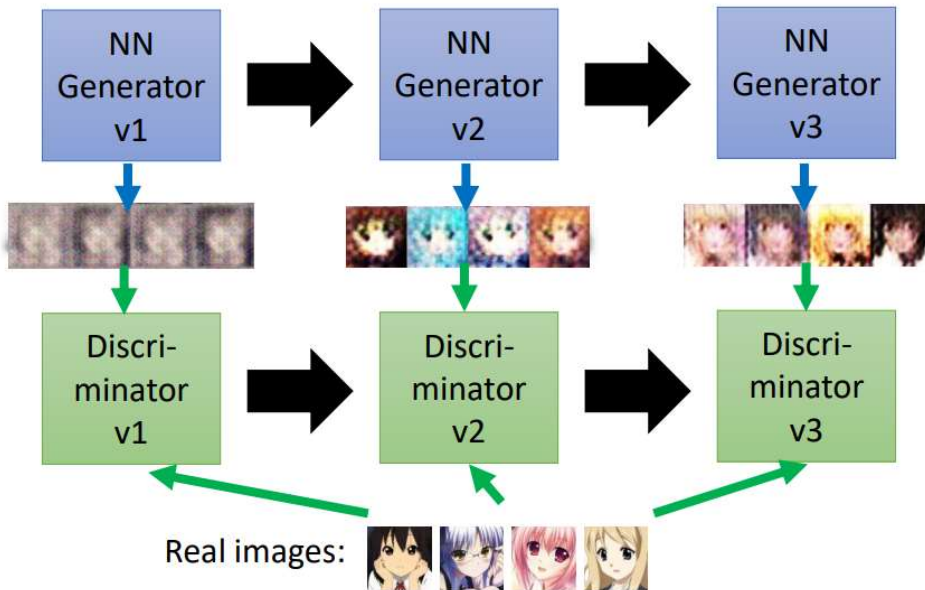
- 在 GAN 裡面，一個特別的地方是，除了 generator 之外要多訓練一個 **discriminator**
- discriminator (本身也是一個 network) 作用：
 - 輸入：一張圖片
 - 輸出：一個數值
 - 數值越大，代表輸出的圖片越像真實的二次元人物圖像

Basic Idea of GAN



Basic Idea of GAN

This is where the term
“*adversarial*” comes from.



- 第一代
 - generator
 - 參數完全是隨機的，所以不知道要怎麼畫二次元人物
 - 畫出來的東西是莫名其妙的雜訊
 - discriminator
 - 學習的目標分辨 generator 的輸出跟真正圖片的不同
 - 有兩個黑黑的眼睛就是二次元人物
 - 沒有眼睛就是 generator 的輸出
- 第二代
 - generator
 - 進化，調整裡面的參數，調整的目標是為要騙過 discriminator
 - 產生眼睛，騙過第一代的 discriminator
 - discriminator
 - 分辨 generator 的輸出與真實二次元人物圖片之間的差異
 - 有沒有嘴巴
- 第三代
 - generator
 - 想辦法去騙過第二代的 discriminator
 - 加上嘴巴
 - discriminator
 - 會越來越嚴苛，可以讓 generator 產生出的圖片越來越像二次元人物
- **Adversarial**：對抗

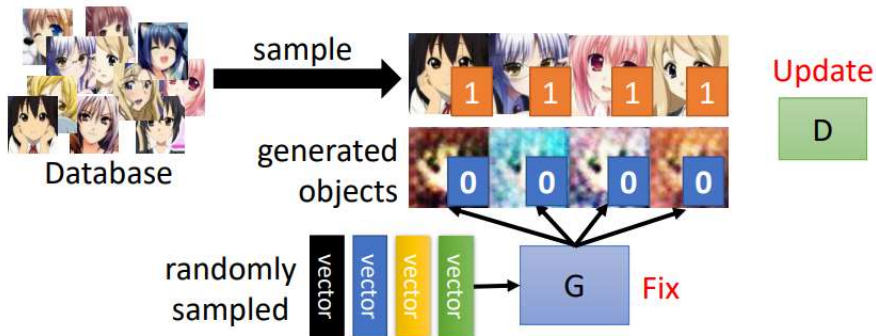
從演算法的角度介紹 **generator** 與 **discriminator** 是如何運作的

Algorithm

- Initialize generator and discriminator
- In each training iteration:



Step 1: Fix generator G, and update discriminator D



Discriminator learns to assign high scores to real objects and low scores to generated objects.

15

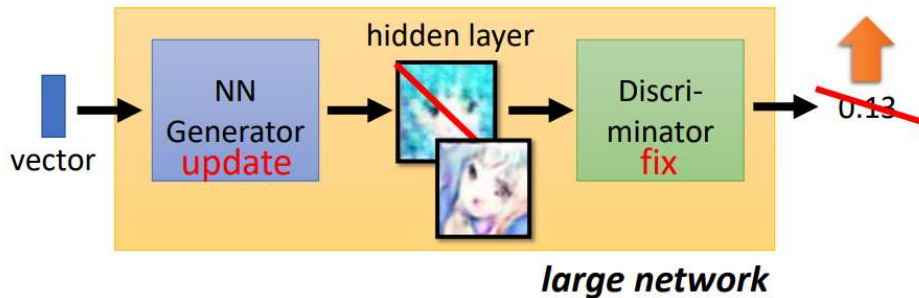
- generator 跟 discriminator 是兩個 network
- network 在訓練之前要先初始化它的參數
- 訓練
 - Step 1 : 訓練 discriminator
 - 定住 generator，只 train discriminator
 - 因為一開始 generator 參數是隨機初始化的，如果又固定住 generator，等於 generator 甚麼事都沒做
 - 所以丟一堆向量給 generator 輸出都是亂七八糟的圖片
 - 從圖庫去 sample 一些二次元人物的頭像
 - 接著拿真正的二次元人物頭像跟 generator 產生出來的結果去訓練 discriminator
 - discriminator 訓練的目標是要分辨真正的二次元人物跟 generator 產生出來的二次元人物之間的差異
- 實際的操作
 - 把真正的人物設 1
 - generator 產生的人物設 0
 - 對 discriminator 來說就是一個分類的問題或是 regression 的問題

Algorithm

- Initialize generator and discriminator G D
- In each training iteration:

Step 2: Fix discriminator D, and update generator G

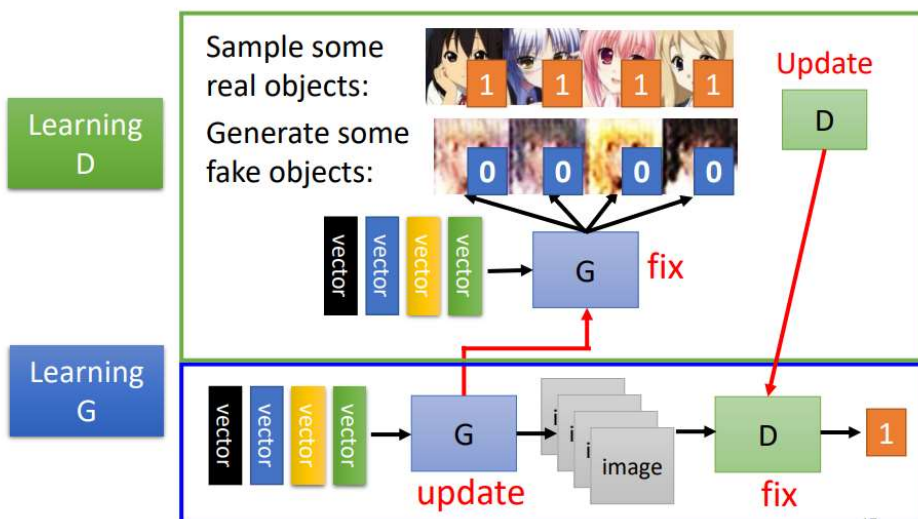
Generator learns to “fool” the discriminator



- Step 2 : 訓練 generator
 - 訓練完 discriminator 之後，定住 discriminator 改成訓練 generator
 - 讓 generator 想辦法去騙過 discriminator
- 實際的操作
 - generator 輸入一個從 gaussian distribution sample 出來的向量，產生一個圖片
 - 接著把圖片丟到 discriminator 裡面，discriminator 會給這個圖片一個分數
 - generator 訓練的目標，是讓 discriminator 的輸出值越大越好 (因為 discriminator 看到好的圖片就給大的分數)

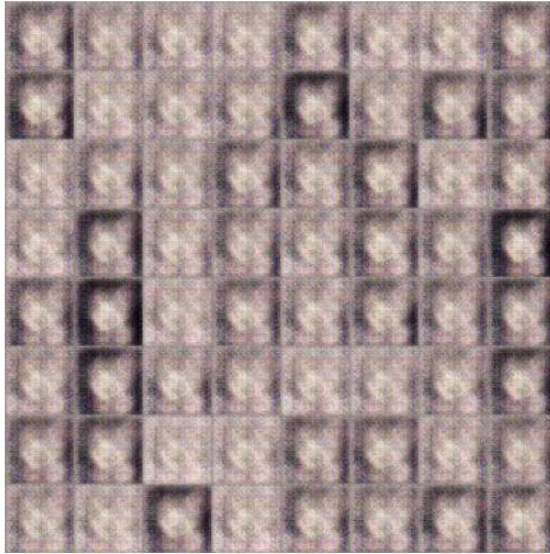
Algorithm

- Initialize generator and discriminator G D
- In each training iteration:



- 接著就是反覆的訓練 discriminator 跟 generator
 - 訓練完 discriminator
 - 固定住 discriminator，訓練 generator
 - 訓練完 generator
 - 用 generator 產生更多的新的圖片，再給 discriminator 做訓練
 - 訓練完 discriminator，再去訓練 generator

Anime Face Generation



100 updates

Source of training data: <https://zhuanlan.zhihu.com/p/24767059>

- <https://zhuanlan.zhihu.com/p/24767059> (<https://zhuanlan.zhihu.com/p/24767059>)
- 訓練 generator、discriminator 反覆 100 次

Anime Face Generation



1000 updates

- 訓練 generator、discriminator 反覆 1000 次，產生眼睛

Anime Face Generation



2000 updates

- 訓練 generator、discriminator 反覆 2000 次，產生嘴巴

Anime Face Generation



5000 updates

- 訓練 generator、discriminator 反覆 5000 次，開始有人臉的樣子

Anime Face Generation



10,000 updates

- 訓練 generator、discriminator 反覆 10000 次

Anime Face Generation



20,000 updates

- 訓練 generator、discriminator 反覆 20000 次

Anime Face Generation



50,000 updates

- 訓練 generator、discriminator 反覆 50000 次

In 2019, with StyleGAN



Source of video:

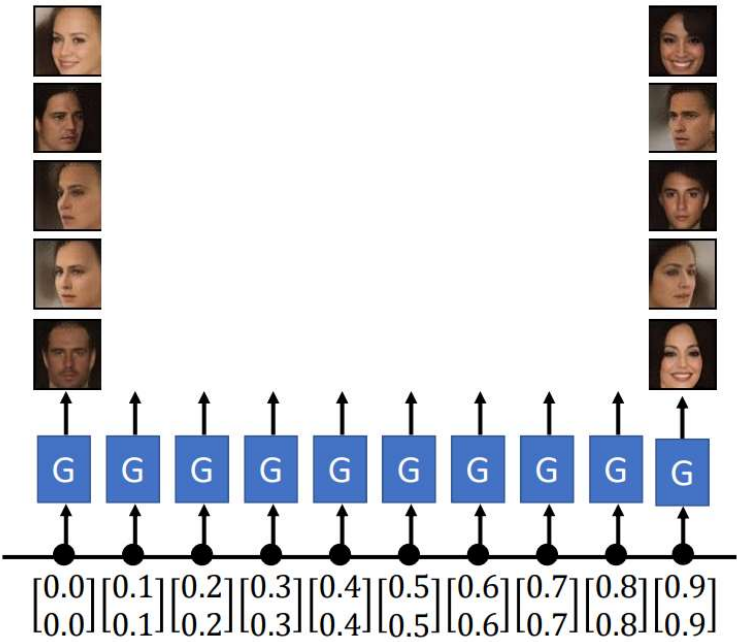
<https://www.gwern.net/Faces>

- 用 StyleGAN 做的

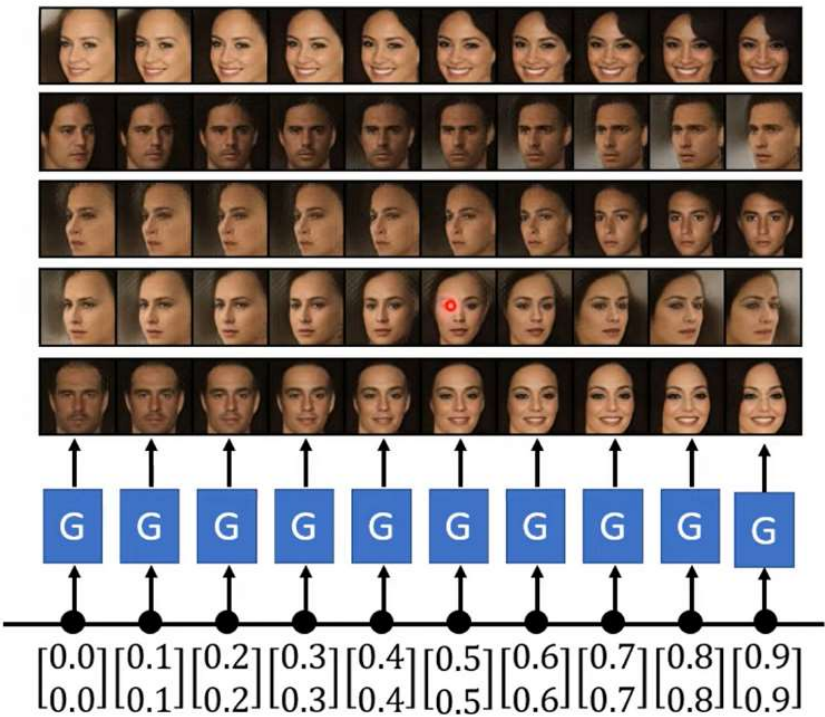
Progressive GAN



- 可以產生真實的人臉
 - 輸入一個向量
 - 輸出一個真實的圖片
- 還可以把輸入的向量做 interpolation
 - 就可以看到兩張圖片之間連續的變化



28



28

- 把向左看跟向右看的做內插，會逐漸向中間看



The first GAN

<https://arxiv.org/abs/1406.2661> (Ian J. Goodfellow)

29



Today BigGAN

<https://arxiv.org/abs/1809.11096>

30



tags: 2022 李宏毅_機器學習