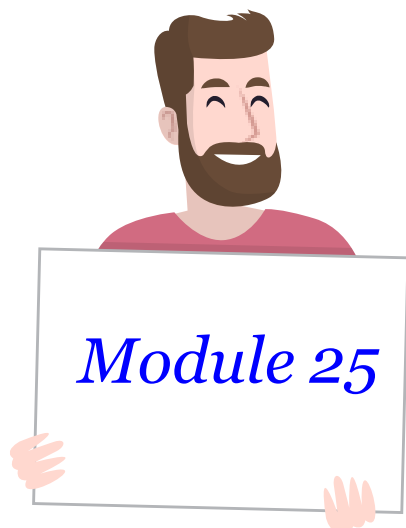




# GAN神經網路介紹



designed by  freepik

Estimated time:  
**45** min.



資訊工業策進會 Institute for Information Industry

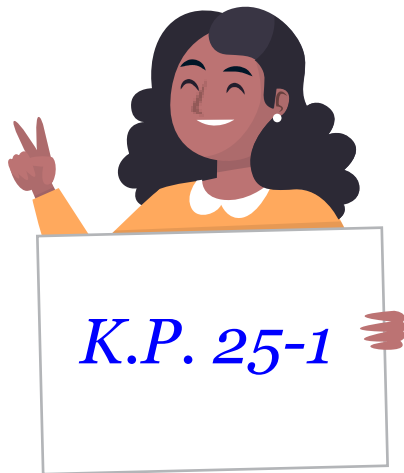
# 學習目標

- 25-1:GAN神經網路介紹
- 25-2:GAN的概念
- 25-3:GAN演算法



# 25-1: GAN 神經網路介紹

- GAN 神經網路介紹
- GAN 熱門程度
- GAN 網路神經網路的組成



designed by freepik

# GAN 神經網路介紹

- **GAN(Generative adversarial networks)**是一種類神經網路
  - 非監督式學習
  - 包含兩個神經網路，這兩個網路會互相對抗
  - 由Ian Goodfellow於2014年所提出



# GAN熱門程度

- **FB的AI研究院院長Yann Lecun說GAN是近20年來深度學習最酷的想法**



The coolest idea in deep learning in the last 20 years

# GAN熱門程度

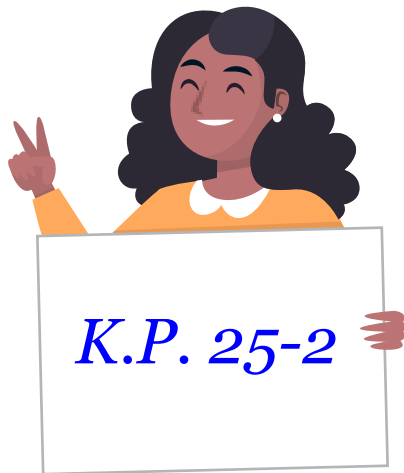
- 自從GAN被提出以來，每年的論文數量持續以驚人的數度增加
  - GAN神經網路的變形也非常多
  - <https://github.com/hindupuravinash/the-gan-zoo>

# GAN網路神經網路的組成

- **GAN神經網路包含兩個部分**
  - **Generator跟Discriminator**
- **Generator network**
  - 目標是試著想要生成假照片來騙過**Discriminator**
- **Discriminator network**
  - 目標是想要辨識到底是真的照片還是**Generator**產生的假照片

## 25-2: GAN的概念

- **Generator與Discriminator**
- **GAN演化的概念**
- **GAN整體架構**

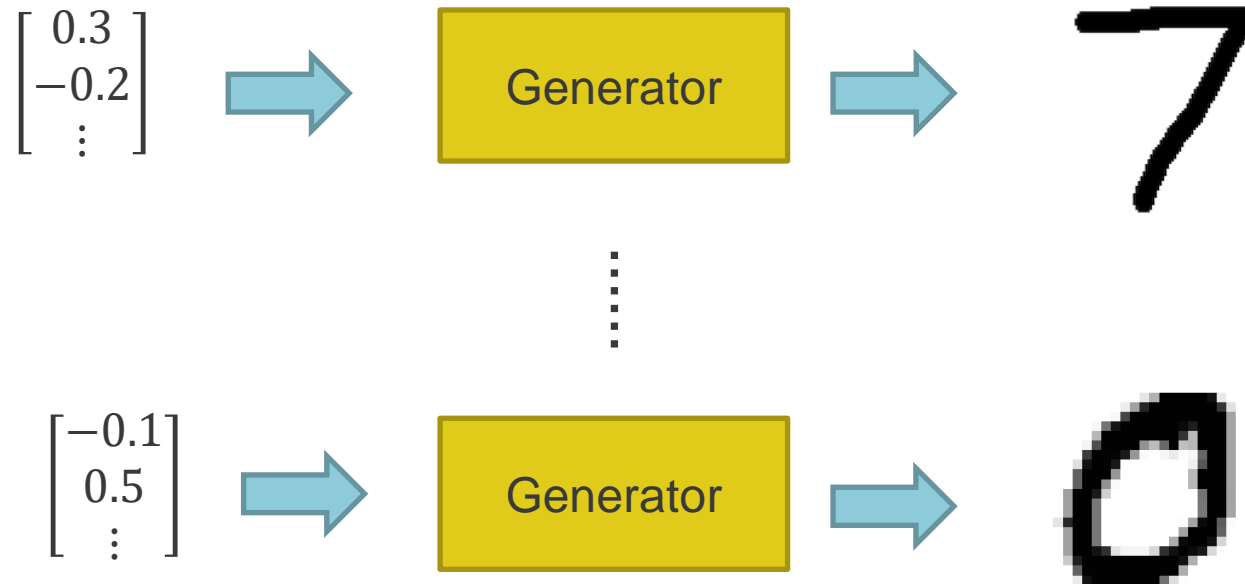


designed by freepik



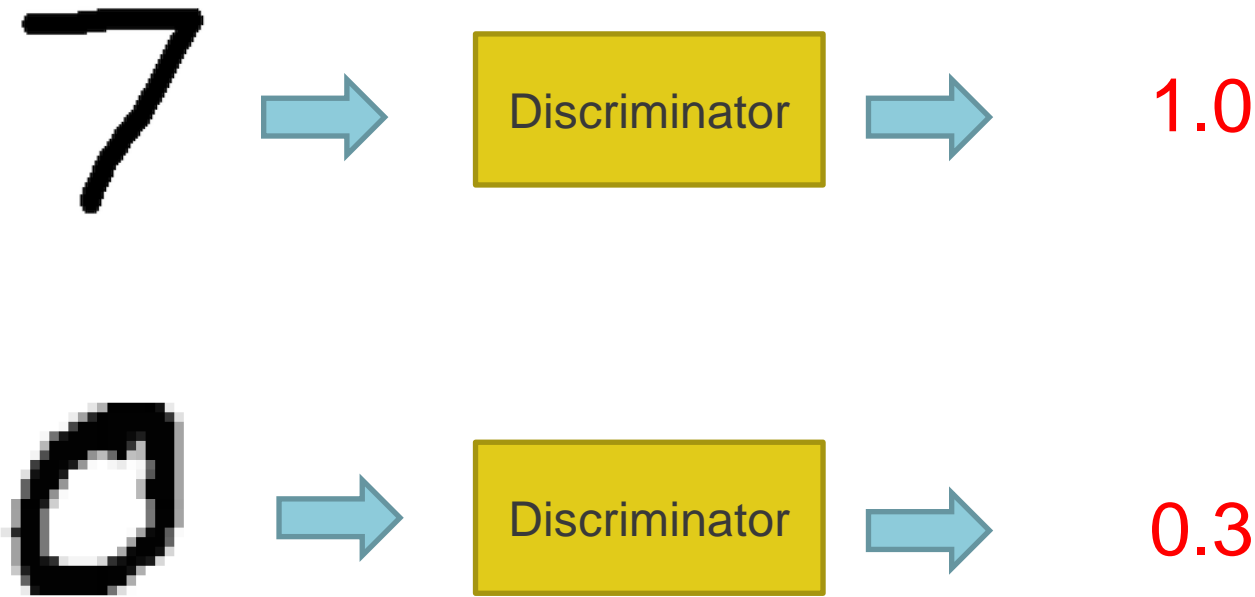
# Generator與Discriminator

- **Generator**的概念是輸入一個隨機的向量，其可以產生一張照片
  - 有時候產生的照片很逼真，有時候很假



# Generator與Discriminator

- 輸入一張照片，**Discriminator**會判別此照片是否為真的照片
  - 輸出數字越接近1代表照片越接近真實，越接近0代表照片越假



# GAN演化的概念

- 假設我們一開始有一個**Generator**以及**Discriminator**
  - 一開始**Discriminator**無法很好的辨識照片是真的還是假的



# GAN演化的概念

- **Discriminator**會開始演化
  - 於是它開始可以辨識出**Generator**產生出來的假照片



# GAN演化的概念

- 為了騙過Discriminator，Generator會開始演化
  - 於是Generator產生出來的假照片開始又可以騙過Discriminator



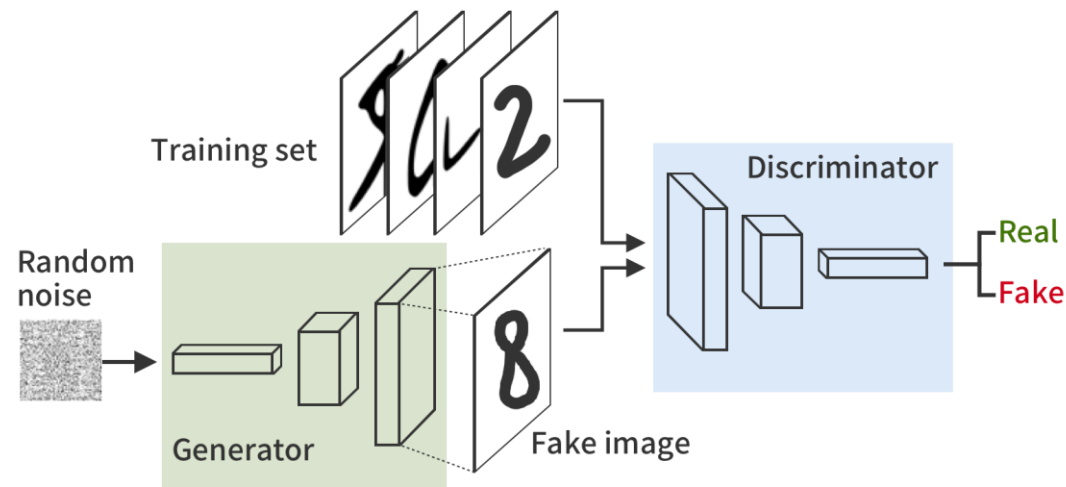
# GAN演化的概念

- 以此類推，**Generator**及**Discriminator**不斷演化下去
  - 我們最終得到一個可以以假亂真的照片**Generator**
  - 還有一個很會辨識真假照片的**Discriminator**



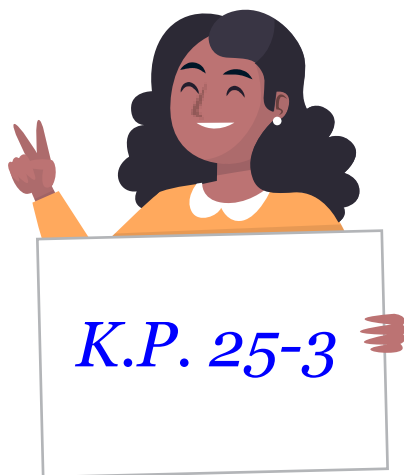
# GAN 整體架構

- GAN的整體架構如下
  - Generator的輸入是noise，輸出是一個以假亂真的照片
  - Discriminator可以辨識出哪些是真照片，哪些是假照片



# 25-3: GAN 演算法

- **GAN 演算法**
- **固定 Generator，更新 Discriminator**
- **更新 Generator，固定 Discriminator**
- **修正 Generator 損失函數**



designed by freepik



# GAN演算法

- **GAN整體演算法如下**
- **初始化Generator及其參數 $\theta_g$ 且初始化Discriminator及其參數 $\theta_d$**
- **在每次訓練迭代當中:**
  - **固定Generator並更新Discriminator參數多次**
  - **固定Discriminator並更新Generator一次**

# 固定 Generator ， 更新 Discriminator

- 隨機從資料分布  $P_{data}(x)$  去從資料集  $\{x^1, \dots, x^m\}$  取樣  $m$  筆資料
- 從常態分佈 **noise**  $\{z^1, \dots, z^m\}$  裡面去取樣  $m$  筆資料
- 計算 **Generator** 產生的資料  $\{\tilde{x}^1, \dots, \tilde{x}^m\}$  ， 這些資料是從  $\tilde{x}^i = G(z^i)$  來的
- 更新 **Discriminator** 參數  $\theta_d$  並極大化以下式子
  - $\tilde{V} = \frac{1}{m} \sum_{i=1}^m \log(D(x^i)) + \frac{1}{m} \sum_{i=1}^m \log(1 - D(\tilde{x}^i))$
  - 可以使用梯度上降法  $\theta_d \leftarrow \theta_d + \eta \nabla \tilde{V}(\theta_d)$
- 重複以上步驟多次

# 更新 Generator ， 固定 Discriminator

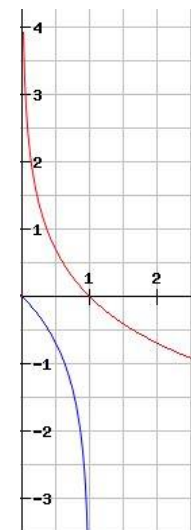
- 從常態分佈noise  $\{z^1, \dots, z^m\}$  裡面去取樣m筆資料
- 更新Generator參數 $\theta_g$ 並極小化以下式子
  - $\tilde{V} = \frac{1}{m} \sum_{i=1}^m \log(D(x^i)) + \frac{1}{m} \sum_{i=1}^m \log(1 - D(G(z^i)))$
  - 可以使用梯度下降法， $\theta_g \leftarrow \theta_g - \eta \nabla \tilde{V}(\theta_g)$

# 修正 Generator 損失函數

- 關於更新 Generator，可以發現  $\theta_g$  只與後面  $\log(1-D(G(z)))$  那一項有關
  - 但是  $\log(1-D(G(z)))$  這個函數在 0 附近的時候微分太平緩了
  - 所以當初設計 GAN 的作者找了另外一個函數來取代它，這樣可以增加訓練速度
  - 把原本藍色的式子變成紅色的式子(如右圖)

$$\tilde{V} = \frac{1}{m} \sum_{i=1}^m \log(D(x^i)) + \frac{1}{m} \sum_{i=1}^m \log(1 - D(G(z^i)))$$

紅色:  $-\log(x)$   
藍色:  $\log(1-x)$

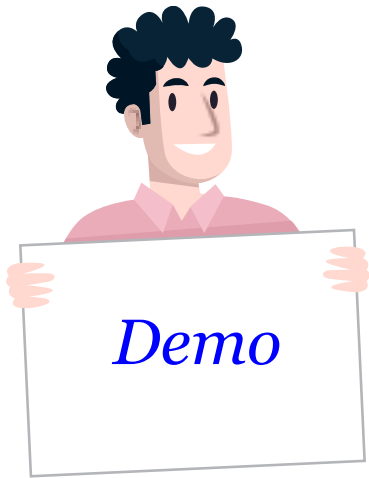


# 修正 Generator 損失函數

- 因此，更新 **Generator**，固定 **Discriminator** 的演算法變成如下
- 從常態分佈 **noise**  $\{z^1, \dots, z^m\}$  裡面去取樣 **m** 筆資料
- 更新 **Generator** 參數  $\theta_g$  並極小化以下式子
  - $\tilde{V} = \frac{1}{m} \sum_{i=1}^m -\log(D(G(z^i)))$
  - 可以使用梯度下降法， $\theta_g \leftarrow \theta_g - \eta \nabla \tilde{V}(\theta_g)$

# Demo 25-3

- 建立Generator
- 建立Discriminator
- GAN神經網路建構



designed by freepik

# 線上Corelab

- 題目1：實作GAN Generator
  - 給予MNIST資料，將Generator程式碼部分完成讓GAN可以產生MNIST照片
- 題目2：實作GAN Discriminator
  - 給予MNIST資料，將Discriminator程式碼部分完成讓GAN可以產生MNIST照片
- 題目3：實作GAN在MNIST資料集上
  - 給予MNIST資料，完成GAN所有程式碼

# 本章重點精華回顧

- **GAN神經網路介紹**
- **GAN的概念**
- **GAN演算法**





# Lab: GAN 神經網路

- **Lab01: 建立 Generator**
- **Lab02: 建立 Discriminator**
- **Lab03: GAN 神經網路建構**

Estimated time:

**20** minutes

