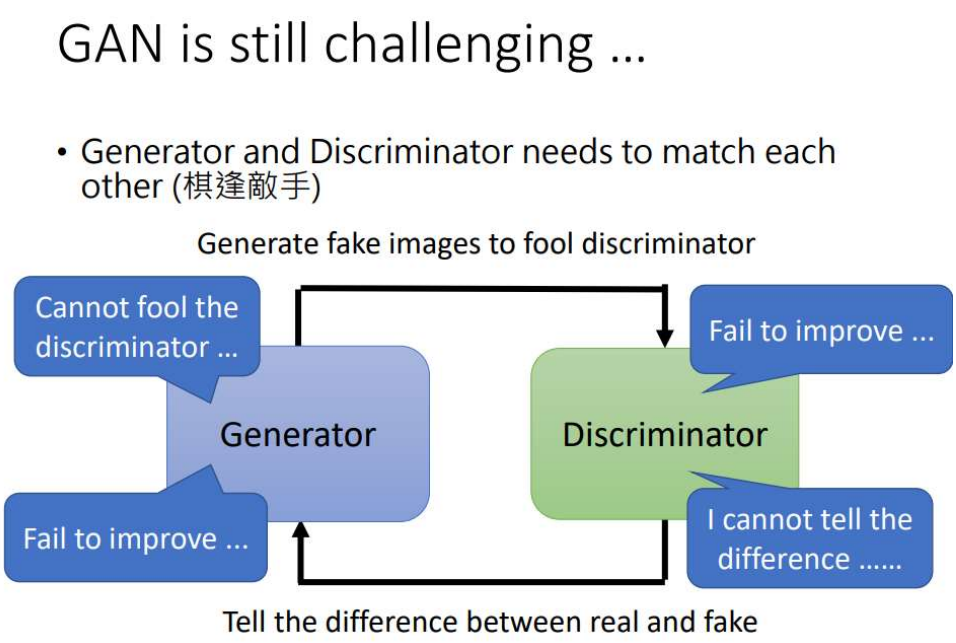
生成式對抗網路 (Generative Adversarial Network, GAN) (三) – 生成器效能評估與條件式生成

Create at 2022/06/22

- 上課資源：
  1. 生成式對抗網路 (Generative Adversarial Network, GAN) (三) – 生成器效能評估與條件式生成 (https://www.youtube.com/watch?v=MP0BnVH2yOo)
- 延伸資料：
  1. Unsupervised Learning - Deep Generative Model (Part II) (https://www.youtube.com/watch?v=8zomhgKrsmQ)
  2. Flow-based Generative Model (https://www.youtube.com/watch?v=uXY18nzdSsM)

為什麼 GAN 很難 train ?



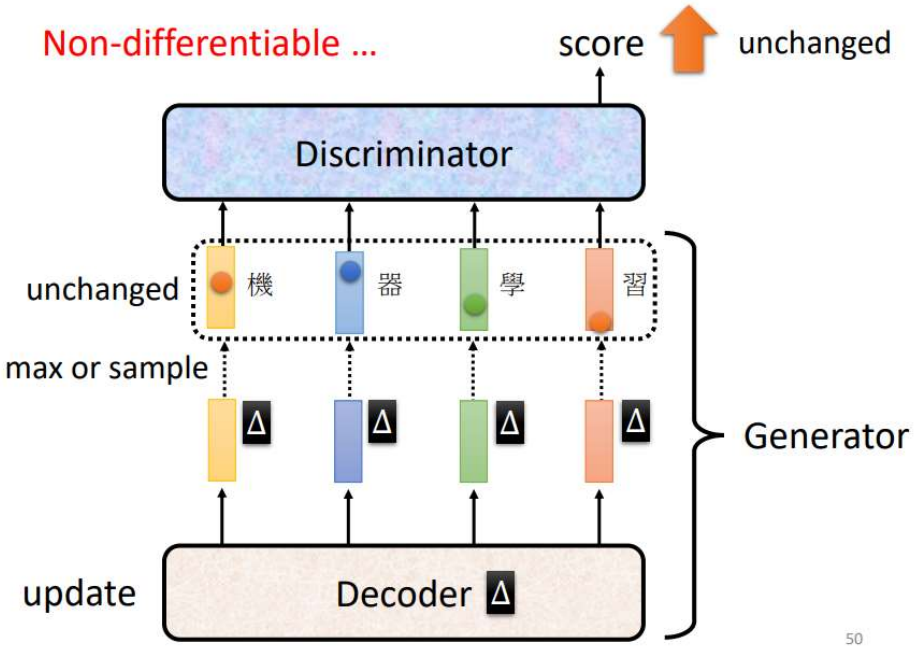
- 雖然有 WGAN，但 GAN 仍然是以很難把它 train 起來聞名
- 為什麼很難被 train 起來？
  - Discriminator 目標：分辨真的圖片跟產生出來的圖片差異
  - Generator 目標：產生假的圖片騙過 discriminator
  - 其中一者不再進步，另外一個就會跟著停下來
  - 互相砥礪才能成長得更好

More Tips

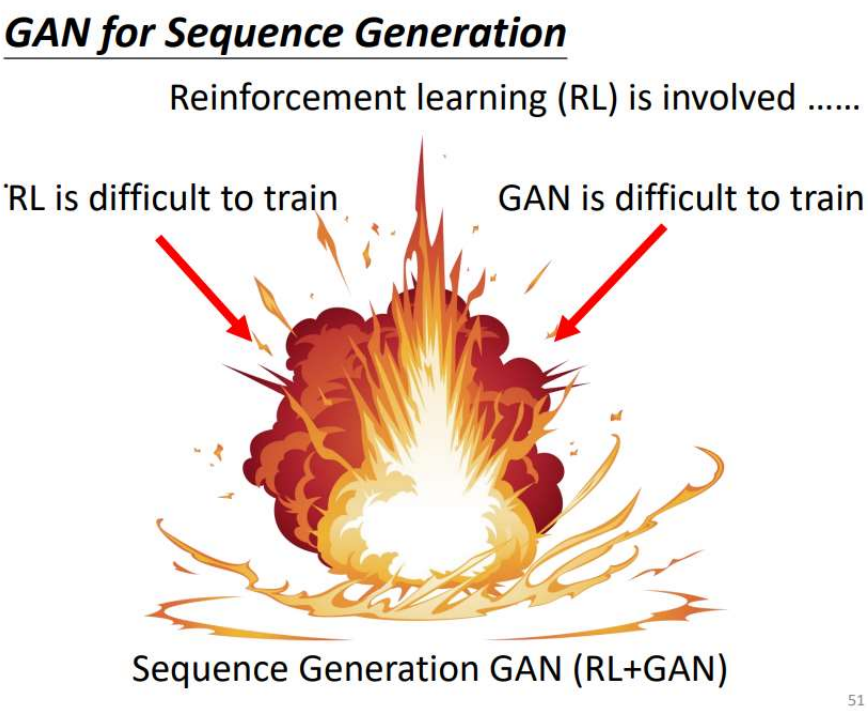
- Tips from Soumith
  - https://github.com/soumith/ganhacks
- Tips in DCGAN: Guideline for network architecture design for image generation
  - https://arxiv.org/abs/1511.06434
- Improved techniques for training GANs
  - https://arxiv.org/abs/1606.03498
- Tips from BigGAN
  - https://arxiv.org/abs/1809.11096

- train GAN 的訣竅

GAN for Sequence Generation



- train GAN 最難的是要拿 GAN 來生成文字
- 為什麼最困難？
  - 生成一段文字會有一個 Sequence To Sequence 的 model 就是 Generator，裡面有一個 Decoder 會產生一段文字
  - 難點在於，如果要用 gradient descent 去 train Decoder 讓 discriminator output 分數越大越好，會發現做不到
    - 為什麼做不到？
      - 當 Decoder 的參數有一點變化的時候，Discriminator 輸出是沒有改變的，所以根本沒辦法做 gradient descent 算微分

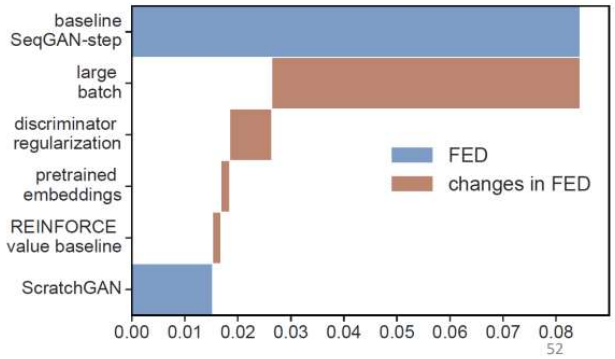


- GAN 以很難 train 起來聞名
- RL 也以很難 train 起來聞名

GAN for Sequence Generation

- Usually, the generator are fine-tuned from a model learned by other approaches.
- However, with enough hyperparameter-tuning and tips, ScarchGAN can train from scratch.

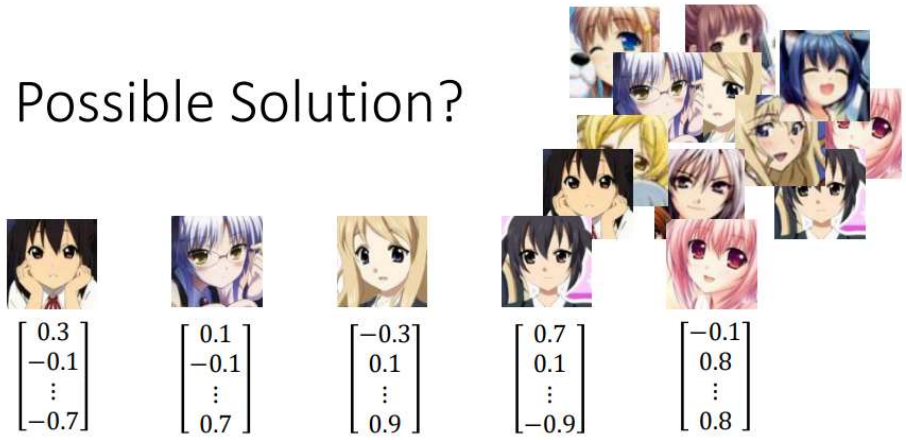
Training language GANs from Scratch
https://arxiv.org/abs/1905.09922



52

- 沒有人能用 GAN 的方式訓練一個 Generator 產生文字
- 通常需要先做 pretrain
- 直到 ScrachGAN

Possible Solution?



Generative Latent Optimization (GLO), https://arxiv.org/abs/1707.05776
Gradient Origin Networks, https://arxiv.org/abs/2007.02798

55

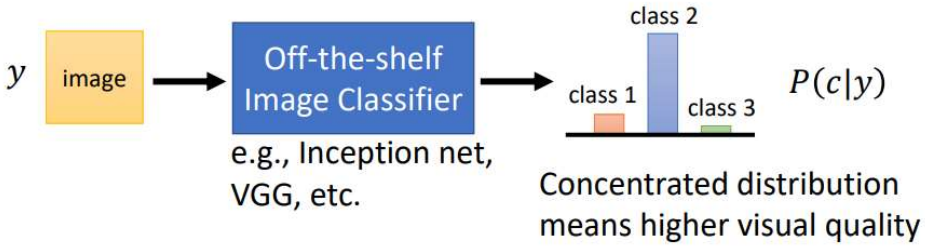
- 每個圖片弄一個 vector 來代表
- 需要特殊的方法去安排那些 vector，否則訓練的結果會很差

GAN 的評估

怎麼看我們現在產生出來的 Generator 好不好？

Quality of Image

- Human evaluation is expensive (and sometimes unfair/unstable).
- How to evaluate the quality of the generated images automatically?



- 人眼判斷，會有很多的問題，例如不客觀、不穩定
- 看有沒有比較客觀的方法去衡量一個 Generator 的好壞
  - 針對一些任務，可以設計一些方法
- 方法：
  - 跑一個影像分類系統，把 GAN 產生出來的圖片丟到影像分類系統裡面
  - 輸入：一張圖片 y
  - 輸出：機率分佈 $P(c|y)$
    - 如果機率分佈越集中，代表產生的圖片可能越好
    - 如果平均分佈，代表 GAN 產生出來的圖片四不像

Diversity - Mode Collapse

86

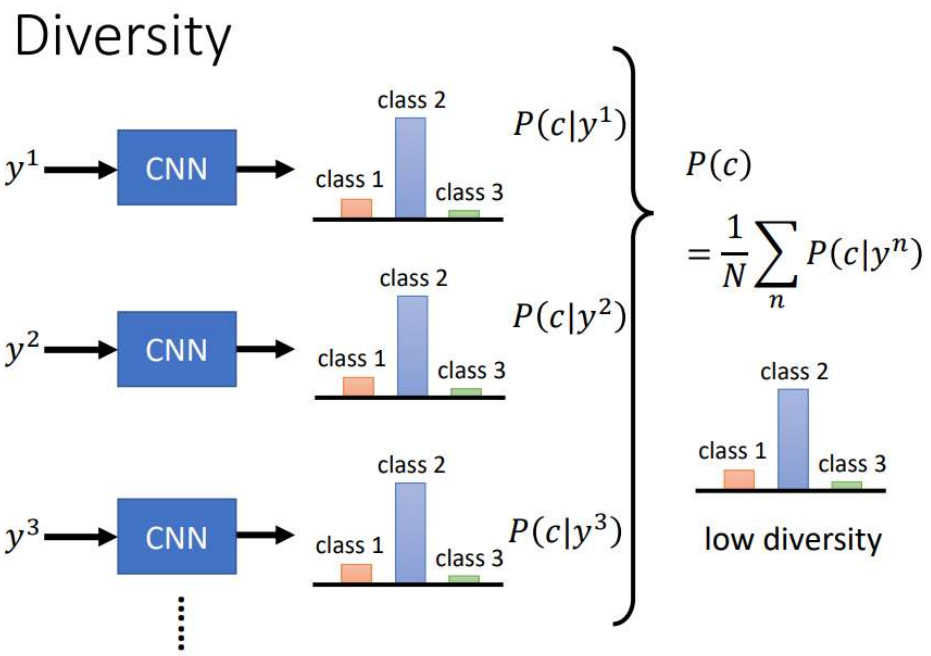
- 這個辨識系統會被一個 Mode Collapse 的問題騙過去
- Mode Collapse :
  - 在 train GAN 最後發現，產生的都是同一張臉
  - 當 Generator 學會產生這種圖片，就永遠都可以騙過 Discriminator，而 Discriminator 無法看出這樣的圖片是假的，是 Discriminator 的盲點

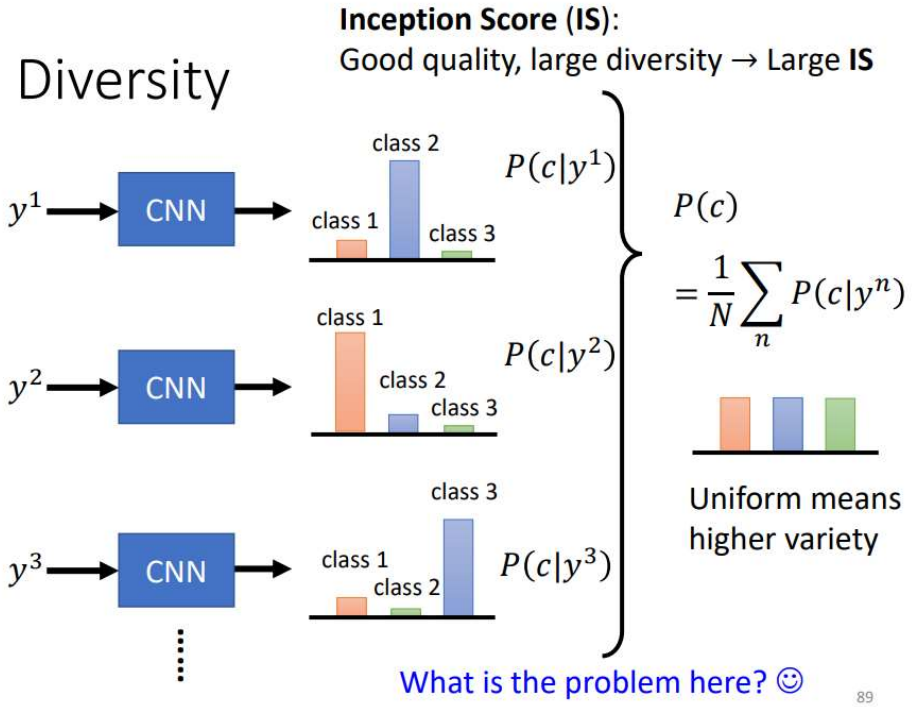
Diversity - Mode Dropping

(BEGAN on CelebA)

87

- 跟 Mode Collapse 有點像，但更難被偵測到的問題叫 Mode Dropping
- Mode Dropping
  - Generator 產生出來的資料只有真實資料的一部分
  - 單純看產生出來的資料，可能會覺得還不錯，因為分佈跟多樣性都夠
  - 但是不知道真實資料的多樣性其實是更大的

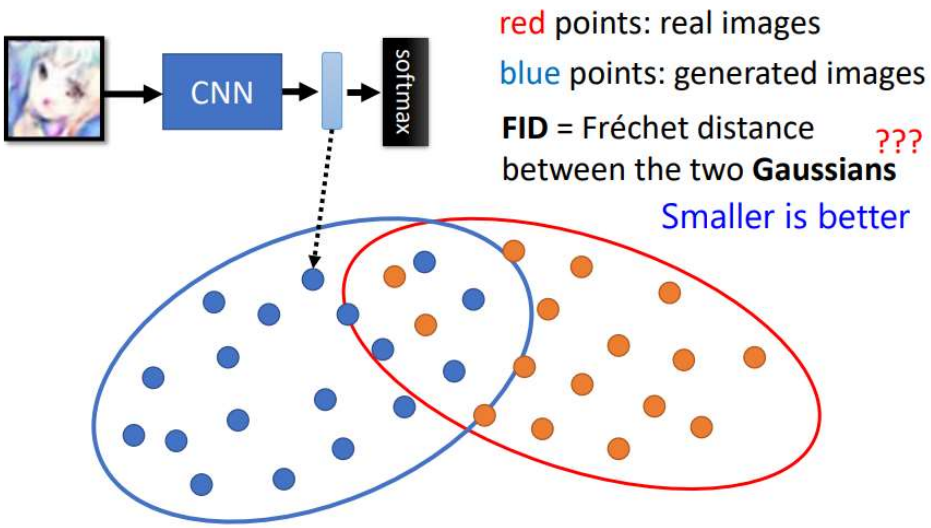




- 需要去測量現在的 Generator 多樣性夠不夠
  - 把你的 Generator 產生的 1000 張圖片，都丟到 image classify 裡面，看它被判斷成哪一個 class
  - 每張圖片都有一個 Distribution
  - 把所有的 Distribution 平均起來，看平均的 Distribution 長怎樣
    - 如果 Distribution 很集中，就代表現在多樣性不夠
    - 如果 Distribution 分佈很平均，就代表現在多樣性是足夠的
- **Inception Score (IS)**

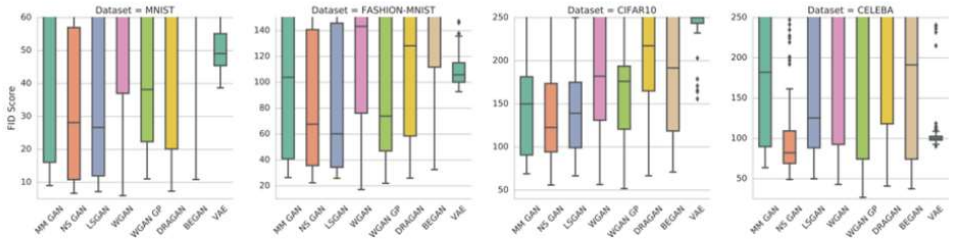
https://arxiv.org/pdf/1706.08500.pdf

Fréchet Inception Distance (FID)



- 作業採取另外一個 Evaluation 的 measure
- Frechet Inception Distance (FID)
  - 把產生出來的二次元人物丟到 Inception Network 裡面，讓 Inception Network 輸出它的類別，得到的可能就是人臉
  - 我們要的是進入 Softmax 之前的 Hidden Layer 的輸出 (進入 Softmax 之前會產生一個向量代表這張圖片)
  - 假設真實的圖片跟產生出來的圖片，它們都是 Gaussians Distribution，然後去計算這兩個 Gaussians Distribution 之間的 Frechet Distance
  - 越小越好，越小代表產生的圖片跟真實的圖片越接近

GAN	Discriminator Loss	Generator Loss
MM GAN	$\mathcal{L}_{\mathrm{D}}^{\mathrm{GAN}} = -\mathbb{E}_{x\sim p_d}[\log(D(x))] + \mathbb{E}_{\hat{x}\sim p_g}[\log(1 - D(\hat{x}))]$	$\mathcal{L}_{\mathrm{G}}^{\mathrm{GAN}} = -\mathcal{L}_{\mathrm{D}}^{\mathrm{GAN}}$
NS GAN	$\mathcal{L}_{\mathrm{D}}^{\mathrm{NSGAN}} = \mathcal{L}_{\mathrm{D}}^{\mathrm{GAN}}$	$\mathcal{L}_{\mathrm{G}}^{\mathrm{NSGAN}} = \mathbb{E}_{\hat{x}\sim p_g}[\log(D(\hat{x}))]$
WGAN	$\mathcal{L}_{\mathrm{D}}^{\mathrm{WGAN}} = -\mathbb{E}_{x\sim p_d}[D(x)] + \mathbb{E}_{\hat{x}\sim p_g}[D(\hat{x})]$	$\mathcal{L}_{\mathrm{G}}^{\mathrm{WGAN}} - = \mathcal{L}_{\mathrm{D}}^{\mathrm{WGAN}}$
WGAN GP	$\mathcal{L}_{\mathrm{D}}^{\mathrm{WGAN}} = \mathcal{L}_{\mathrm{D}}^{\mathrm{WGAN}} + \lambda\mathbb{E}_{\hat{x}\sim p_g}[(\|\nabla D(\alpha x + (1-\alpha\hat{x})\|_2 - 1)^2]$	$\mathcal{L}_{\mathrm{G}}^{\mathrm{WGAN}} = -\mathbb{E}_{\hat{x}\sim p_g}[D(\hat{x})]$
LS GAN	$\mathcal{L}_{\mathrm{D}}^{\mathrm{LSGAN}} = -\mathbb{E}_{x\sim p_d}[(D(x) - 1)^2] + \mathbb{E}_{\hat{x}\sim p_g}[D(\hat{x})^2]$	$\mathcal{L}_{\mathrm{G}}^{\mathrm{LSGAN}} = -\mathbb{E}_{\hat{x}\sim p_g}[(D(\hat{x} - 1)^2]$
DRAGAN	$\mathcal{L}_{\mathrm{D}}^{\mathrm{DRAGAN}} = \mathcal{L}_{\mathrm{D}}^{\mathrm{GAN}} + \lambda\mathbb{E}_{\hat{x}\sim p_d + \mathcal{N}(0,c)}[(\|\nabla D(\hat{x})\|_2 - 1)^2]$	$\mathcal{L}_{\mathrm{G}}^{\mathrm{DRAGAN}} = -\mathcal{L}_{\mathrm{D}}^{\mathrm{NS\ GAN}}$
BEGAN	$\mathcal{L}_{\mathrm{D}}^{\mathrm{BEGAN}} = \mathbb{E}_{x\sim p_d}[\|\|x - \mathrm{AE}(x)\|\|_1] - k_t\mathbb{E}_{\hat{x}\sim p_g}[\|\|\hat{x} - \mathrm{AE}(\hat{x})\|\|_1]$	$\mathcal{L}_{\mathrm{G}}^{\mathrm{BEGAN}} = \mathbb{E}_{\hat{x}\sim p_g}[\|\|\hat{x} - \mathrm{AE}(\hat{x})\|\|_1]$



FIT: Smaller is better

Are GANs Created Equal? A Large-Scale Study
https://arxiv.org/abs/1711.10337

91

https://arxiv.org/pdf/1511.01844.pdf

We don't want memory GAN.

Real Data

Generated Data

Same as real data ...

Generated Data

Simply flip real data ...

- 問題：
  - 訓練的 Generator 產生出來的 data 跟真實資料一模一樣

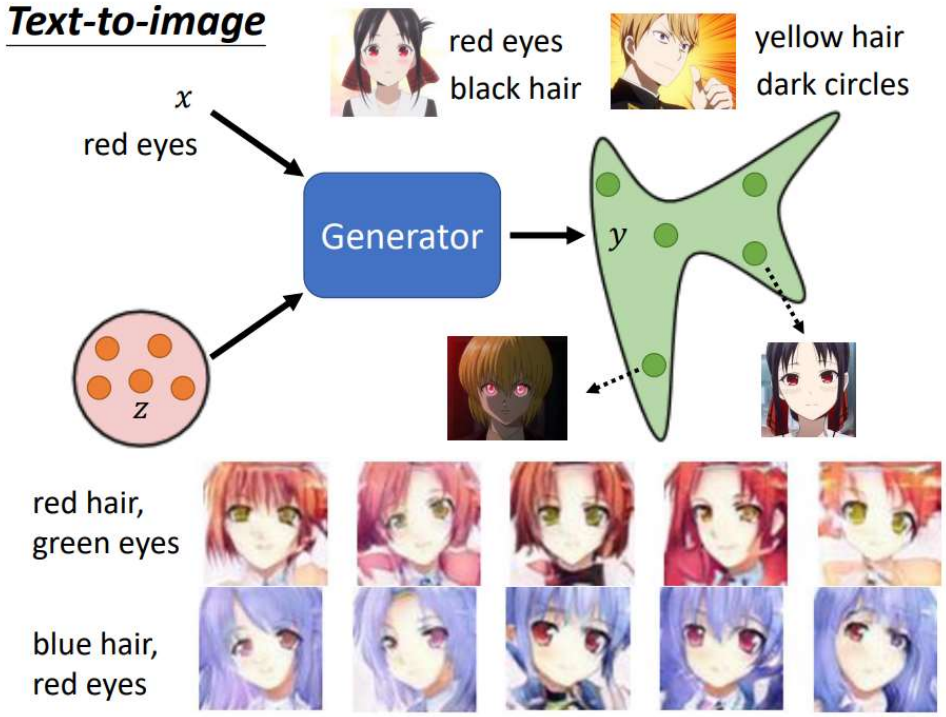
To learn more about evaluation ...

	Measure	Description
Quantitative	1. Average Log-likelihood [18, 22]	• Log likelihood of explaining realworld held out/test data using a density estimated from the generated data (*e.g.* using KDE or Parzen window estimation). $L = \frac{1}{N}\sum_i \log P_{model}(\mathbf{x}_i)$
	2. Coverage Metric [13]	• The probability mass of the true data "covered" by the model distribution $C = P_{data}(dP_{model} > t)$ with t such that $P_{model}(dP_{model} > t) = 0.95$
	3. Inception Score (IS) [3]	• KLD between conditional and marginal label distributions over generated data. $\exp(\mathbb{E}_{\mathbf{x}}[\mathbb{KL}(p(\mathbf{y}\mid\mathbf{x}) \parallel p(\mathbf{y}))])$
	4. Modified Inception Score (m-IS) [34]	• Encourages diversity within images sampled from a particular category. $\exp(\mathbb{E}_{\mathbf{x}_1}[\mathbb{E}_{\mathbf{x}_2}[\mathbb{KL}(P(\mathbf{y}\mid\mathbf{x}_1) \parallel P(\mathbf{y}\mid\mathbf{x}_2))]])$
	5. Mode Score (MS) [35]	• Similar to IS but also takes into account the prior distribution of the labels over real data. $\exp(\mathbb{E}_{\mathbf{x}}[\mathbb{KL}(p(\mathbf{y}\mid\mathbf{x}) \parallel p(\mathbf{y}^{train})]) - \mathbb{KL}(p(\mathbf{y}) \parallel p(\mathbf{y}^{train}))])$
	6. AM Score [36]	• Takes into account the KLD between distributions of training labels vs. predicted labels, as well as the entropy of predictions. $\mathbb{KL}(p(\mathbf{y}^{train}) \parallel p(\mathbf{y})) + \mathbb{E}_{\mathbf{x}}[H(p(\mathbf{x}))]$
	7. Fréchet Inception Distance (FID) [37]	• Wasserstein-2 distance between multi-variate Gaussians fitted to data embedded into a feature space $FID(r,g) = \|\mu_r - \mu_g\|_2^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{\frac{1}{2}})$
	8. Maximum Mean Discrepancy (MMD) [38]	• Measures the dissimilarity between two probability distributions P_r and P_g using samples drawn independently from each distribution. $M_k(P_r, P_g) = \mathbb{E}_{\mathbf{x},\mathbf{x}'\sim P_r}[k(\mathbf{x},\mathbf{x}')] - 2\mathbb{E}_{\mathbf{x}\sim P_r,\mathbf{y}\sim P_g}[k(\mathbf{x},\mathbf{y})] + \mathbb{E}_{\mathbf{y},\mathbf{y}'\sim P_g}[k(\mathbf{y},\mathbf{y}')]$
	9. The Wasserstein Critic [39]	• The critic (*e.g.* an NN) is trained to produce high values at real samples and low values at generated samples $\hat{W}(\mathbf{x}_{test}, \mathbf{x}_g) = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}_{test}[i]) - \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}_g[i])$
	10. Birthday Paradox Test [27]	• Measures the support size of a discrete (continuous) distribution by counting the duplicates (near duplicates)
	11. Classifier Two Sample Test (C2ST) [40]	• Answers whether two samples are drawn from the same distribution (*e.g.* by training a binary classifier)
	12. Classification Performance [1, 15]	• An indirect technique for evaluating the quality of unsupervised representations (*e.g.* feature extraction; FCN score). See also the GAN Quality Index (GQI) [41].
	13. Boundary Distortion [42]	• Measures diversity of generated samples and covariate shift using classification methods.
	14. Number of Statistically-Different Bins (NDB) [43]	• Given two sets of samples from the same distribution, the number of samples that fall into a given bin should be the same up to sampling noise
	15. Image Retrieval Performance [44]	• Measures the distributions of distances to the nearest neighbors of some query images (*i.e.* diversity)
	16. Generative Adversarial Metric (GAM) [31]	• Compares two GANs by having them engaged in a battle against each other by swapping discriminators or generators. $p(\mathbf{x}\mid y=1; M_1)/p(\mathbf{x}\mid y=1; M_2) = (p(y=1\mid\mathbf{x}; D_1)p(\mathbf{x}; G_2))/(p(y=1\mid\mathbf{x}; D_2)p(\mathbf{x}; G_1))$
	17. Tournament Win Rate and Skill Rating [45]	• Implements a tournament in which a player is either a discriminator that attempts to distinguish between real and fake data or a generator that attempts to fool the discriminators into accepting fake data as real.
	18. Normalized Relative Discriminative Score (NRDS) [32]	• Compares n GANs based on the idea that if the generated samples are closer to real ones, more epochs would be needed to distinguish them from real samples.
	19. Adversarial Accuracy and Divergence [46]	• Adversarial Accuracy. Computes the classification accuracies achieved by the two classifiers, one trained on real data and another on generated data, on a labeled validation set to approximate $P_g(y\mid\mathbf{x})$ and $P_r(y\mid\mathbf{x})$. Adversarial Divergence: Computes $\mathbb{KL}(P_g(y\mid\mathbf{x}), P_r(y\mid\mathbf{x}))$
	20. Geometry Score [47]	• Compares geometrical properties of the underlying data manifold between real and generated data.
	21. Reconstruction Error [48]	• Measures the reconstruction error (*e.g.* L_2 norm) between a test image and its closest generated image by optimizing for z (*i.e.* $min_{\mathbf{z}}\|G(\mathbf{z}) - \mathbf{x}^{(test)}\|^2$)
	22. Image Quality Measures [49, 50, 51]	• Evaluates the quality of generated images using measures such as SSIM, PSNR, and sharpness difference
	23. Low-level Image Statistics [52, 53]	• Evaluates how similar low-level statistics of generated images are to those of natural scenes in terms of mean power spectrum, distribution of random filter responses, contrast distribution, etc.
	24. Precision, Recall and F_1 score [23]	• These measures are used to quantify the degree of overfitting in GANs, often over toy datasets.
Qualitative	1. Nearest Neighbors	• To detect overfitting, generated samples are shown next to their nearest neighbors in the training set
	2. Rapid Scene Categorization [18]	• In these experiments, participants are asked to distinguish generated samples from real images in a short presentation time (*e.g.* 100 ms); *i.e.* real v.s fake
	3. Preference Judgment [54, 55, 56, 57]	• Participants are asked to rank models in terms of the fidelity of their generated images (*e.g.* pairs, triples)
	4. Mode Drop and Collapse [58, 59]	• Over datasets with known modes (*e.g.* a GMM or a labeled dataset), modes are computed as by measuring the distances of generated data to mode centers
	5. Network Internals [1, 60, 61, 62, 63, 64]	• Regards exploring and illustrating the internal representation and dynamics of models (*e.g.* space continuity) as well as visualizing learned features

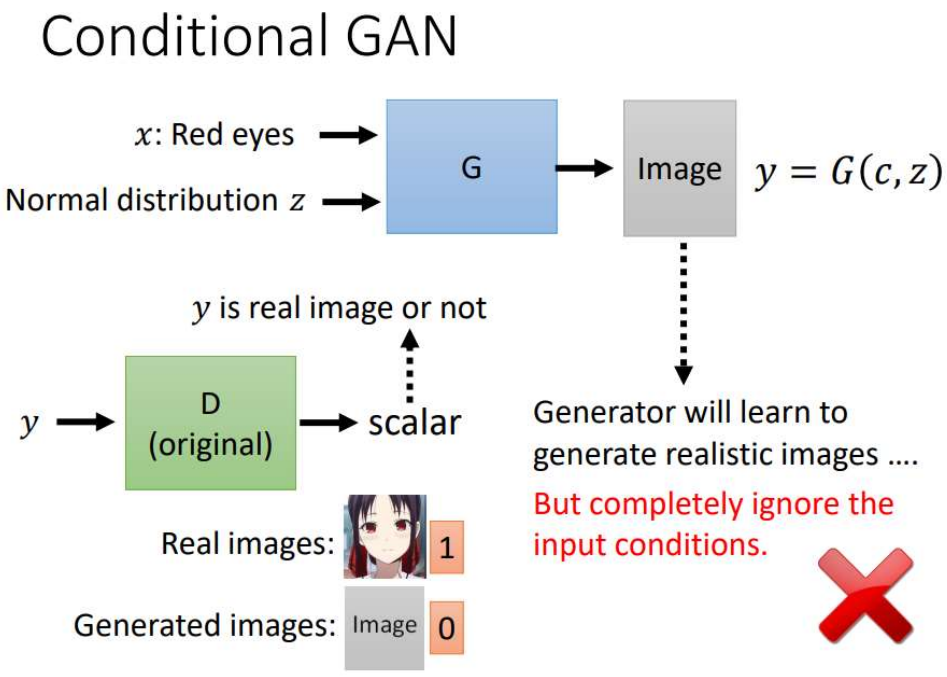
Pros and cons of GAN evaluation measures
https://arxiv.org/abs/1802.03446 93

- 評估 Generation 做的好不好的問題也是一個可以研究的問題

Conditional Generation

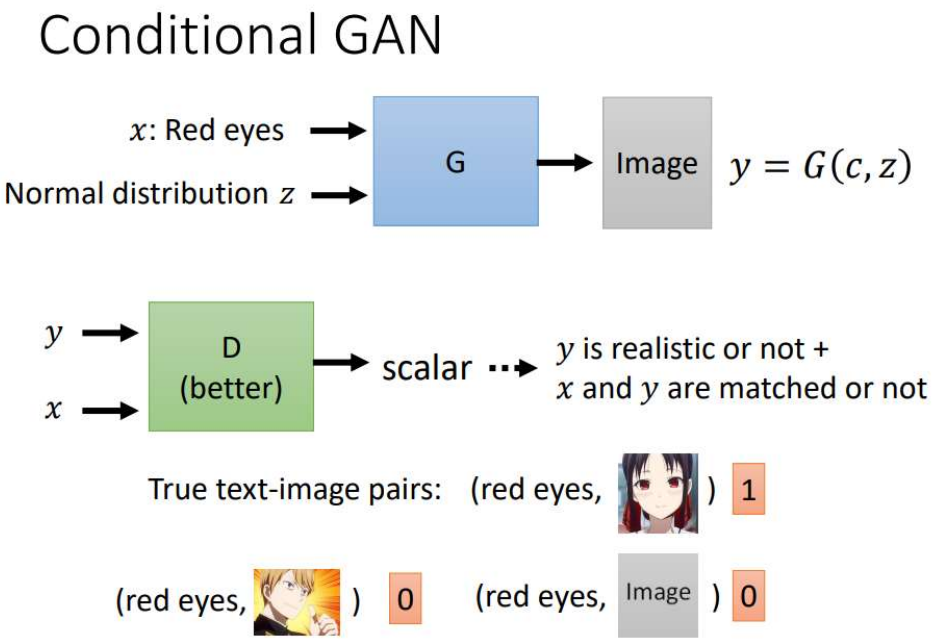


- 希望可以操控 Generation 的輸出
- 給一個 condition x，根據 x、z 來產生 y
- 應用：
  - 可以做文字對圖片的生成，是一個 supervised learning 的問題，需要一些 label 的 data



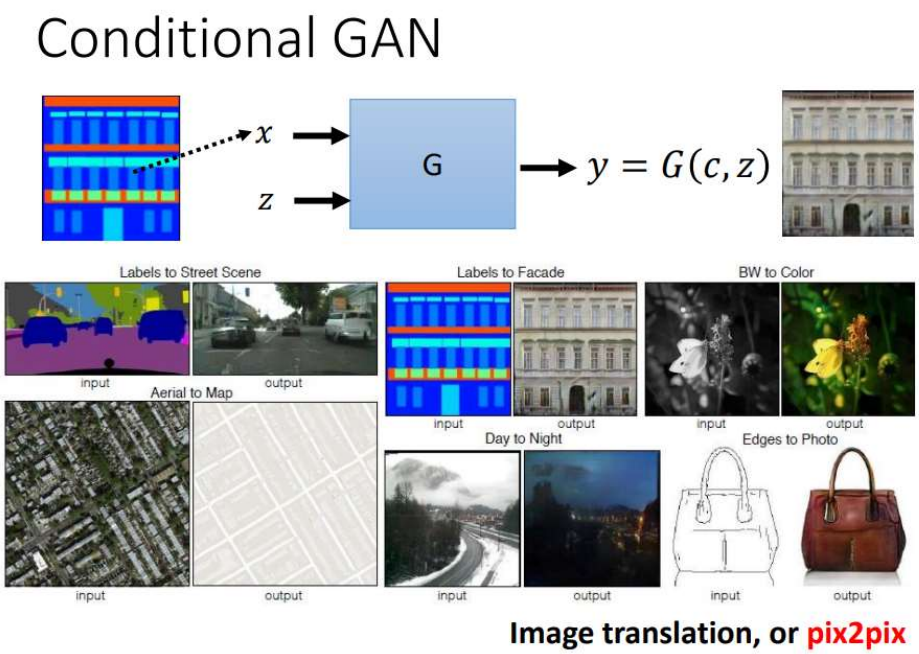
- 如何做 Conditional GAN
  - 現在的 Generator 有兩個輸入
    - 從 normal distribution sample 出來的 z
    - 一段文字 x
  - 輸出一張圖片 y
- 需要一個 discrimination，輸入一張圖片，輸出一個數值 (代表輸入的圖片多像真實的圖片)
- 這個方法沒辦法真的解 Conditional GAN 的問題
  - 因為如果我們只有 train 這個 discriminator，而 discriminator 只會看 y 當作輸入的話，Generator 會學到的是會產生可以騙過 discriminator 清晰的圖片，但是跟輸入完全沒有關係 (可能跟輸入的文字完全無關，因為目的是騙過 discriminator)

https://arxiv.org/abs/1605.05396



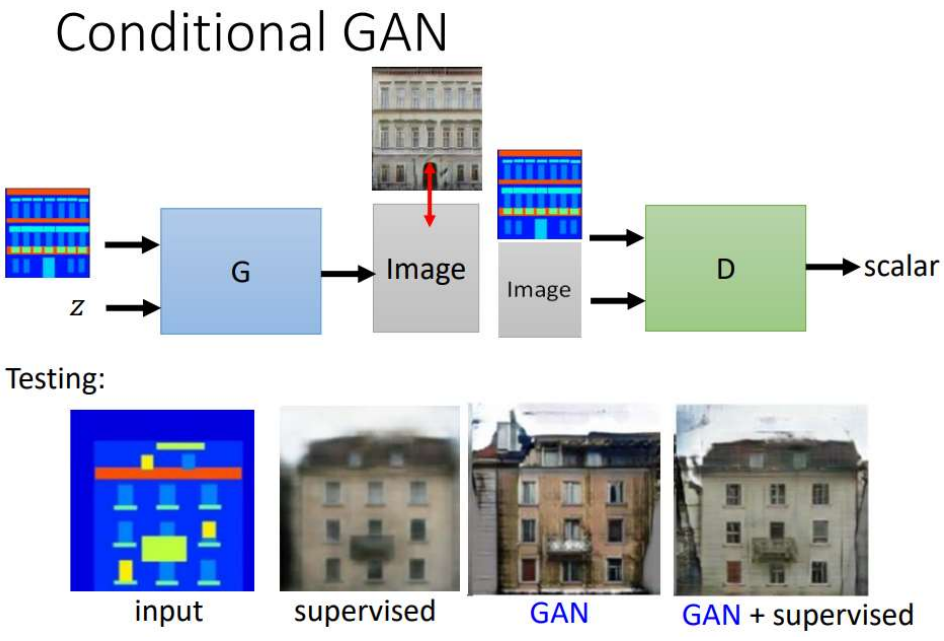
- 所以在 Conditional GAN 要做有點不一樣的設計
- Discriminator 不是只輸入圖片 y，還要輸入 Condition x，然後產生一個數值，數值不只看 y 好不好，也看圖片 y 跟文字的敘述 x 必須搭配得起來，discriminator 才會給高分
- Conditional GAN 的訓練需要 Pair data

https://arxiv.org/abs/1611.07004



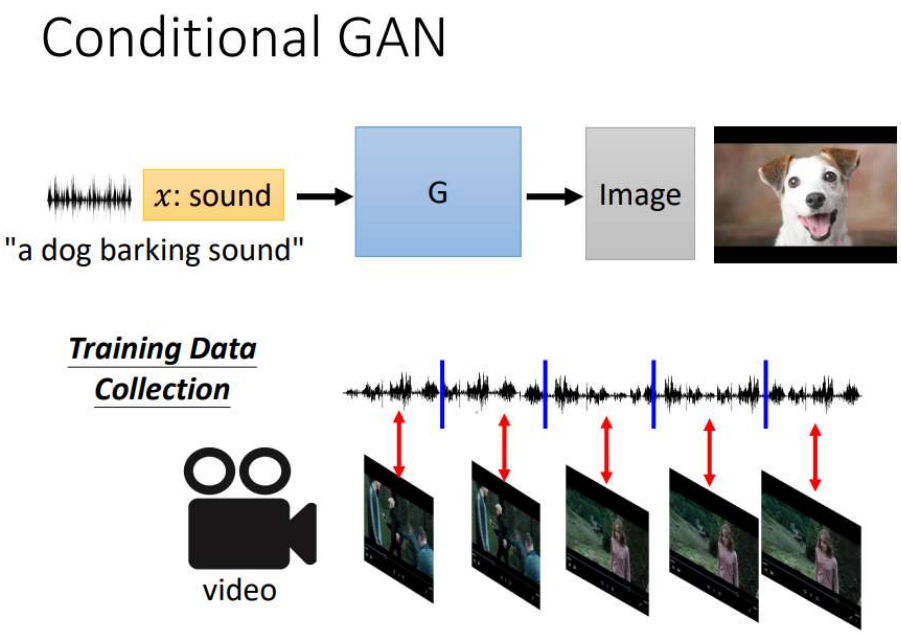
- 也可以用在看一張圖片產生一張圖片
- 稱為 image translation 或 Pix2pix

https://arxiv.org/abs/1611.07004

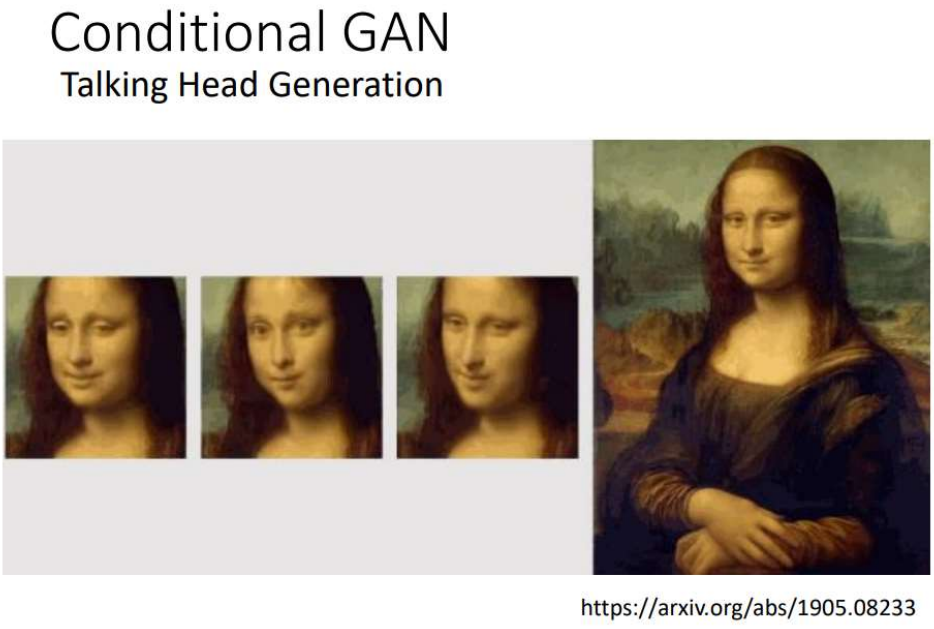


- 想要做到最好是 GAN + supervised leanring 同時使用
- Generator 在訓練的時候，一方面要去騙過 Discriminator 同時又想要產生一張圖片跟標準答案越像越好，同時做這兩件事產生的結果往往是最好的

https://arxiv.org/abs/1808.04108

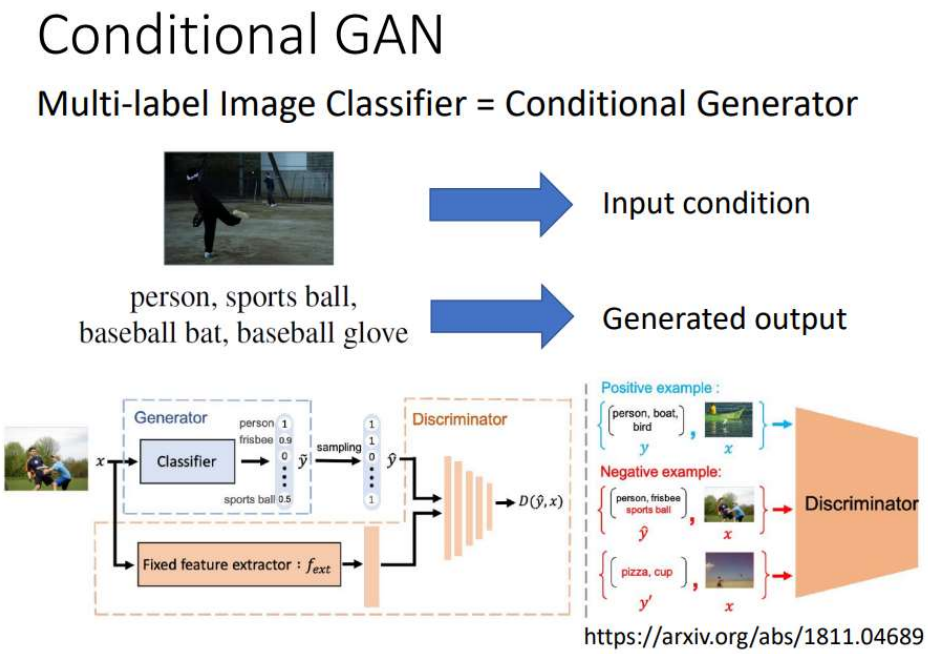


- 莫名其妙的應用
  - 給 Conditional GAN 聽一段聲音，產生一個圖片



https://arxiv.org/abs/1905.08233

- 用 Conditional GAN 產生一個會動的圖片



tags: `2022 李宏毅_機器學習`