Introduction of Generative Adversarial Network (GAN)

李宏毅

Hung-yi Lee

Generative Adversarial Network (GAN)

- How to pronounce “GAN”?

Google 小姐

Yann LeCun’s comment

What are some recent and potentially upcoming breakthroughs in unsupervised learning?

Yann LeCun, Director of AI Research at Facebook and Professor at NYU

Written Jul 29 · Upvoted by Joaquin Quiñonero Candela, Director Applied Machine Learning at Facebook and Huang Xiao

有史以來

Adversarial training is the coolest thing since sliced bread.

↳ 有好東西問世時会用這個俗語

I’ve listed a bunch of relevant papers in a previous answer.

Expect more impressive results with this technique in the coming years.

What’s missing at the moment is a good understanding of it so we can make it work reliably. It’s very finicky. Sort of like ConvNet were in the 1990s, when I had the reputation of being the only person who could make them work (which wasn’t true).

https://www.quora.com/What-are-some-recent-and-potentially-upcoming-breakthroughs-in-unsupervised-learning

Yann LeCun’s comment

What are some recent and potentially upcoming breakthroughs in deep learning?

Yann LeCun, Director of AI Research at Facebook and Professor at NYU

Written Jul 29 · Upvoted by Joaquin Quiñonero Candela, Director Applied Machine Learning at Facebook and Nikhil Garg, I lead a team of Quora engineers working on ML/NLP problems

……

The most important one, in my opinion, is adversarial training (also called GAN for Generative Adversarial Networks). This is an idea that was originally proposed by Ian Goodfellow when he was a student with Yoshua Bengio at the University of Montreal (he since moved to Google Brain and recently to OpenAI).

GAN和他的种种变形是近10年来他覺得ML最有趣的idea

This, and the variations that are now being proposed is the most interesting idea in the last 10 years in ML, in my opinion.

https://www.quora.com/What-are-some-recent-and-potentially-upcoming-breakthroughs-in-deep-learning

All Kinds of GAN …

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

2个 LSGAN < Least square GAN / Lose square GAN

GAN

ACGAN

BGAN

CGAN

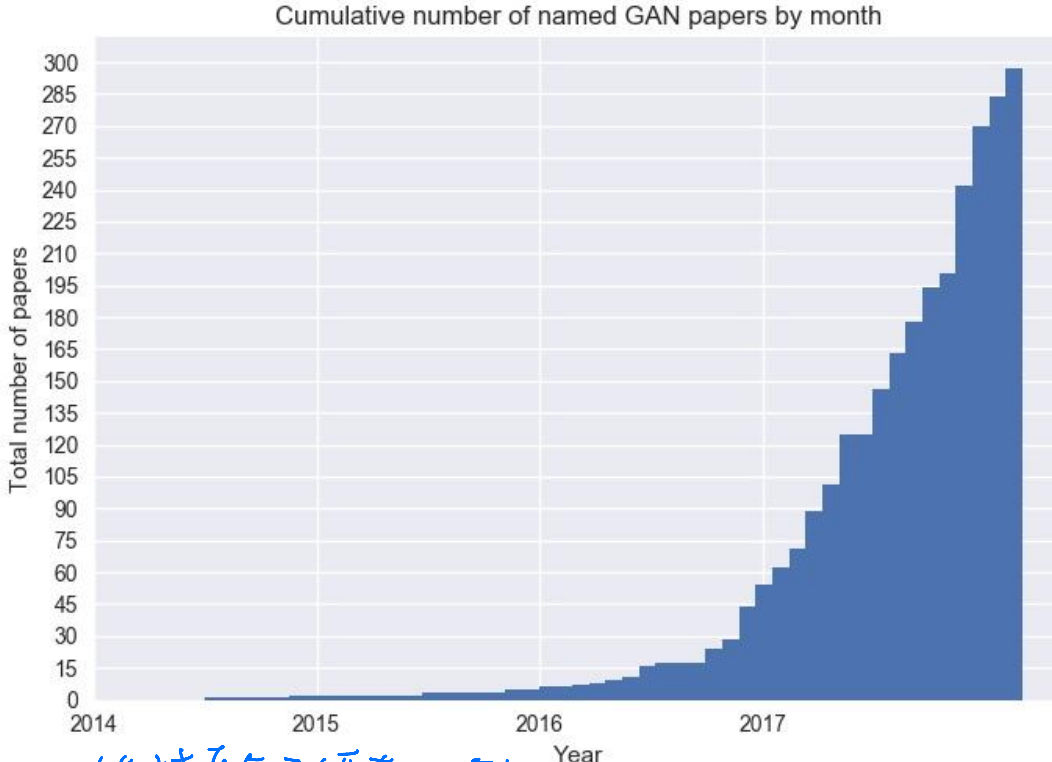
DCGAN

EBGAN

fGAN

GoGAN

⋮



⇒ 統計至今已經有300種的GAN了

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

AGAN or AEGAN ⇒ but 都被用掉了 ⇒ α-GAN

[2] We use the Greek α prefix for α-GAN, as AEGAN and most other Latin prefixes seem to have been taken

https://deephunt.in/the-gan-zoo-79597dc8c347.

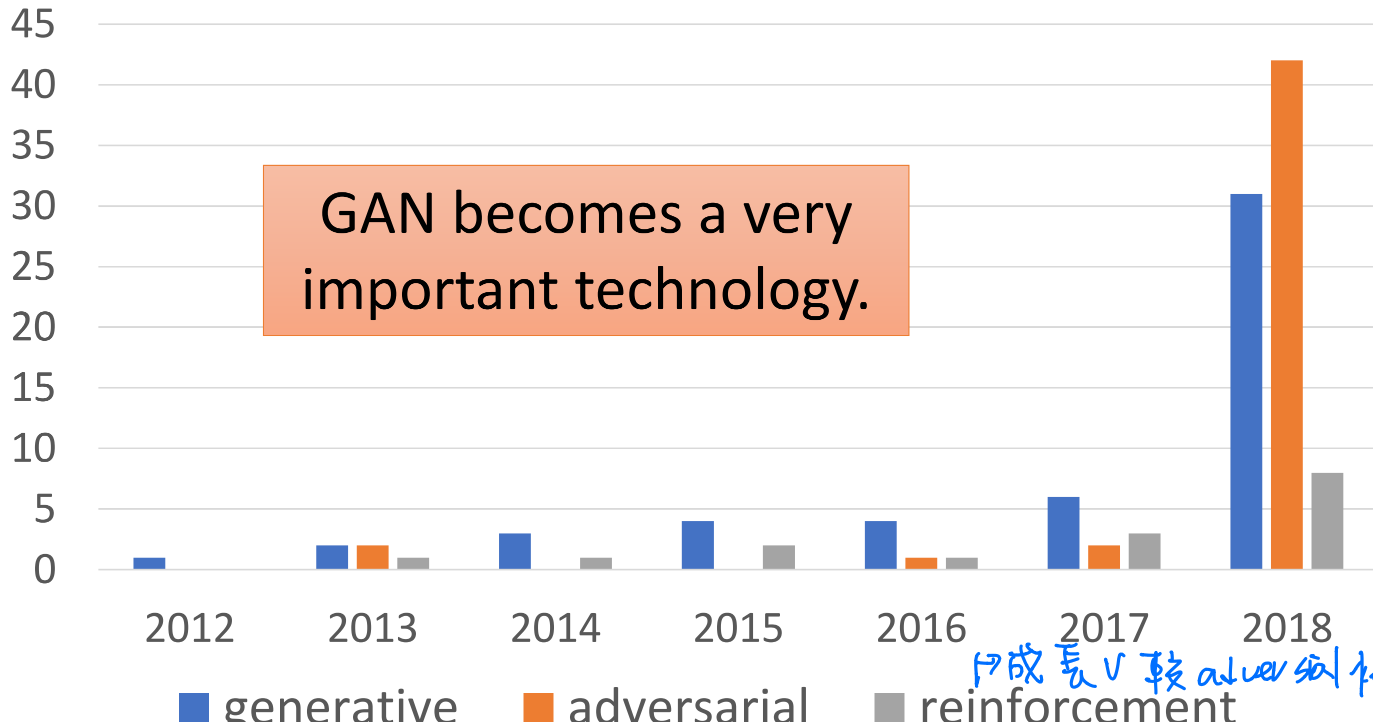
is signal Processing conference

ICASSP

Keyword search on session index page, so session names are included.

⇒ 近年 ICASSP 和 GAN 有關的 paper

Number of papers whose titles include the keyword



GAN becomes a very important technology.

成長 v 較 adversarial 後

不過有些雖然有 adversarial 這个 title but 和 GAN 无关

Outline

★ 給一个大的概念讓我們知道這个技術運作的情況

- Basic Idea of GAN
- GAN as structured learning
- Can Generator learn by itself?
- Can Discriminator generate?
- A little bit theory

☆ GAN 想做的事：讓機器生成東西 ex 影像, 產生句子, 文章

Generation

$\boxed{\text{need}}$ → 訓練一個 generator

We will control what to generate latter. → Conditional Generation (較有用)

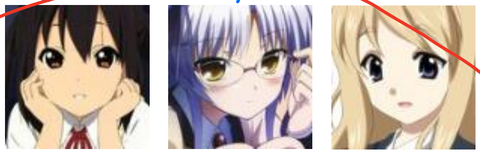
→ 可以輸入一些條件 ex 輸入文字讓機器產生對應的圖片

可以控制的 ex 文字/影像

Image Generation

有個 Gaussian distribution 裡 random sample 一個 vector, then 把 vector 丟到 generator 中

$$\begin{bmatrix} 0.3 \\ -0.1 \\ \vdots \\ -0.7 \end{bmatrix} \begin{bmatrix} 0.1 \\ -0.1 \\ \vdots \\ 0.7 \end{bmatrix} \begin{bmatrix} -0.3 \\ 0.1 \\ \vdots \\ 0.9 \end{bmatrix} \rightarrow \text{NN Generator} \rightarrow$$



In a specific range

(不同 vector) →

→ 產生不同 image

Sentence Generation

$$\begin{bmatrix} 0.3 \\ -0.1 \\ \vdots \\ -0.7 \end{bmatrix} \begin{bmatrix} 0.1 \\ -0.1 \\ \vdots \\ 0.2 \end{bmatrix} \begin{bmatrix} -0.3 \\ 0.1 \\ \vdots \\ 0.5 \end{bmatrix} \rightarrow \text{NN Generator} \rightarrow$$

How are you?
Good morning.
Good afternoon.

輸入隨機的向量 → output 想要的 object

→ 其實沒有很大的用處

Powered by: http://mattya.github.io/chainer-DCGAN/

Basic Idea of GAN

It is a neural network (NN), or a function.

↳ 輸入一个東西 輸出一个東西

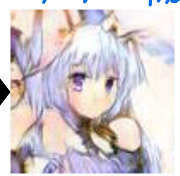
vector → Generator → image

high dimensional vector

Ex 在影像生成，其 output 為 高維的向量 (很長) → 會對應到每个 pixel 的 color

→ 二次元人物頭像生成

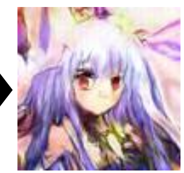
$\begin{bmatrix} 0.1 \\ -3 \\ \vdots \\ 2.4 \\ 0.9 \end{bmatrix}$ → Generator →



Each dimension of input vector represents some characteristics.

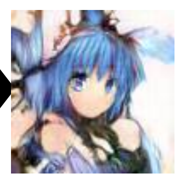
(每个維度會對應到圖片的某种特徵)

$\begin{bmatrix} \boxed{3} \\ -3 \\ \vdots \\ 2.4 \\ 0.9 \end{bmatrix}$ → Generator →



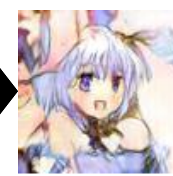
Longer hair

$\begin{bmatrix} 0.1 \\ 2.1 \\ \vdots \\ \boxed{5.4} \\ 0.9 \end{bmatrix}$ → Generator →



blue hair

$\begin{bmatrix} 0.1 \\ -3 \\ \vdots \\ 2.4 \\ \boxed{3.5} \end{bmatrix}$ → Generator →



Open mouth

Basic Idea of GAN

It is a neural network (NN), or a function.

image → Discri-minator → scalar (數值)

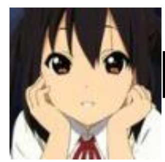
大→越 real (真實)

Larger value means real, smaller value means fake.

↳ 此數值越大表示其 quality (質量) 越高 → 看起來越像真實的圖片

 → Discri-minator → 1.0

(可能 the max)

 → Discri-minator → 1.0

 → Discri-minator → 0.1

 → Discri-minator → 0.1

What is the relation between generator & discriminator.

↳ [对抗 / 合作] 关係

Basic Idea of GAN

↳ 就好像獵食者 & 獵物之間的关係

Generator

Brown

veins

有无葉脈的紋路

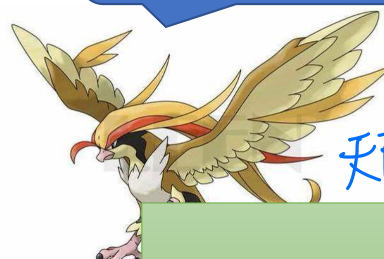
if no 被吃

有别的標準來判斷 弱的 吃

Butterflies are not brown

Butterflies do not have veins

……..



天敵

Discriminator

麻雀會吃枯葉蝶 ∴在天擇的压力下，变成棕色不会被吃

↳ 也会演化

First 準備 database 內有很多真實人物的頭像

一開始參數 is random ⇒ 也不知如何產生

Basic Idea of GAN

二次元頭像 ⇒ 只能產生雜訊

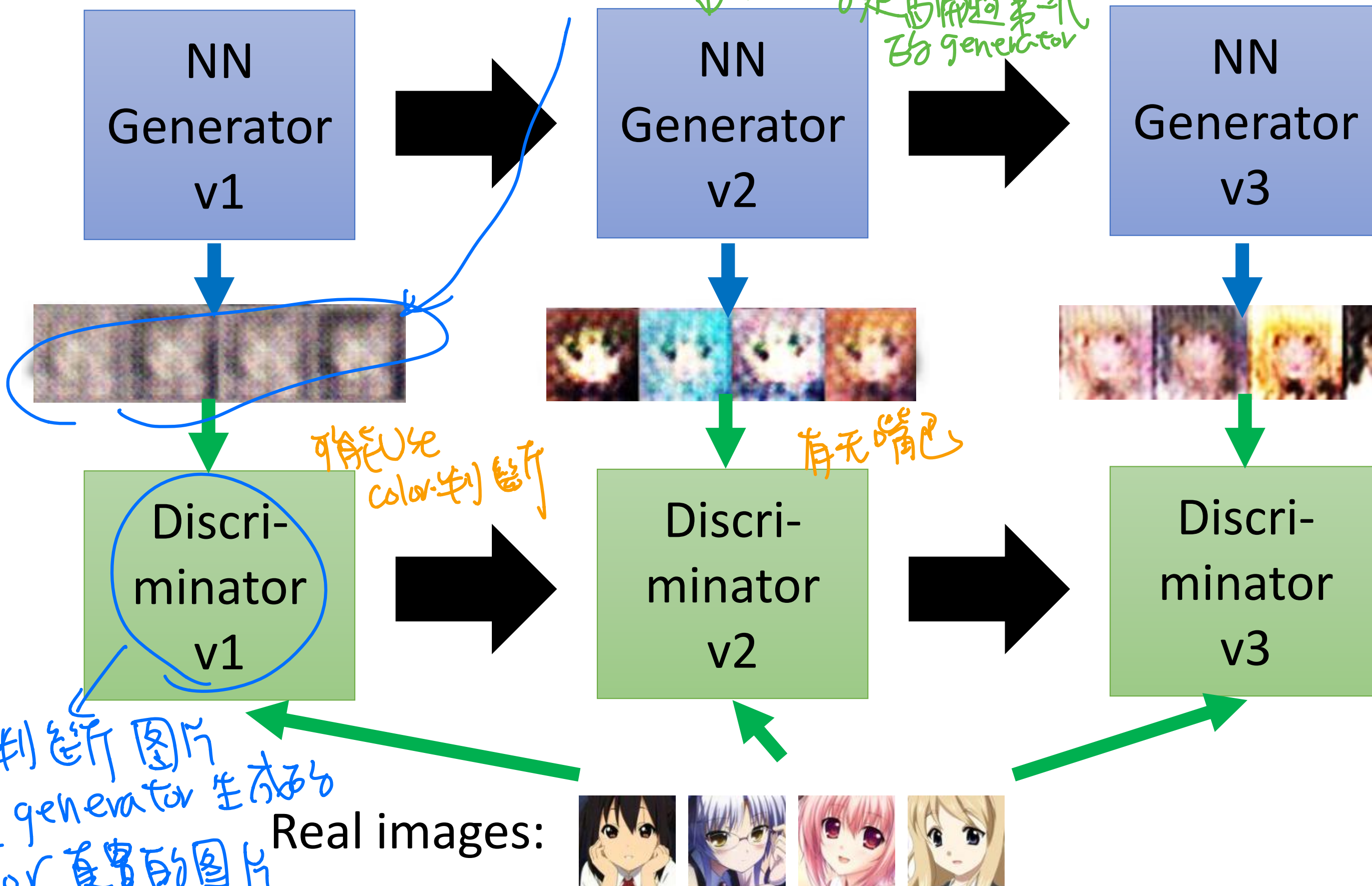
This is where the term "***adversarial***" comes from.

也可以看作是合作

对抗 (兩個network 对抗)

You can explain the process in different ways.......

要做的是騙過第一代的generator



可能use color判斷

有无嘴巴

会越來越進化

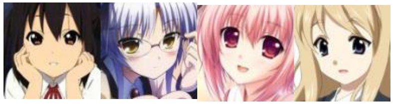
所以generator 產生出的圖片会 → 越real

判斷圖片是generator生成的 or 真實的圖片

Basic Idea of GAN (和平的比喻)

Generator (student)

Discriminator (teacher)



Generator v1

Discriminator v1

沒有兩個圈

Generator v2

Discriminator v2

沒有彩色

Generator v3

為什麼不自己學？

為什麼不自己做？

Generator v.s. Discriminator

- 寫作敵人，唸做朋友

Algorithm (GAN) → 操作原理

1、參數 random initialized generator & discriminator
2、iterative 的 train generator & discriminator.
L[1. 2. generator 參數固定, 只調 discriminator 參數

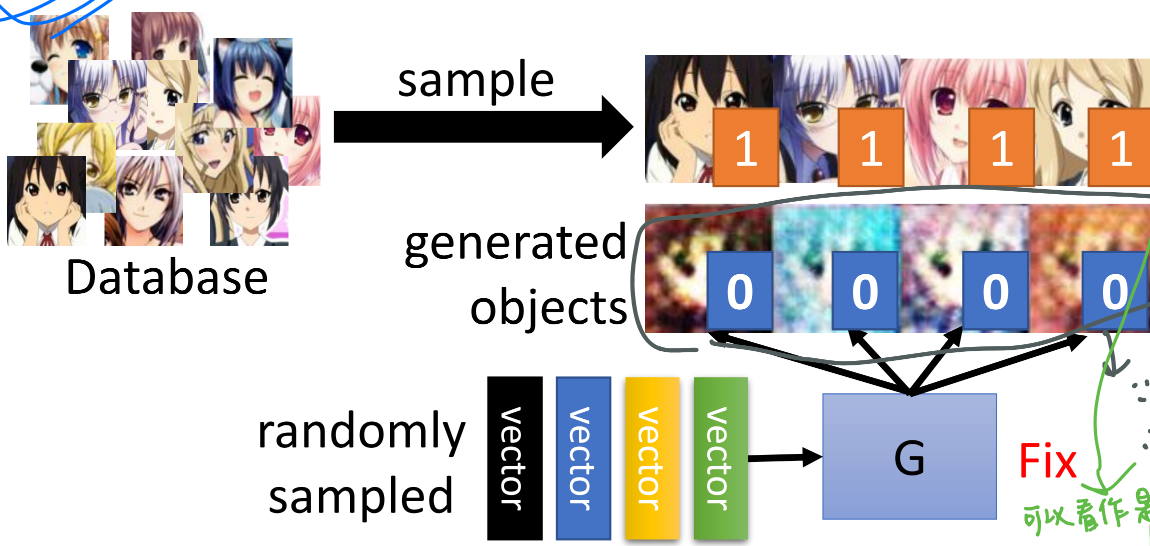
- Initialize generator and discriminator

- In each training iteration:

兩組圖片 [generator 生成的
database sample 出來的

Step 1: Fix generator G, and update discriminator D

then 調整 discriminator 參數
→ how to 調? database sample 出來的
ANS: if image is 真實的 就給高分



if is generator 產生 就給低分

∵ input is random
∴ output 不會特別好

可以看作是 regression problem
L classification

目的: 訓練 network, 使其 output (大: database
小: generator

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

☆ output 要離 1 越近 越好 (上面)
0 〃 〃 (下面)

Algorithm

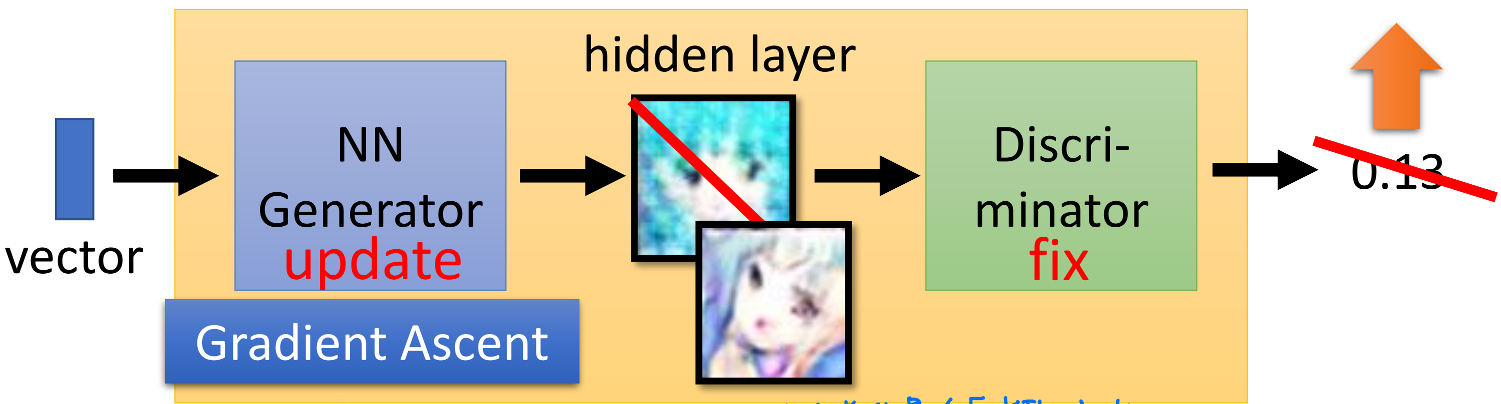
- Initialize generator and discriminator G D
- In each training iteration:

∵ discriminator 訓練好了 ∴ how fix

Step 2: Fix discriminator D, and update generator G

Generator learns to "fool" the discriminator

hope generator 產生出來的圖片 discriminator 可以給比較高的分數 ⇒ generator 產出的是較真實的



實際上在 write code 時会把 generator & discriminator 合起來當作是一个巨大的 network

large network ⇒ 目標 output 值越大越好

train network 時，固定最后几个 hidden layer，只調前几个 hidden layer

☆ 不是 Gradient descent → 目標是越小越好

Algorithm

Initialize θ_d for D and θ_g for G

可以不一樣

- In each training iteration:

batch size ex 32.64

Learning D

- Sample m examples $\{x^1, x^2, \ldots, x^m\}$ from database
- Sample m noise samples $\{z^1, z^2, \ldots, z^m\}$ from a distribution // 可能沒有顯著影響 : 可以是 Gaussian / uniform

參數 = n維自己調

產生m張圖片

x tilde (telda)

- Obtaining generated data $\{\tilde{x}^1, \tilde{x}^2, \ldots, \tilde{x}^m\}$, $\tilde{x}^i = G(z^i)$

want max objection ft

- Update discriminator parameters θ_d to maximize

只是一个範例 搞不好用其他的也不錯

原始GAN paper寫的

learning rate 如何去調 : 可用 Adam 方法

D的值会落在 0~1之間 ∵ if no log出來会有問題

- $\tilde{V} = \frac{1}{m}\sum_{i=1}^{m} logD\left(x^i\right) + \frac{1}{m}\sum_{i=1}^{m} log\left(1 - \underbrace{D\left(\tilde{x}^i\right)}_{\text{假的圖片}}\right)$

sigmoid

gradient ascent

- $\theta_d \leftarrow \theta_d + \eta \nabla \tilde{V}(\theta_d)$

if "−" : descent

if want loss 越小越好

要執行n次是可以調的

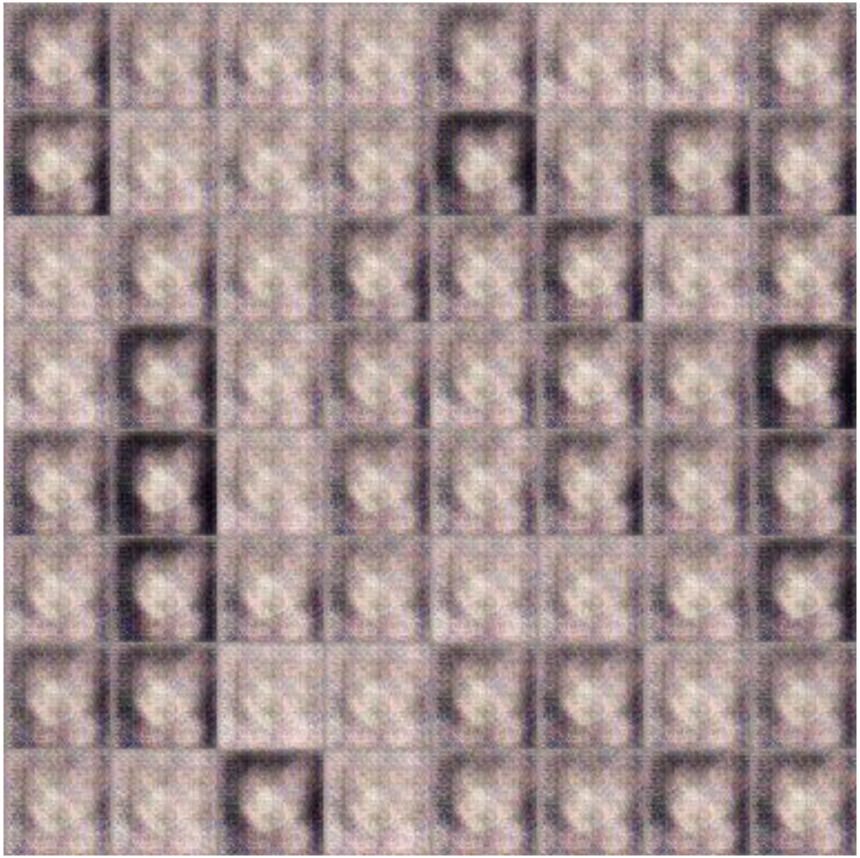
Learning G

- Sample m noise samples $\{z^1, z^2, \ldots, z^m\}$ from a distribution
- Update generator parameters θ_g to maximize

產生一張圖片

- $\tilde{V} = \frac{1}{m}\sum_{i=1}^{m} log\left(\underbrace{D\left(G\left(z^i\right)\right)}_{\text{數值}}\right)$
- $\theta_g \leftarrow \theta_g - \eta \nabla \tilde{V}(\theta_g)$

Anime Face Generation



100 updates

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

Anime Face Generation

有 eye

1000 updates

Anime Face Generation

hmncs

2000 updates

Anime Face Generation

K eye

5000 updates

Anime Face Generation

水彩畫

10,000 updates

Anime Face Generation

20,000 updates

Anime Face Generation

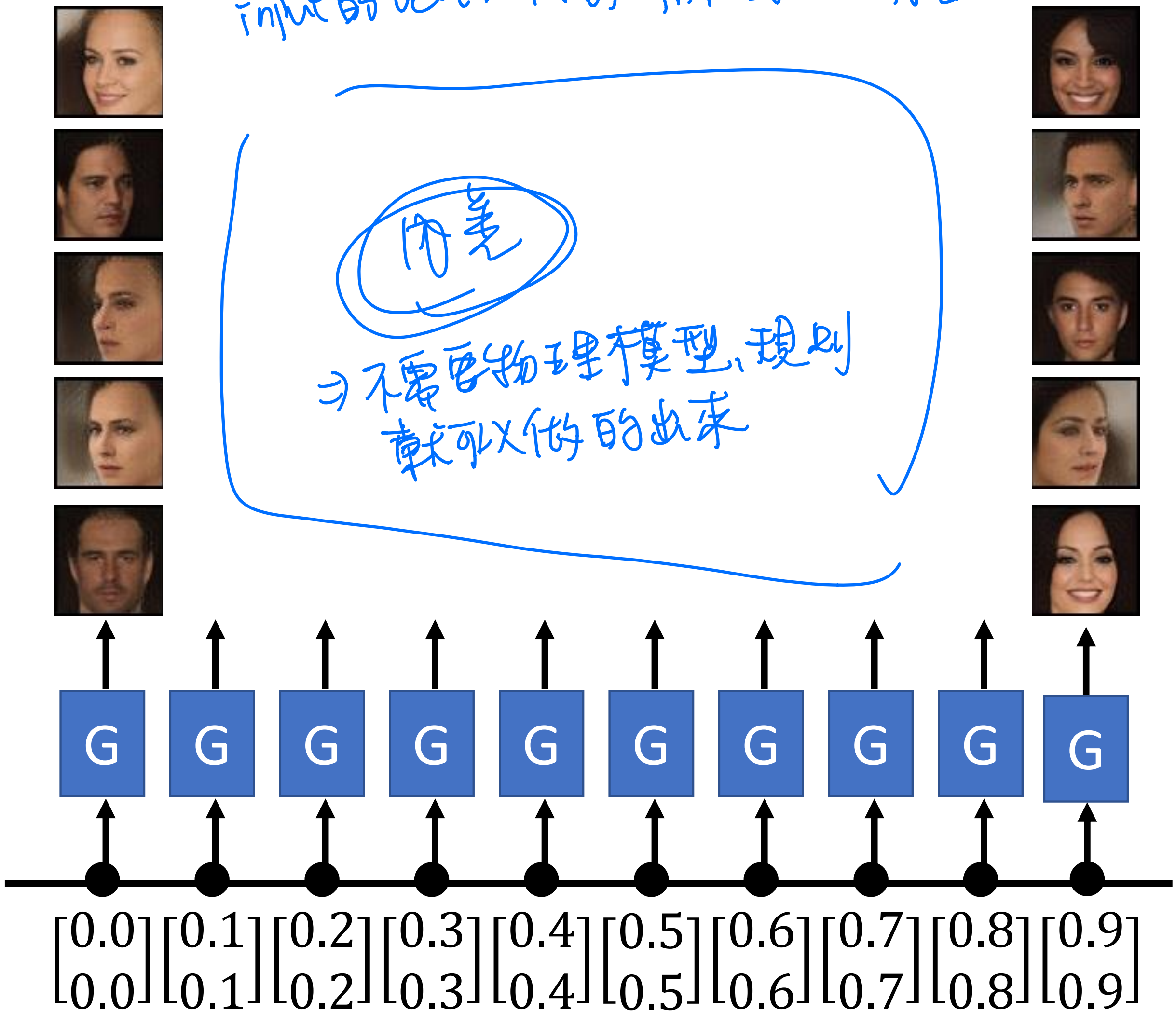
50,000 updates

The faces generated by machine.

圖片生成：
吳宗翰、謝濬丞、
陳延昊、錢柏均

input的vector代表輸出圖片的特性

2次元
人物
生成



內差

⇒不需要物理模型、規則
就可以做的出來

機器可以生成沒有看過的圖

感謝陳柏文同學提供實驗結果

Outline

- Basic Idea of GAN
- GAN as structured learning (原理)
- Can Generator learn by itself?
- Can Discriminator generate?
- A little bit theory

可以视为 Structed Learning 的特例

Structured Learning

Machine learning is to find a function f

$$f : X \rightarrow Y$$

Regression: output a scalar

Classification: output a "class" (one-hot vector)

1	0	0

Class 1

0	1	0

Class 2

0	0	1

Class 3

Structured Learning/Prediction: output a sequence, a matrix, a graph, a tree ……

Output is composed of components with dependency

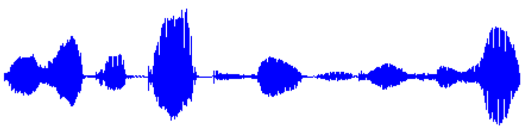
Output Sequence $f: X \rightarrow Y$

Machine Translation

X : “機器學習及其深層與結構化”
(sentence of language 1)

Y : “Machine learning and having it deep and structured”
(sentence of language 2)

Speech Recognition

X : 
(speech)

Y : 感謝大家來上課”
(transcription)

Chat-bot

X : “How are you?”
(what a user says)

Y : “I’m fine.”
(response of machine)

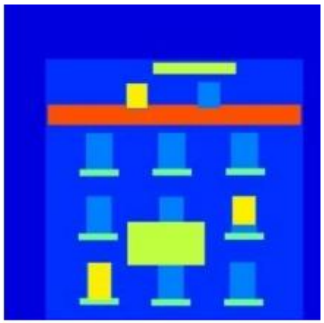
Output Matrix

$$f : X \rightarrow Y$$

Image to Image

Colorization:

X :



Y :

Ref: https://arxiv.org/pdf/1611.07004v1.pdf

Text to Image

X : "this white and yellow flower have thin white petals and a round yellow stamen"

Y :

ref: https://arxiv.org/pdf/1605.05396.pdf

Why Structured Learning Challenging?

• 假設我們把机器每一种可能的output視為一个class
structure learning 是一个極端的 one shot learning or zero-shot

可以被視為極端的

→根本沒有範例 or 很少很少範例

train / test data 可能不会重複

- **One-shot/Zero-shot Learning**:
  - In classification, each class has some examples.
  - In structured learning,
    - If you consider each possible output as a "class" ……

⇒ 表示在 training data 只会出現一次　testing data 中的 class 可能在 training 沒出現過

    - Since the output space is huge, most "classes" do not have any training data.
    - Machine has to create new stuff during testing.

•如何學到一般化，如何學著輸出從沒有看過的東西變成很重要

    - Need more intelligence

机器要学会創造 才能解 structure learning

(training 沒 see 過

Why Structured Learning Challenging?

- Machine has to learn to do ***planning***
  - Machine generates objects component-by-component, but it should have a big picture in its mind.
  - Because the output components have dependency, they should be considered globally.

規畫

重要的不是產生了什麼component 而是component之間的關係

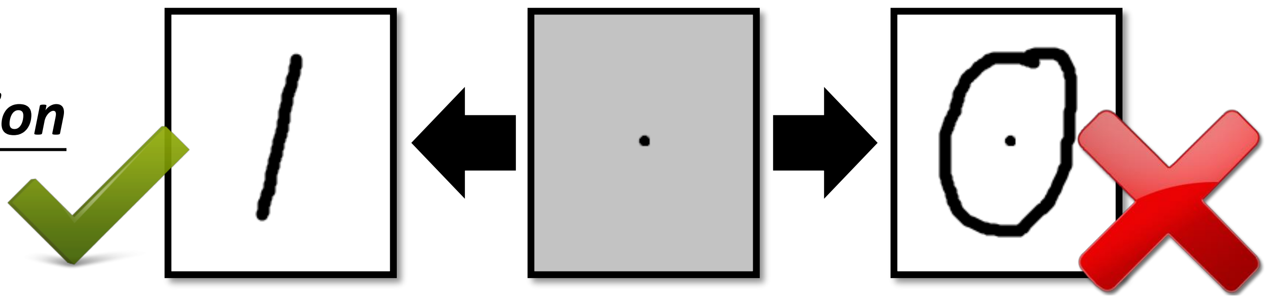
⇒ ∵ 單看一部份 不知道 output好不好

機器要產生很複雜的物件

是由簡單的 component 組成

need 看全局

Image Generation



Sentence Generation

這個婆娘不是人

九天玄女下凡塵

GAN 是 structured Learning 的 sol.

Structured Learning Approach

2个方法

Generator

Learn to generate the object at the component level

一个一个 component 分開去產生這個物件
(可能会失去大局观)

Discriminator

Evaluating the whole object, and find the best one

從產生完一个完整的物件以后
再去從整体來看產生的物件好不好 (but 很難做 generation)

1. Bottom Up

+

2. Top Down

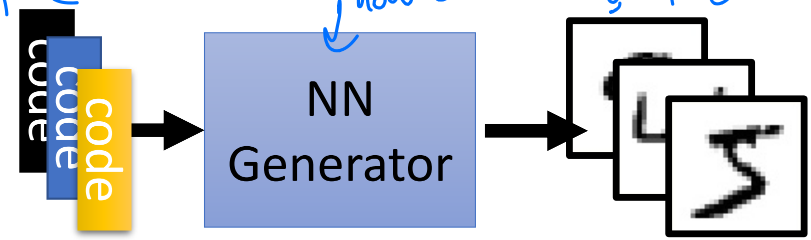
→ Generative Adversarial Network (GAN)

Outline

- Basic Idea of GAN
- GAN as structured learning
- **Can Generator learn by itself?**
- Can Discriminator generate?
- A little bit theory

傳統給 input & output
就會知道結果

Generator



code: $\begin{bmatrix} 0.1 \\ -0.5 \end{bmatrix}$ $\begin{bmatrix} 0.1 \\ 0.9 \end{bmatrix}$ $\begin{bmatrix} 0.2 \\ -0.1 \end{bmatrix}$ $\begin{bmatrix} 0.3 \\ 0.2 \end{bmatrix}$

(where does they come from?)

Image:

和 train supervised learning same
用 gradient descent train

$\begin{bmatrix} 0.1 \\ 0.9 \end{bmatrix}$ → NN Generator → image ↔ (image of 1)

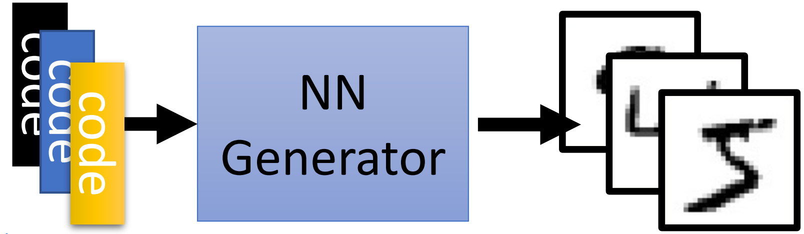
As close as possible

實際上就是輸出很長的向量

As close as possible

c.f. (image of 1) → NN Classifier → $\begin{bmatrix} y_1 \\ y_2 \\ \vdots \end{bmatrix}$ ↔ $\begin{bmatrix} 1 \\ 0 \\ \vdots \end{bmatrix}$

Generator



vectors

可能它會希望input的vector和output的特徵有关

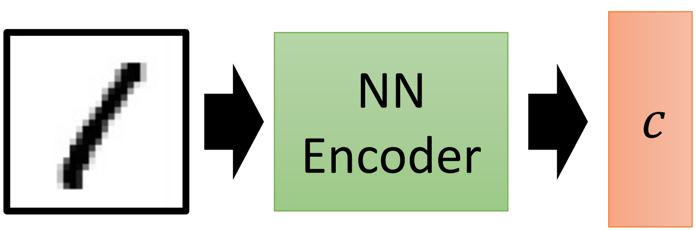
code: $\begin{bmatrix} 0.1 \\ -0.5 \end{bmatrix}$ $\begin{bmatrix} 0.1 \\ 0.9 \end{bmatrix}$ $\begin{bmatrix} 0.2 \\ -0.1 \end{bmatrix}$ $\begin{bmatrix} 0.3 \\ 0.2 \end{bmatrix}$

(where does they come from?)

Image:

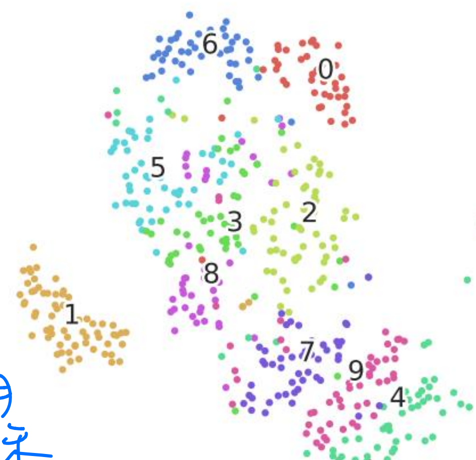
∴ learn encoder

Encoder in auto-encoder provides the code ☺



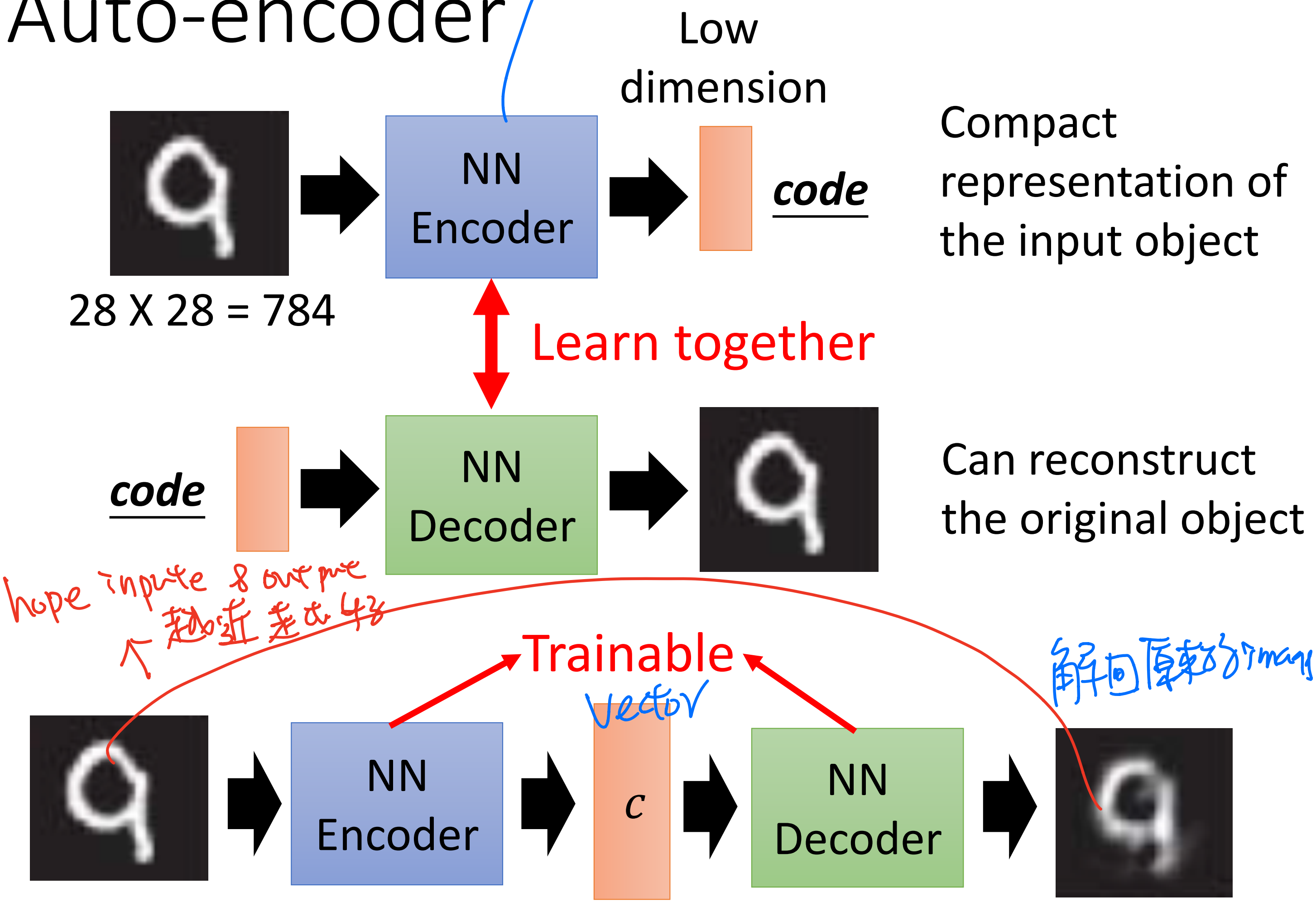
图片

把其特征用一个向量表示

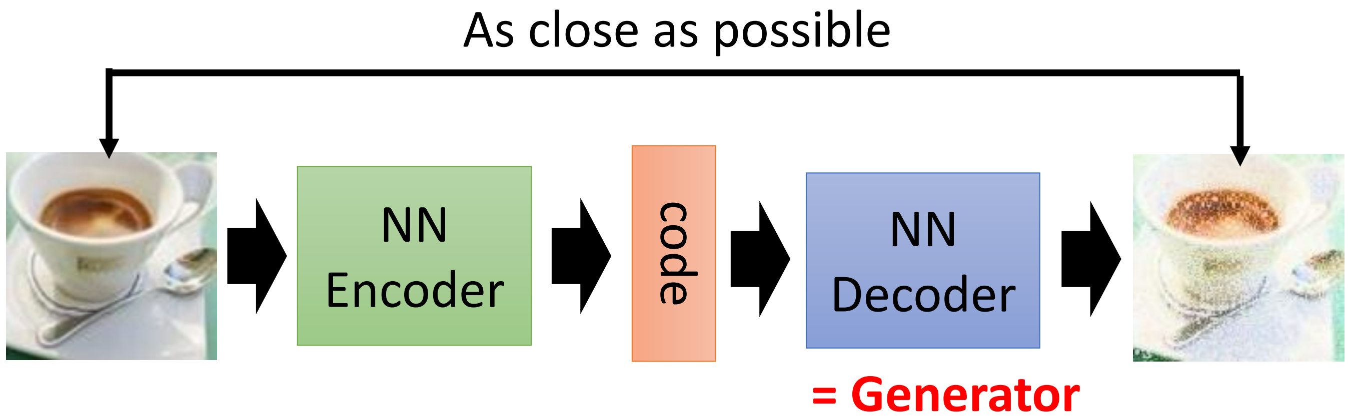


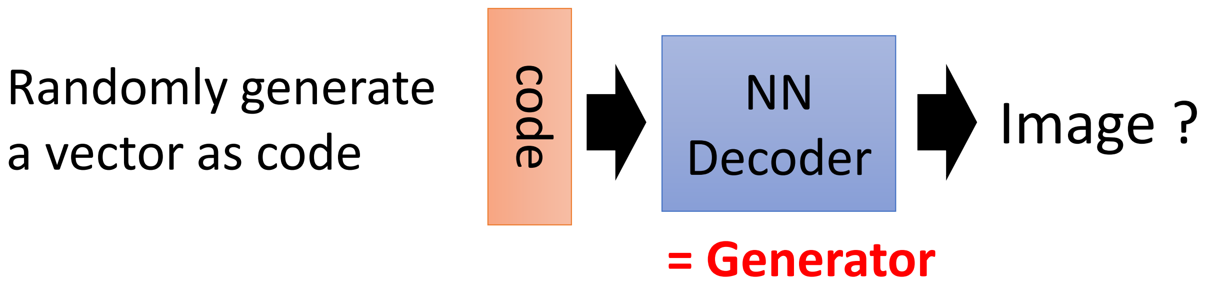
Auto-encoder

本身無法train 要加个decoder才可



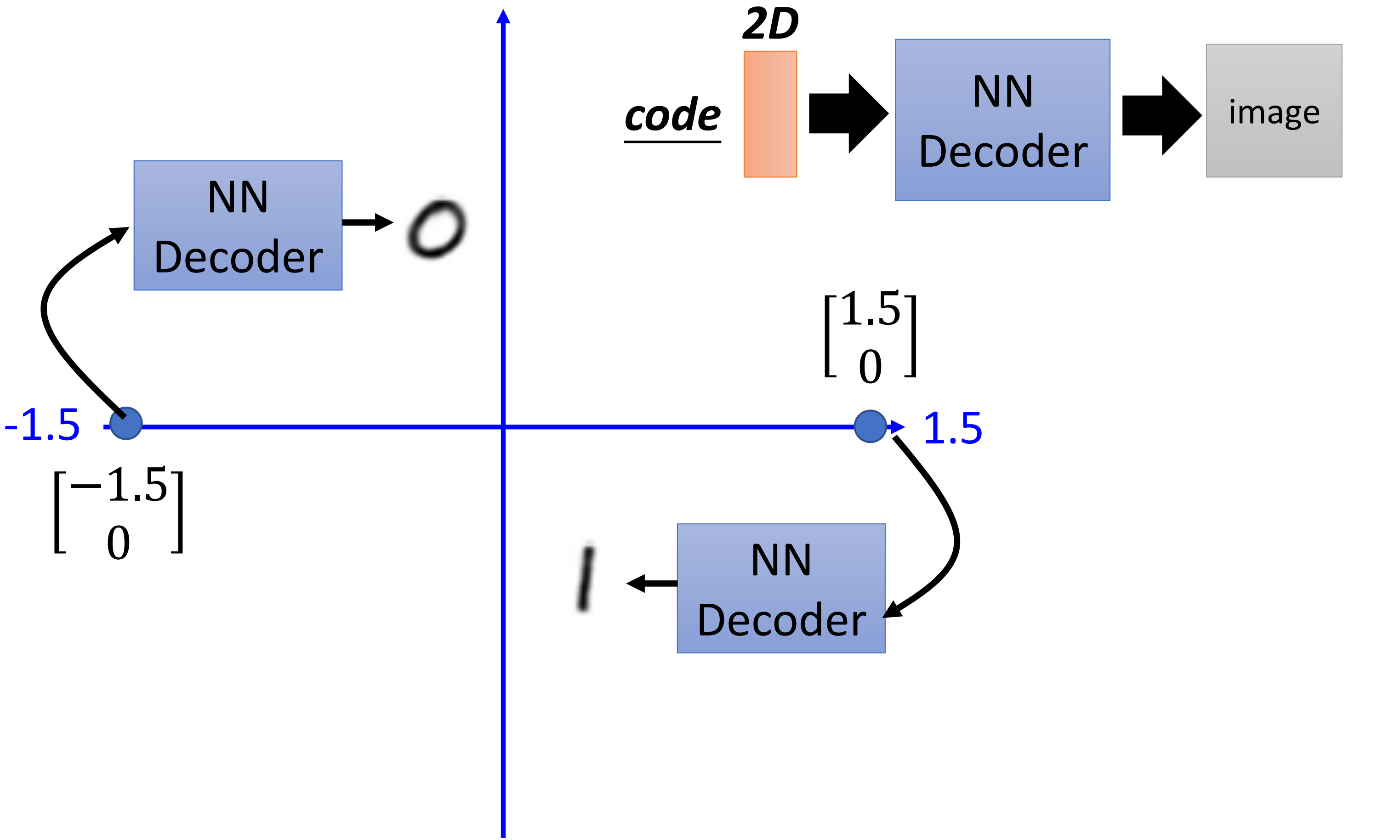
Auto-encoder





(real examples)

Auto-encoder

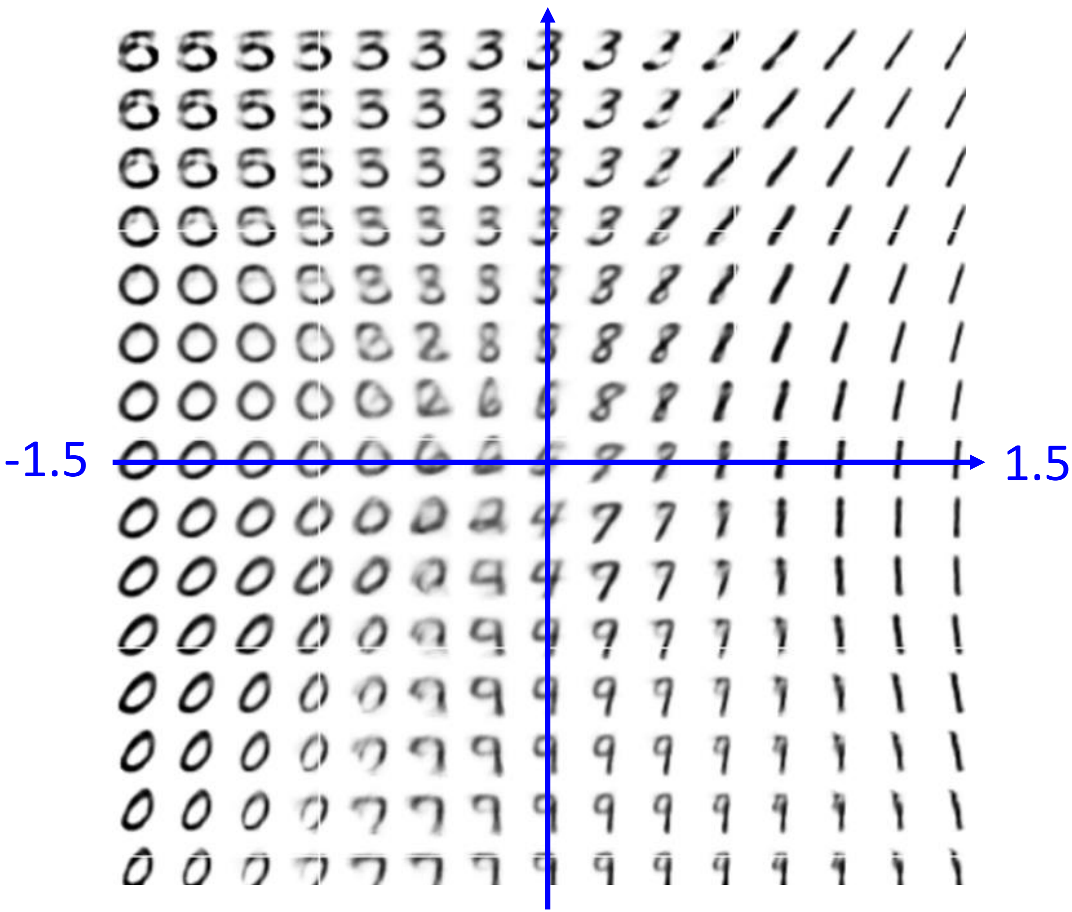


(real examples)

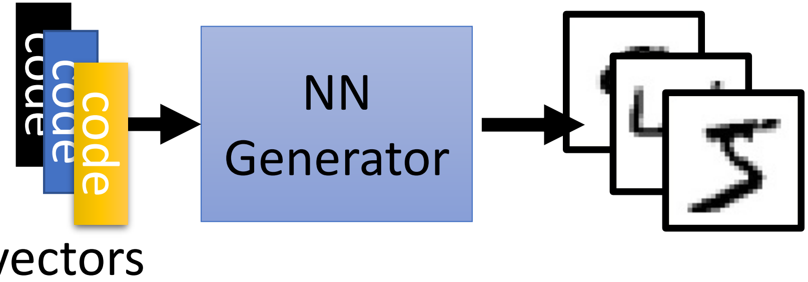
Auto-encoder

和方向有关

和 有关



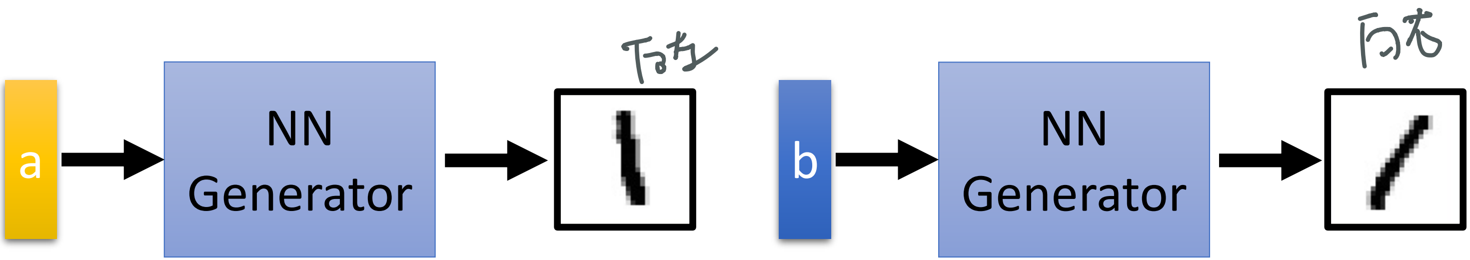
Auto-encoder

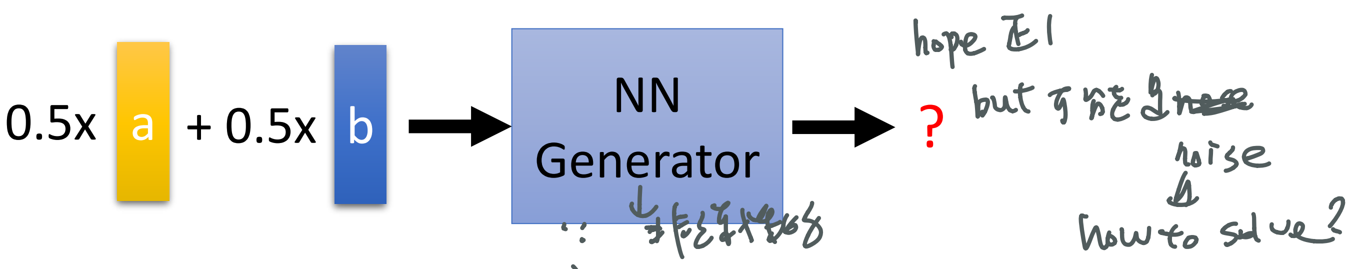


code: (where does them come from?)

$\begin{bmatrix} 0.1 \\ -0.5 \end{bmatrix}$ $\begin{bmatrix} 0.1 \\ 0.9 \end{bmatrix}$ $\begin{bmatrix} 0.2 \\ -0.1 \end{bmatrix}$ $\begin{bmatrix} 0.3 \\ 0.2 \end{bmatrix}$

Image:

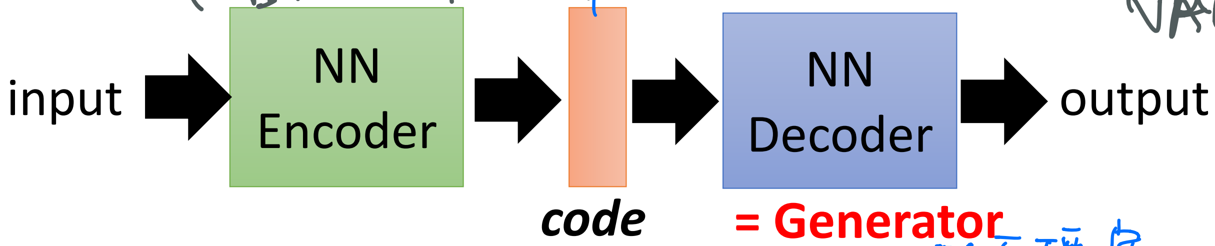




Auto-encoder

(if train 不起來 增加 dim 來減少有效) low dimension → 5,10 dim need 調配

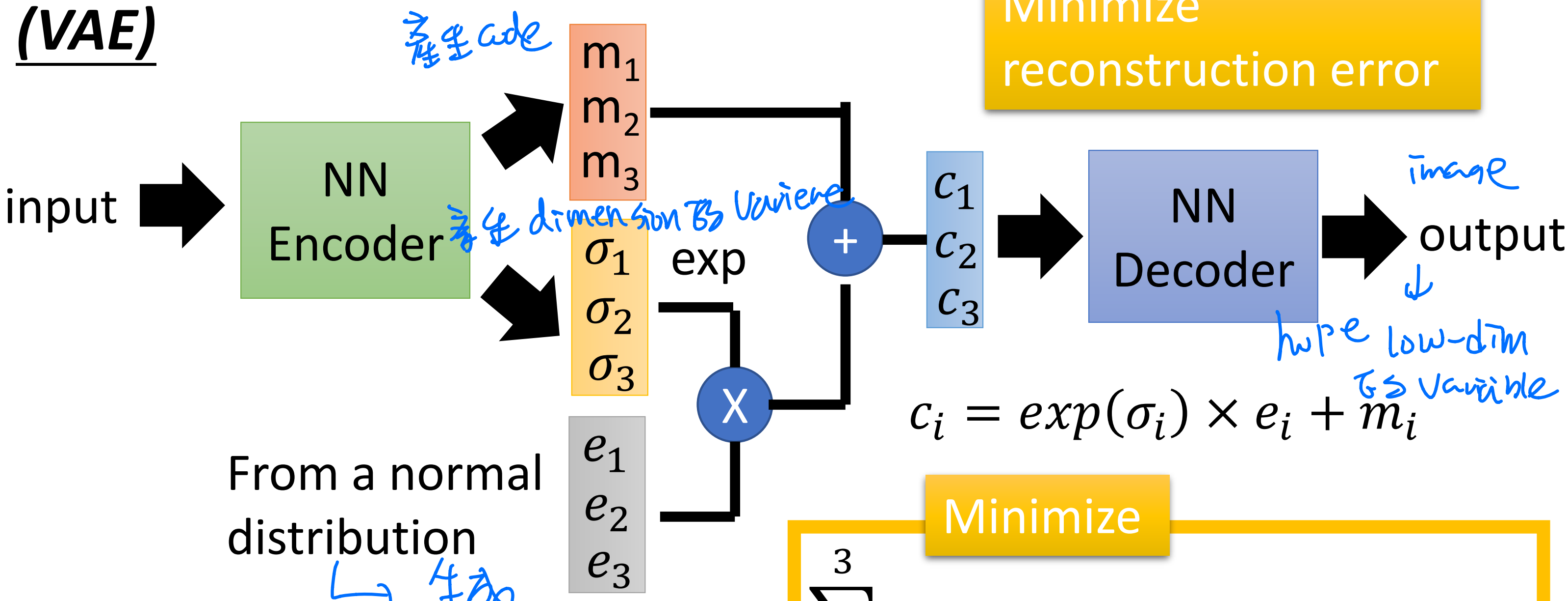
VAE



Variational Auto-encoder (VAE)

⇒ 可以把 decoder train 的更穩定

Minimize reconstruction error



$$c_i = exp(\sigma_i) \times e_i + m_i$$

Minimize

$$\sum_{i=1}^{3} (exp(\sigma_i) - (1 + \sigma_i) + (m_i)^2)$$

最后train出來的 σ ⇒ 0

—所以的 σ 不能太小，更加近似的

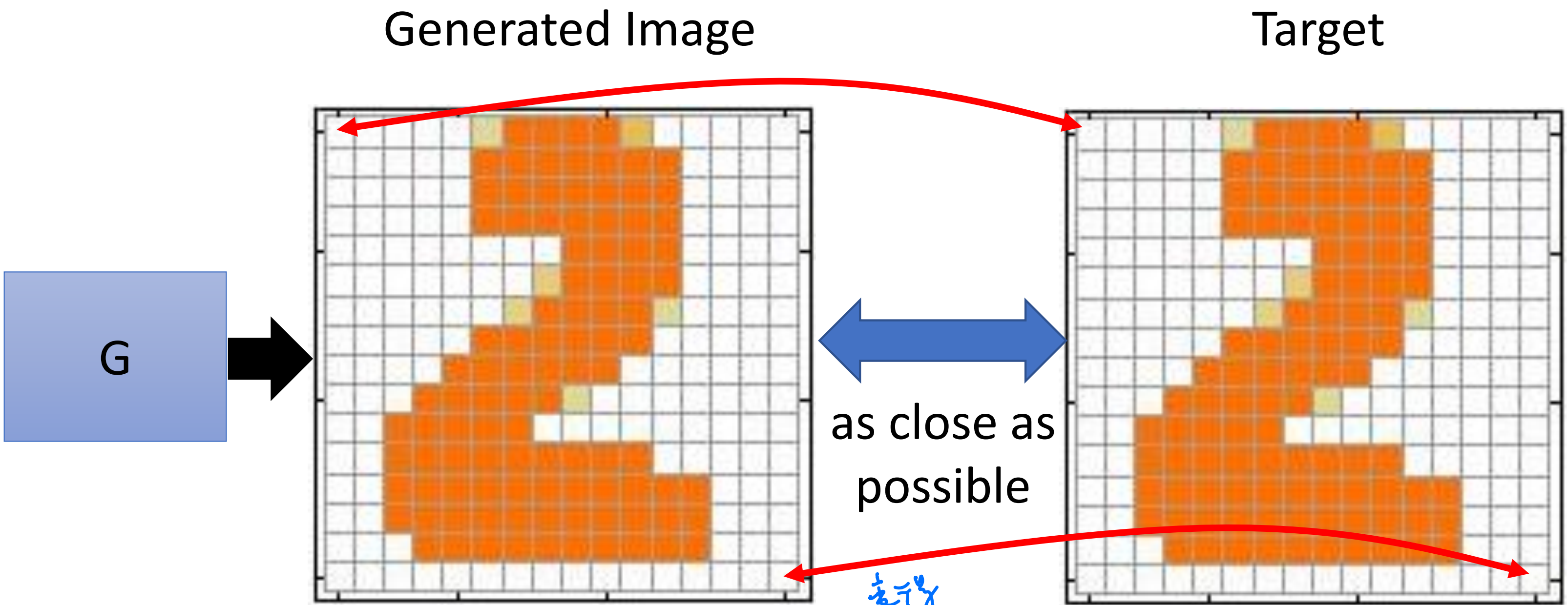
Auto-encoder ?

What do we miss?

把這兩張圖片表示成 vector

們 算 兩個 vector 的 euclidean distance ⇒ 這個就是要 minimize 的對象

hope 越小越好



要不要

⇒可能沒有辦法完全 copy image ⇒ 會有取捨 ⇒ 在哪些部份 → 重要

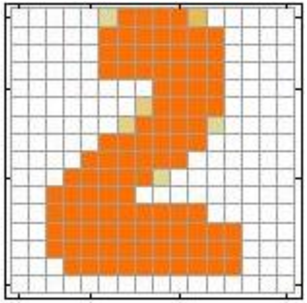
It will be fine if the generator can truly copy the target image.

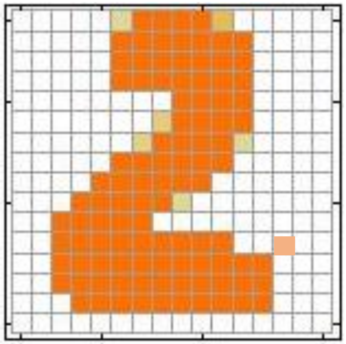
What if the generator makes some mistakes …….

Some mistakes are serious, while some are fine.

What do we miss?

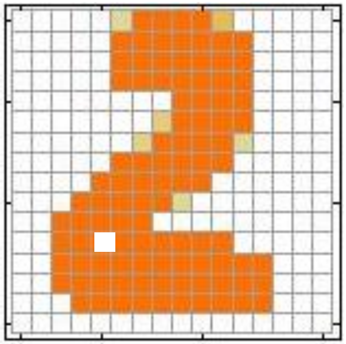
Target





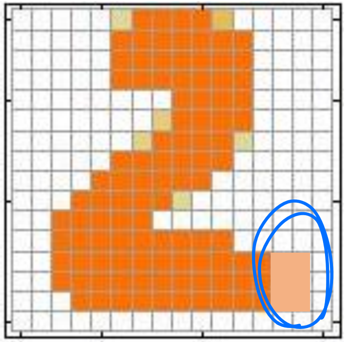
1 pixel error

我覺得不行



1 pixel error

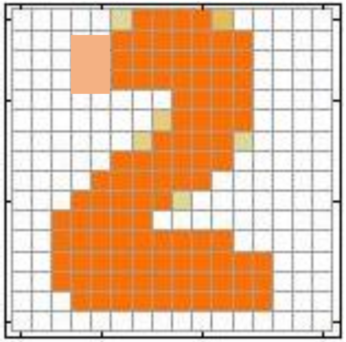
我覺得不行



6 pixel errors

我覺得其實可以

筆畫太長



6 pixel errors

我覺得其實可以

GAN train：可產生圖片

Auto-encoder train：need 更大的network 才能和GAN結果相近

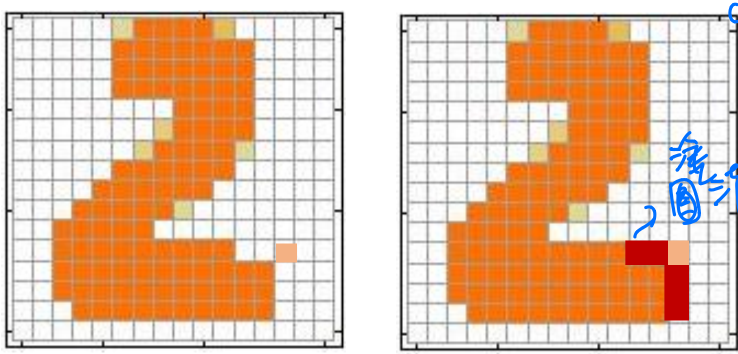
train 一個 generator 產生好的圖片 會需要改一個 network 的架構 沒有那麼容易

What do we miss?

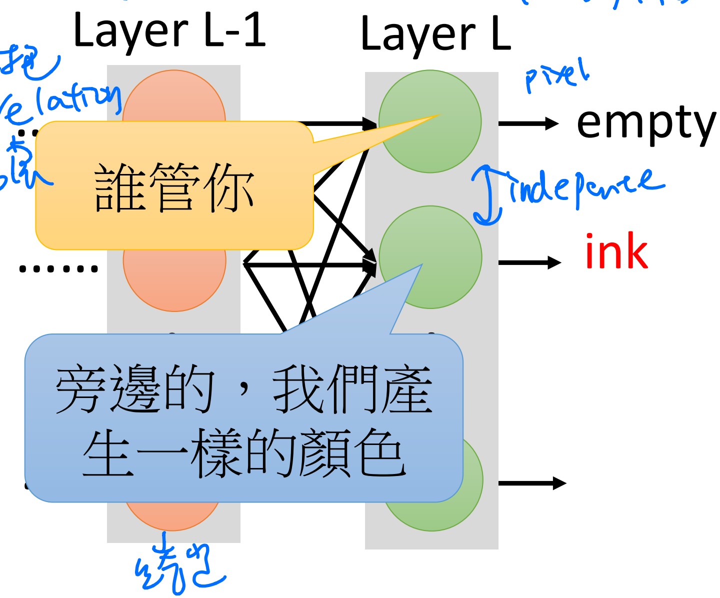
Each neural in output layer corresponds to a pixel.

容易的把 component & component 之間的關係做

• 看一個維度無法考慮 pixel & pixel 之間的關係 but 加hidden layer 可以把 correlation 考慮



我覺得不行　　我覺得其實可以



The relation between the components are critical.

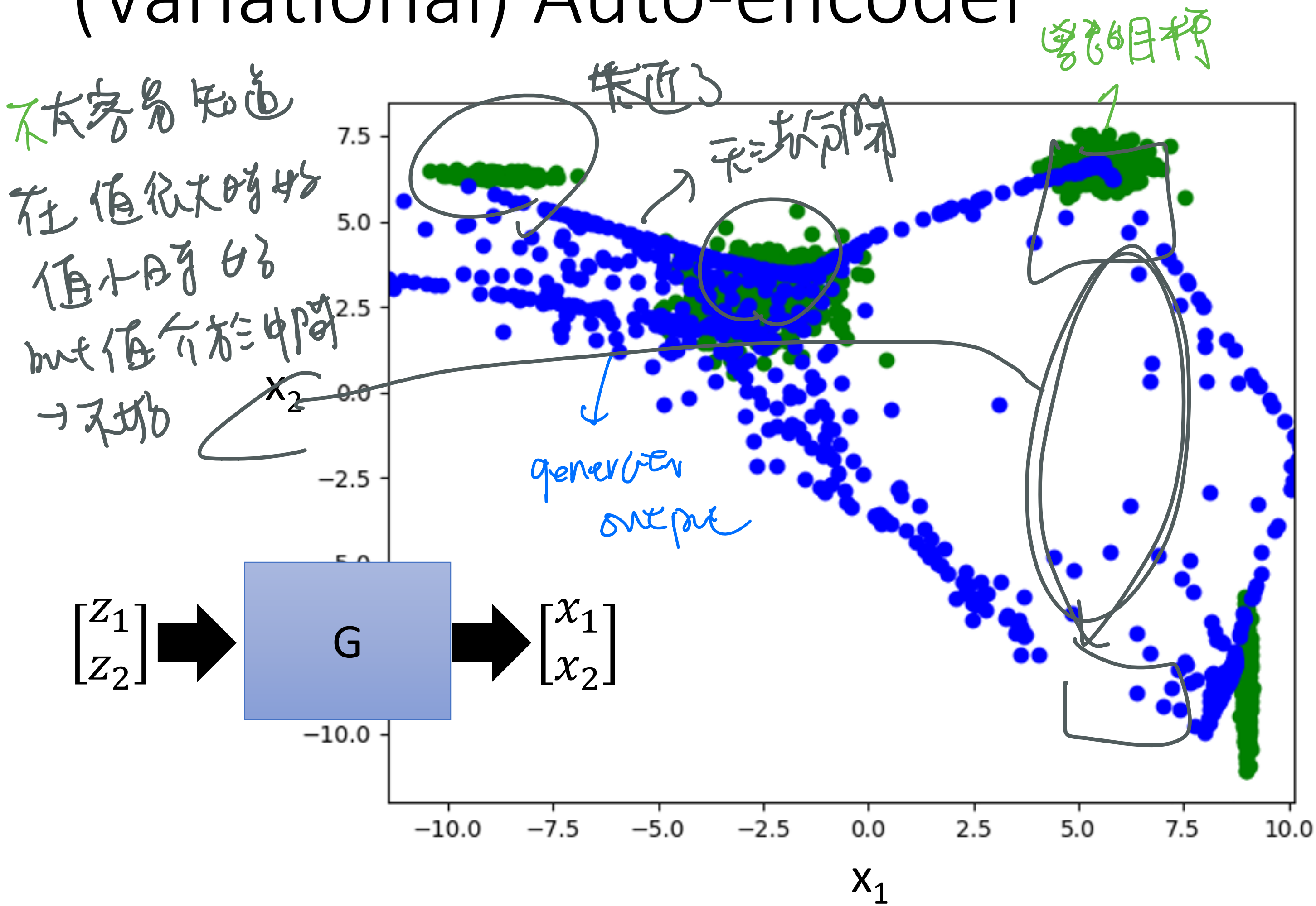
Although highly correlated, they cannot influence each other.

Need deep structure to catch the relation between components.

把 correlation 考慮進去 need deep network

感謝 黃淞楓 同學提供結果

(Variational) Auto-encoder



Outline

- Basic Idea of GAN
- GAN as structured learning
- Can Generator learn by itself?
- Can Discriminator generate?
- A little bit theory

Discriminator

不同的名字

Evaluation function, Potential Function, Energy Function …

- Discriminator is a function D (network, can deep)

$$D: X \rightarrow \mathrm{R}$$

→分数

  - Input x: an object x (e.g. an image)
  - Output D(x): scalar which represents how "good" an object x is

Can we use the discriminator to generate objects?

Yes.

discriminator 相較generator 有什麼優勢

Discriminator

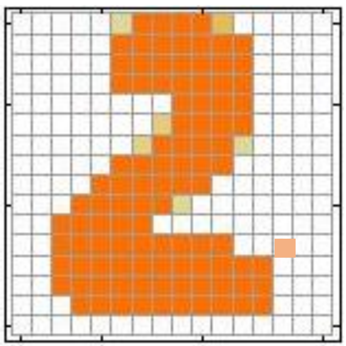
考慮 compare 之間的關係 較easy

一個一個component 獨立生成 → 考慮 compare 之間的關係

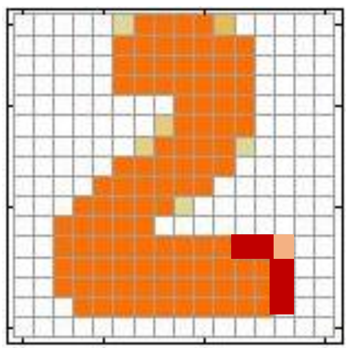
可能是個 convolution Network

- It is easier to catch the relation between the components by top-down evaluation. 比較單純的

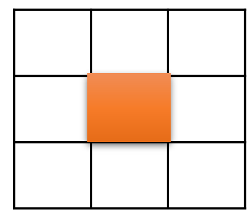
裡面有個 filter → 看去see 有沒有 獨立的filter ⇒ 低分



我覺得不行



我覺得其實 OK



⇒ 低分

This CNN filter is good enough.

Discriminator

可以input一个X see他是好or坏

=> 穷举所有可能的x 一个一个丢到discriminator中

- Suppose we already have a good discriminator D(x) …

see那个x discriminator会给他很高的分数

-> 最高分的那个x就是ANS

Inference

- Generate object $\tilde{x}$ that

$$\tilde{x} = \arg\max_{x \in X} D(x)$$

Enumerate all possible x !!!

痛苦

It is feasible ???

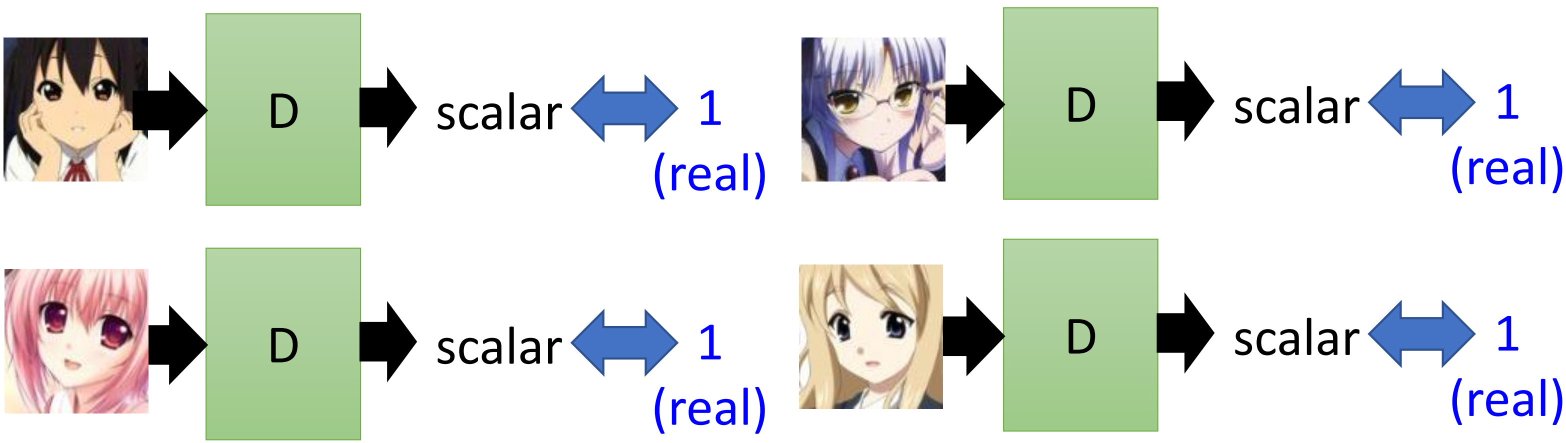
可能拿discrim 来做生成 东西很痛苦

How to learn the discriminator?

Discriminator - Training

- I have some real images

only have 正面的例子
沒看過假的

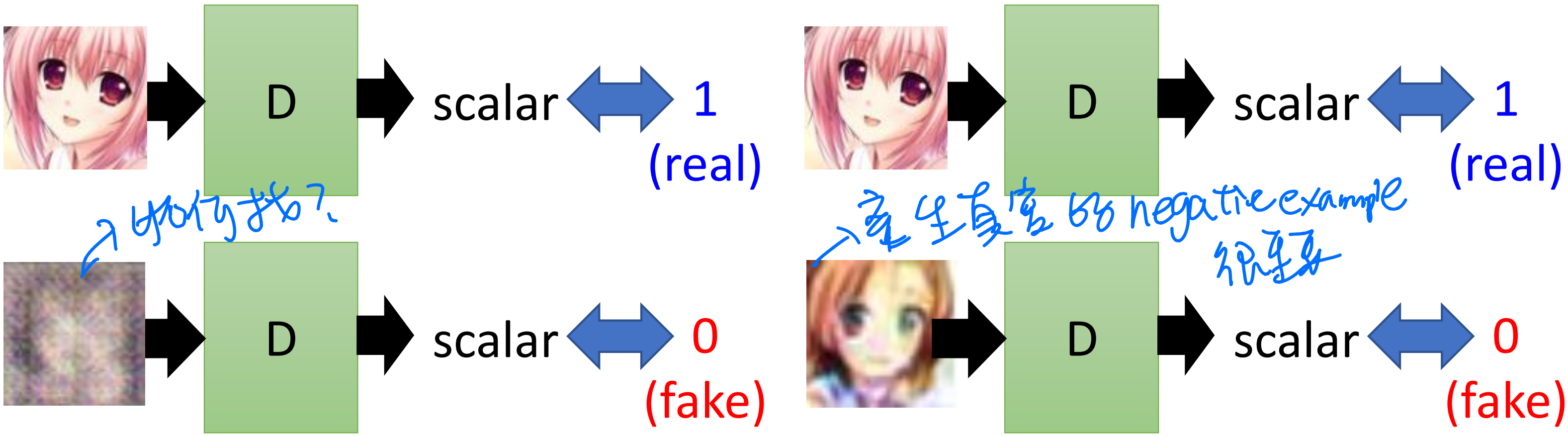


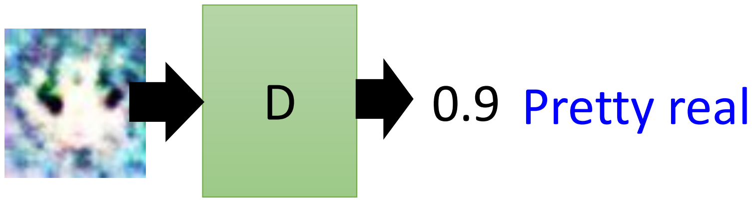
Discriminator only learns to output “1” (real).

Discriminator training needs some negative examples.

Discriminator - Training

- Negative examples are critical.



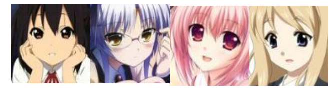


How to generate realistic negative examples?

Use iterative 来解

Discriminator - Training

- **General Algorithm**

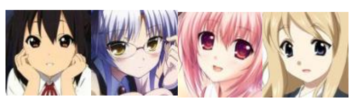


  - Given a set of positive examples, randomly generate a set of negative examples.

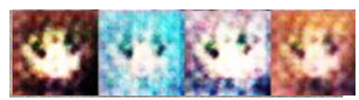
→ random sample 出来的 noise

  - In each iteration
    - Learn a discriminator D that can discriminate positive and negative examples.

重复做



v.s.



→ D

    - Generate negative examples by discriminator D

$$\tilde{x} = \arg\max_{x \in X} D(x)$$

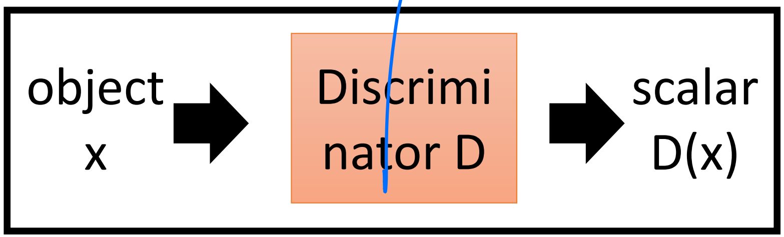
好的 image

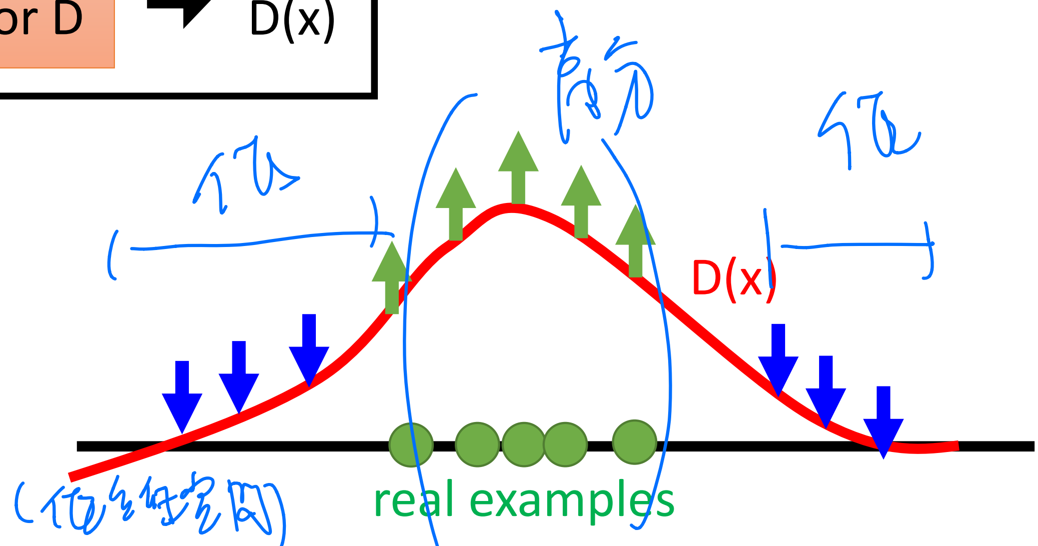
一定要解这个 才可以用

discr 做 generator

but in 高维空間 很難把高維中沒出現的 real example 的分數压低

Discriminator - Training

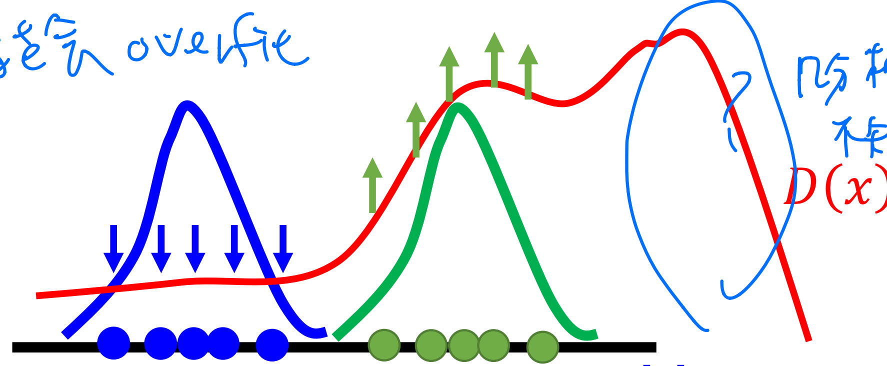




In practice, you cannot decrease all the x other than real examples. object 隨著分布的位置不同 discrim. 会給不同的分數

if data很少 可能是会 overfit

Discriminator - Training



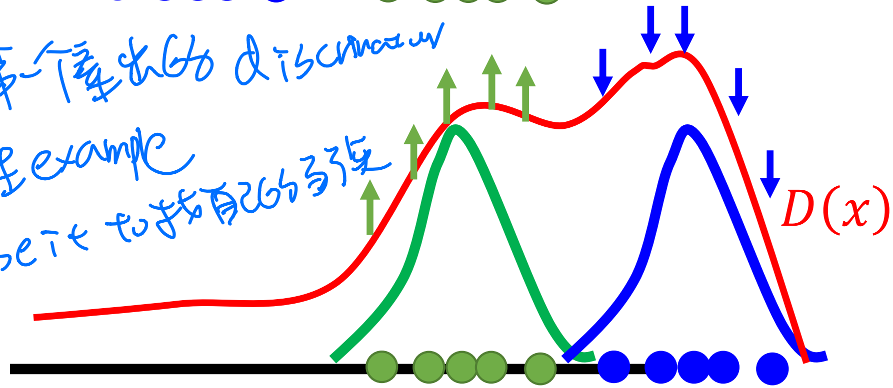
陷 机器 不知

● real

● generated

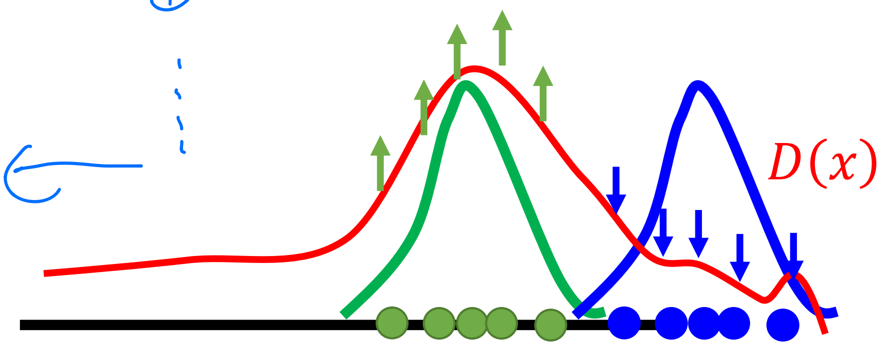
用第一个產出的 discriminator 產生example

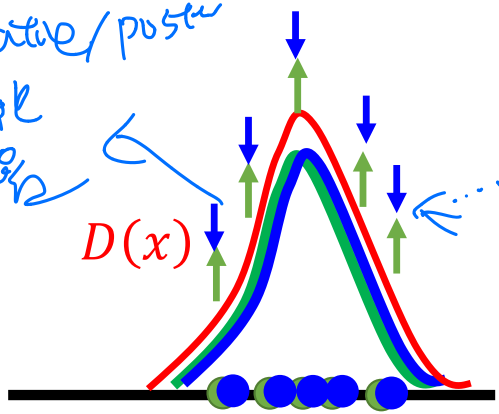
→ use it 去找自己的弱点



In the end ......

negative/positive example 不重要





剛剛講的 train discriminator process

Structured Learning

其實就是 Graphical Model 的training

general的 → Graphical Model

在不同的方法裡講法會有差不同

- Structured Perceptron
- Structured SVM
- Gibbs sampling
- Hidden information
- Application: sequence labelling, summarization

Bayesian Network (Directed Graph)

Markov Random Field (Undirected Graph)

上面有 Potential 函式 → 這個東西 discriminator

Energy-based Model: http://www.cs.nyu.edu/~yann/research/ebm/

Conditional Random Field

Markov Logic Network

Boltzmann Machine

輸入 observation graph 會告訴 data 產生出來的機率 = discriminator 的分數

Segmental CRF

(Only list some of the approaches)

Restricted Boltzmann Machine

都是 iterative 的去 train (train model → sample 出 negative → update model)

Generator v.s. Discriminator

- ***Generator*** 生成快
  - Pros:
    - Easy to generate even with deep model
  - Cons: → 只學到表象
    - Imitate the appearance
    - Hard to learn the correlation between components

- ***Discriminator***
  - Pros: → 考慮大局
    - Considering the big picture
  - Cons: generate 到底 解 Arg max problem 難
    - Generation is not always feasible 若 model need linear
      - Especially when your model is deep
    - How to do negative sampling?

→取代了Argmax的problem（在像dream会给奇怪的image）

Generator + Discriminator

- **General Algorithm**

  - Given a set of positive examples, randomly generate a set of negative examples.

  - In each iteration
    - Learn a discriminator D that can discriminate positive and negative examples.

 v.s.  → D

    - Generate negative examples by discriminator D

G → $\tilde{x}$ = $\tilde{x} = \arg\max_{x \in X} D(x)$

→可以產生奇怪的image

Benefit of GAN

- From Discriminator's point of view
  - Using generator to generate negative samples

$$\boxed{G \longrightarrow \tilde{x}} = \boxed{\tilde{x} = \arg\max_{x \in X} D(x)}$$

efficient ↳ loss 将是来自大局观的 discriminator

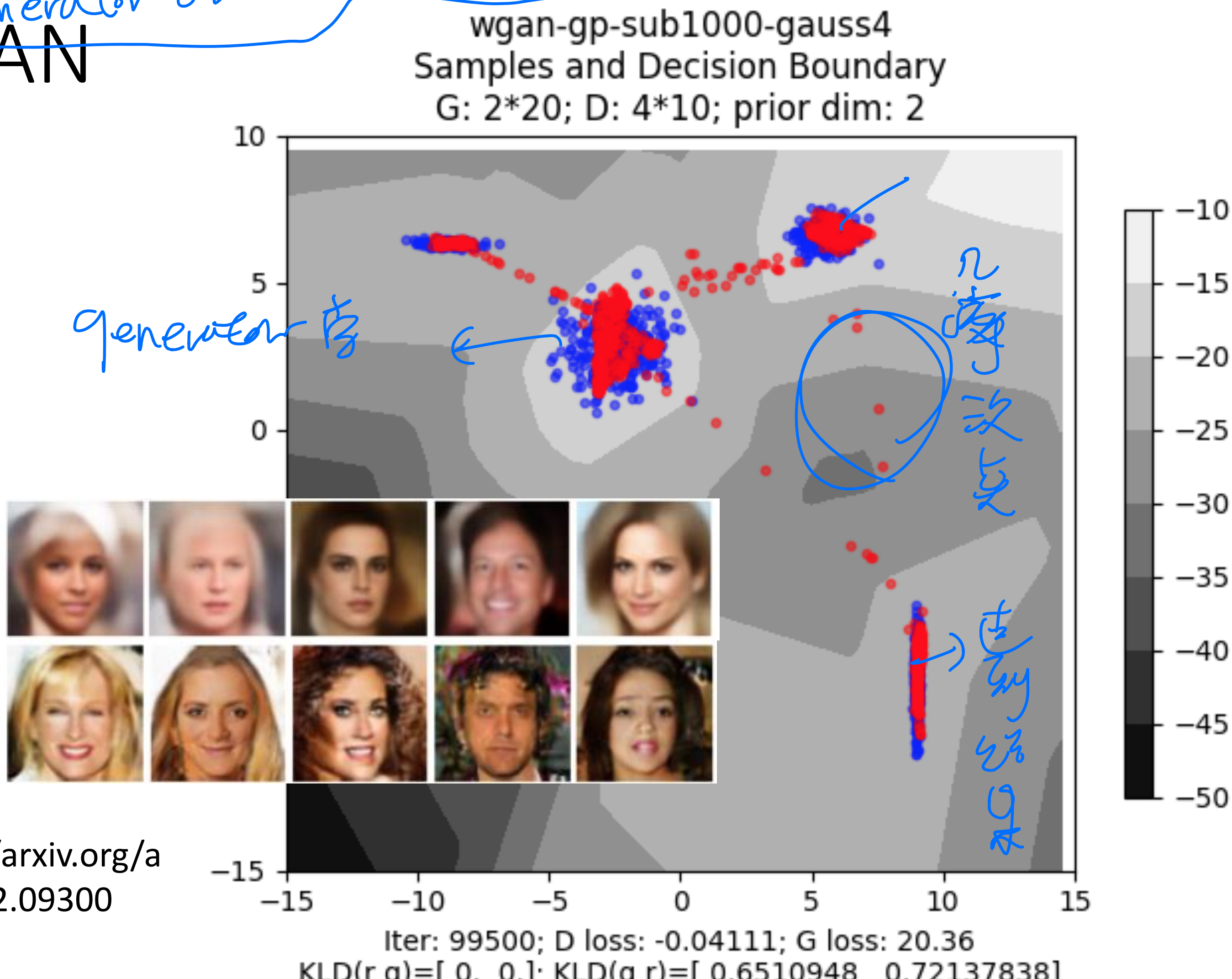
- From Generator's point of view
  - Still generate the object component-by-component
  - But it is learned from the discriminator with global view.

感謝 段逸林 同學提供結果

有discriminator的generator 比VAE的generator好多

GAN

generator 產



凡 沒 是

真的結果

VAE

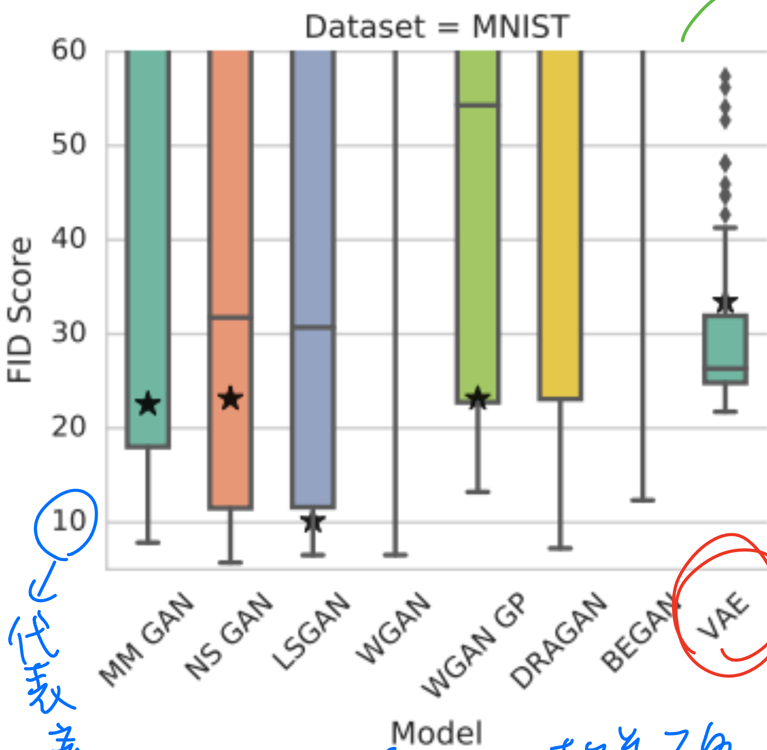
GAN

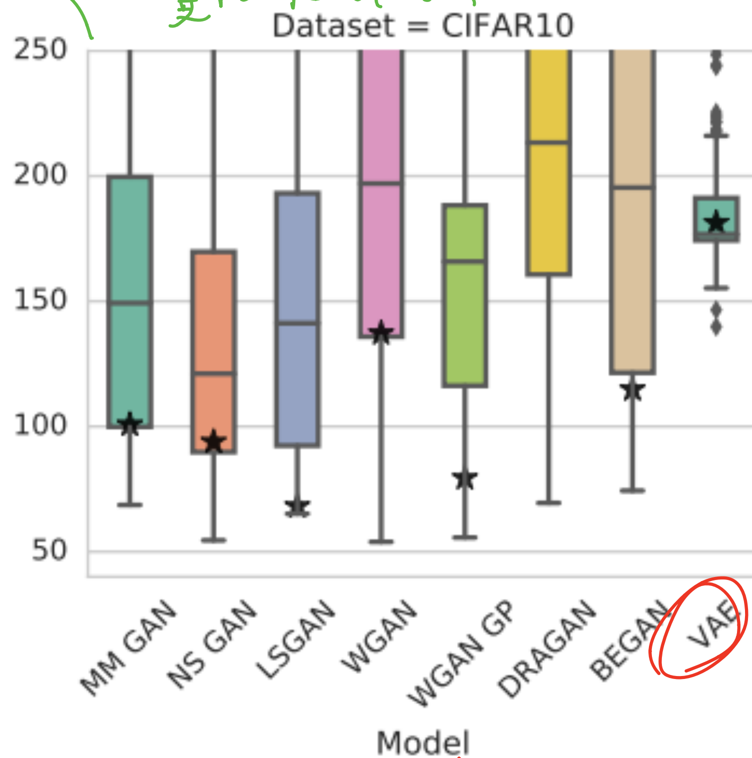
https://arxiv.org/abs/1512.09300

[Mario Lucic, et al. arXiv, 2017]

比较各种GAN的

不同方法之間差距没有什么不同
且range都很大





代表产生出来的图片越real

⇒ performance 都差不多

VAE和GAN有明显差别

FID[Martin Heusel, et al., NIPS, 2017]: Smaller is better

① VAE稳 (分数很集中)

② 最好结果=GAN

Next Time

- Preview
  - https://youtu.be/0CKeqXl5IY0
  - https://youtu.be/KSN4QYgAtao

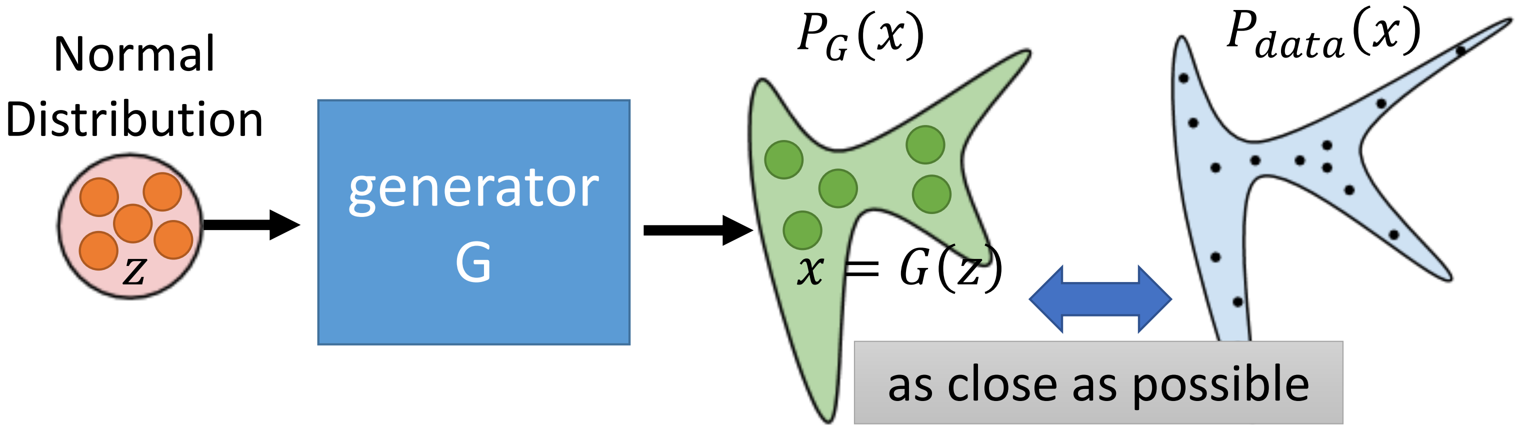
Outline

- Basic Idea of GAN
- GAN as structured learning
- Can Generator learn by itself?
- Can Discriminator generate?
- A little bit theory

Generator

x: an image (a high-dimensional vector)

- A generator G is a network. The network defines a probability distribution P_G



$$G^* = arg \min_G \underline{Div(P_G, P_{data})}$$

Divergence between distributions P_G and P_{data}

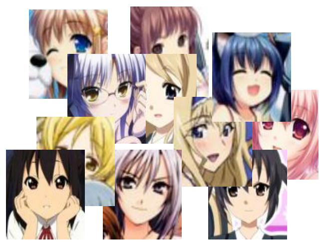
How to compute the divergence?

Discriminator

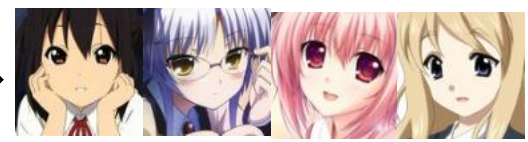
$$G^* = arg \min_G Div(P_G, P_{data})$$

Although we do not know the distributions of P_G and P_{data}, we can sample from them.

Database



sample



Sampling from P_{data}

sample from normal

vector

vector

vector

vector

G

Sampling from P_G

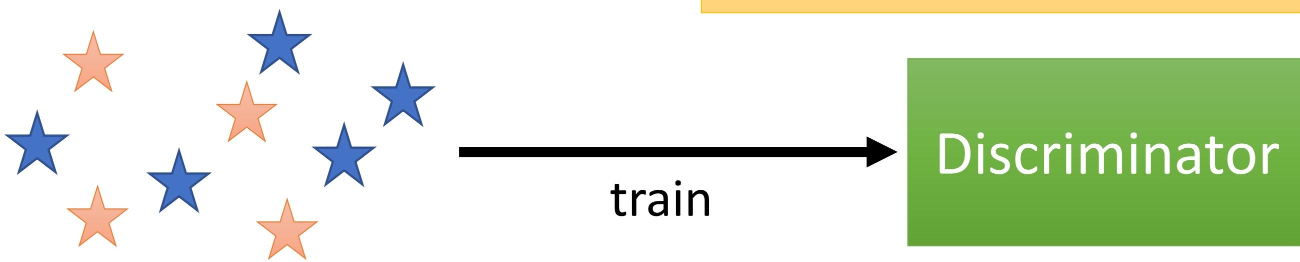
Discriminator

$$G^* = arg \min_G Div(P_G, P_{data})$$

: data sampled from P_{data}

: data sampled from P_G

Using the example objective function is exactly the same as training a binary classifier.



Example Objective Function for D

$$V(G,D) = E_{x\sim P_{data}}[logD(x)] + E_{x\sim P_G}\left[log\left(1 - D(x)\right)\right]$$

(G is fixed)

Training: $D^* = arg \boxed{\max_D V(D,G)}$

The maximum objective value is related to JS divergence.

Discriminator

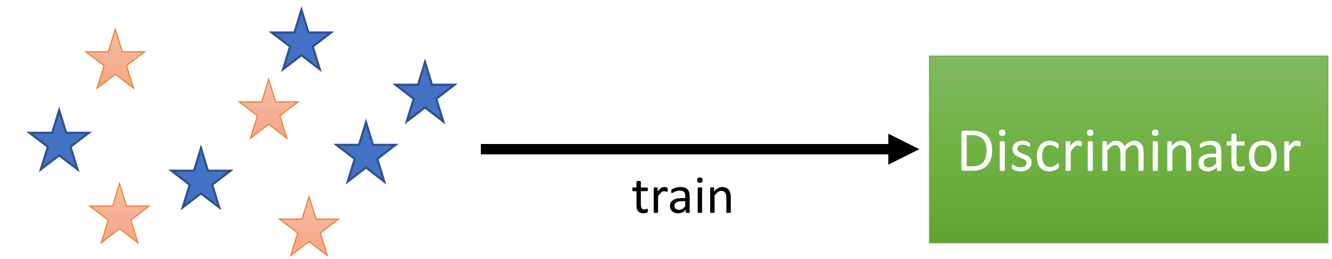
$$G^* = arg\min_G Div(P_G, P_{data})$$

: data sampled from P_{data}

: data sampled from P_G

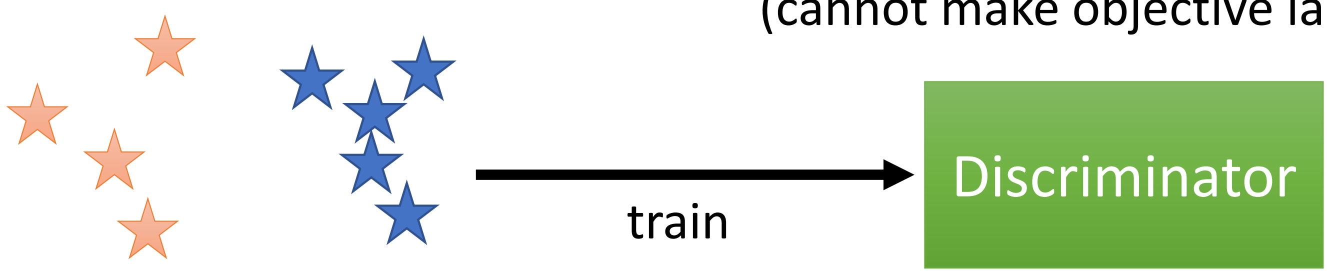
Training:

$$D^* = arg\boxed{\max_D V(D,G)}$$



small divergence

hard to discriminate
(cannot make objective large)



large divergence

easy to discriminate

[Goodfellow, et al., NIPS, 2014]

$$G^* = arg \min_G \boxed{\max_D V(G, D)}$$

$$D^* = arg \boxed{\max_D V(D, G)}$$

The maximum objective value is related to JS divergence.

- Initialize generator and discriminator
- In each training iteration:

  Step 1: Fix generator G, and update discriminator D

  Step 2: Fix discriminator D, and update generator G

Can we use other divergence?

Name	$D_f(P\|Q)$	Generator $f(u)$
Total variation	$\frac{1}{2}\int \lvert p(x)-q(x)\rvert \,\mathrm{d}x$	$\frac{1}{2}\lvert u-1\rvert$
Kullback-Leibler	$\int p(x)\log\frac{p(x)}{q(x)}\,\mathrm{d}x$	$u\log u$
Reverse Kullback-Leibler	$\int q(x)\log\frac{q(x)}{p(x)}\,\mathrm{d}x$	$-\log u$
Pearson χ^2	$\int \frac{(q(x)-p(x))^2}{p(x)}\,\mathrm{d}x$	$(u-1)^2$
Neyman χ^2	$\int \frac{(p(x)-q(x))^2}{q(x)}\,\mathrm{d}x$	$\frac{(1-u)^2}{u}$
Squared Hellinger	$\int \left(\sqrt{p(x)}-\sqrt{q(x)}\right)^2\,\mathrm{d}x$	$\left(\sqrt{u}-1\right)^2$
Jeffrey	$\int (p(x)-q(x))\log\left(\frac{p(x)}{q(x)}\right)\,\mathrm{d}x$	$(u-1)\log u$
Jensen-Shannon	$\frac{1}{2}\int p(x)\log\frac{2p(x)}{p(x)+q(x)}+q(x)\log\frac{2q(x)}{p(x)+q(x)}\,\mathrm{d}x$	$-(u+1)\log\frac{1+u}{2}+u\log u$
Jensen-Shannon-weighted	$\int p(x)\pi\log\frac{p(x)}{\pi p(x)+(1-\pi)q(x)}+(1-\pi)q(x)\log\frac{q(x)}{\pi p(x)+(1-\pi)q(x)}\,\mathrm{d}x$	$\pi u\log u-(1-\pi+\pi u)\log(1-\pi+\pi u)$
GAN	$\int p(x)\log\frac{2p(x)}{p(x)+q(x)}+q(x)\log\frac{2q(x)}{p(x)+q(x)}\,\mathrm{d}x-\log(4)$	$u\log u-(u+1)\log(u+1)$

Using the divergence you like ☺

Name	Conjugate $f^*(t)$
Total variation	t
Kullback-Leibler (KL)	$\exp(t-1)$
Reverse KL	$-1-\log(-t)$
Pearson χ^2	$\frac{1}{4}t^2+t$
Neyman χ^2	$2-2\sqrt{1-t}$
Squared Hellinger	$\frac{t}{1-t}$
Jeffrey	$W(e^{1-t})+\frac{1}{W(e^{1-t})}+t-2$
Jensen-Shannon	$-\log(2-\exp(t))$
Jensen-Shannon-weighted	$(1-\pi)\log\frac{1-\pi}{1-\pi e^{t/\pi}}$
GAN	$-\log(1-\exp(t))$