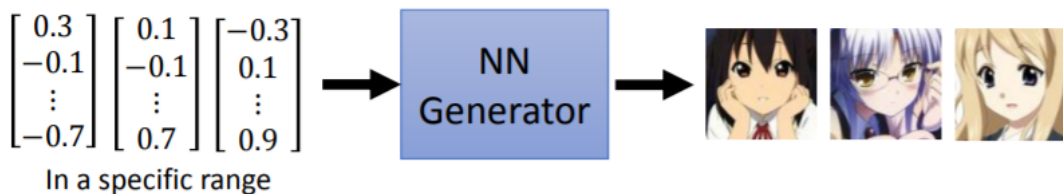


Introduction to GAN

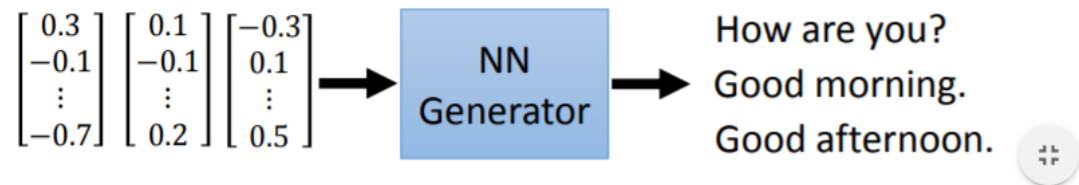
Basic Idea of GAN

GAN想要达到的目的是让机器生成某样东西（图片，句子等），所以要训练出一个Generator，比如下图中，将Vector输入到训练好的Generator当中，如果是图像生成，就会生成不同的图片；如果是文本生成，就会得到不同的句子。

Image Generation



Sentence Generation

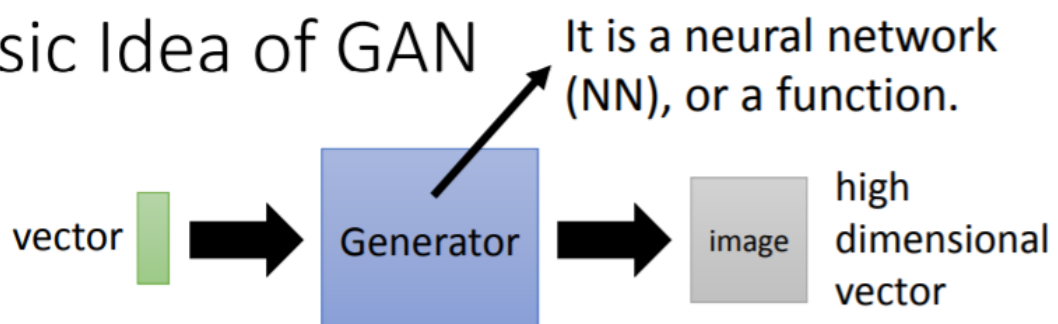


上述的输入一个随机的Vector得到生成的图片或者文字可能看起来没有什么用处，真正有用的是Conditional Generation，也就是输入一些条件，比如输入图片让机器产生对应的文字；或者输入文字让机器产生对应的图片。也就是输入了解的东西，然后让机器产生对应的内容。

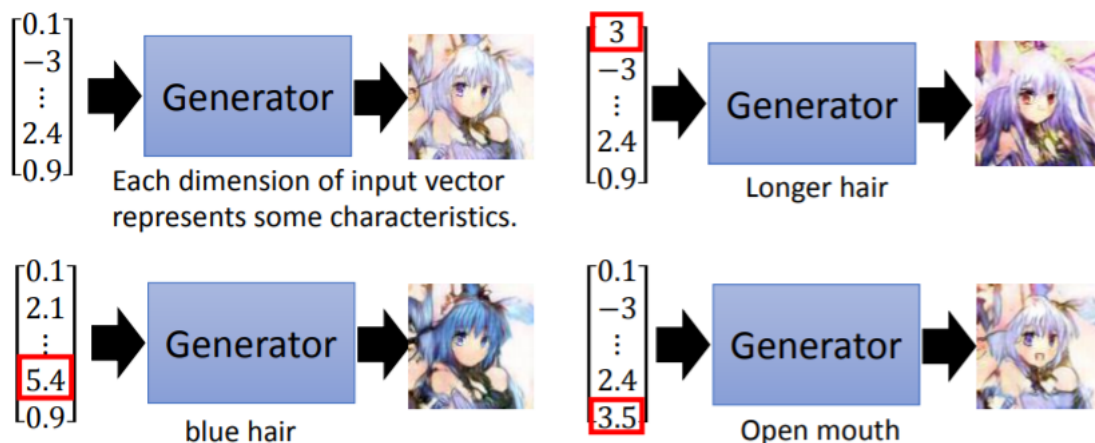
Generator (生成器)

Generator是一个Neural network,或者是一个function。它接受一个低维的向量，然后输出高维向量作为结果，这个高维向量就是我们需要的输出。比如对于图像生成，就是一张图片。

Basic Idea of GAN

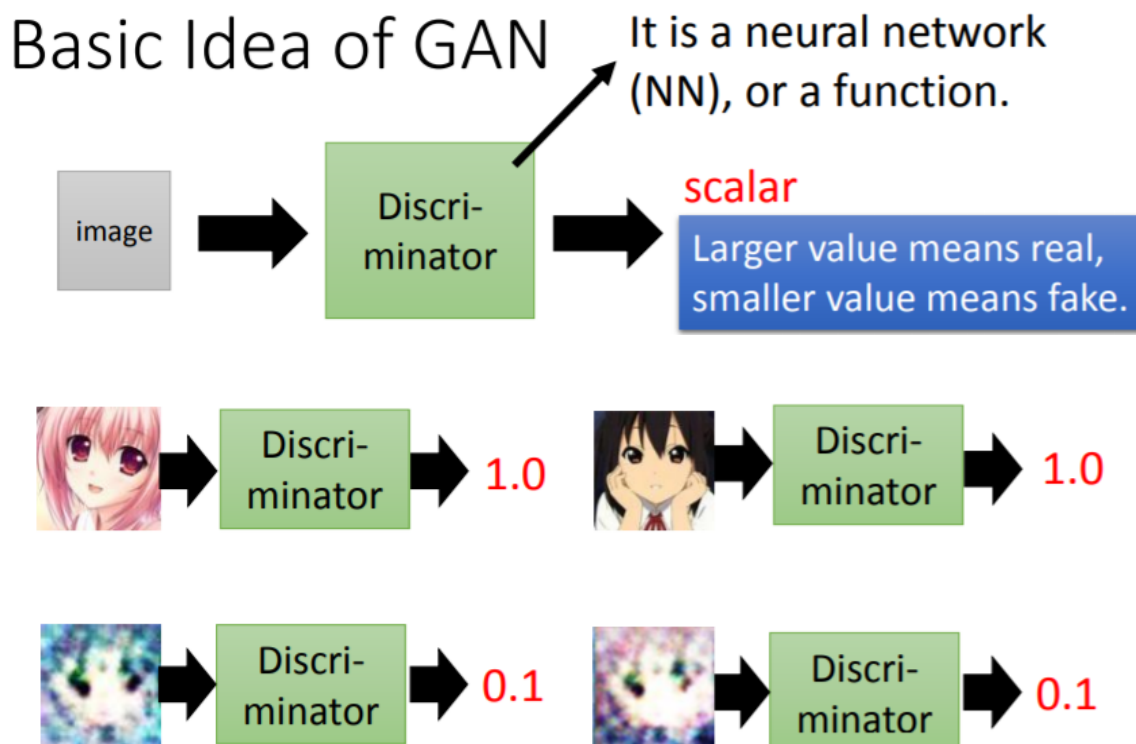


更具体的，以生成动漫头像为例。输入向量的每一个维度都对应图片的某种特征。比如第一维代表头发长度；倒数第二维代表头发蓝色的程度；最后一个维度代表嘴巴大小。调整不同的维度，得到结果所对应的特征也会变化。



Discriminator(判别器)

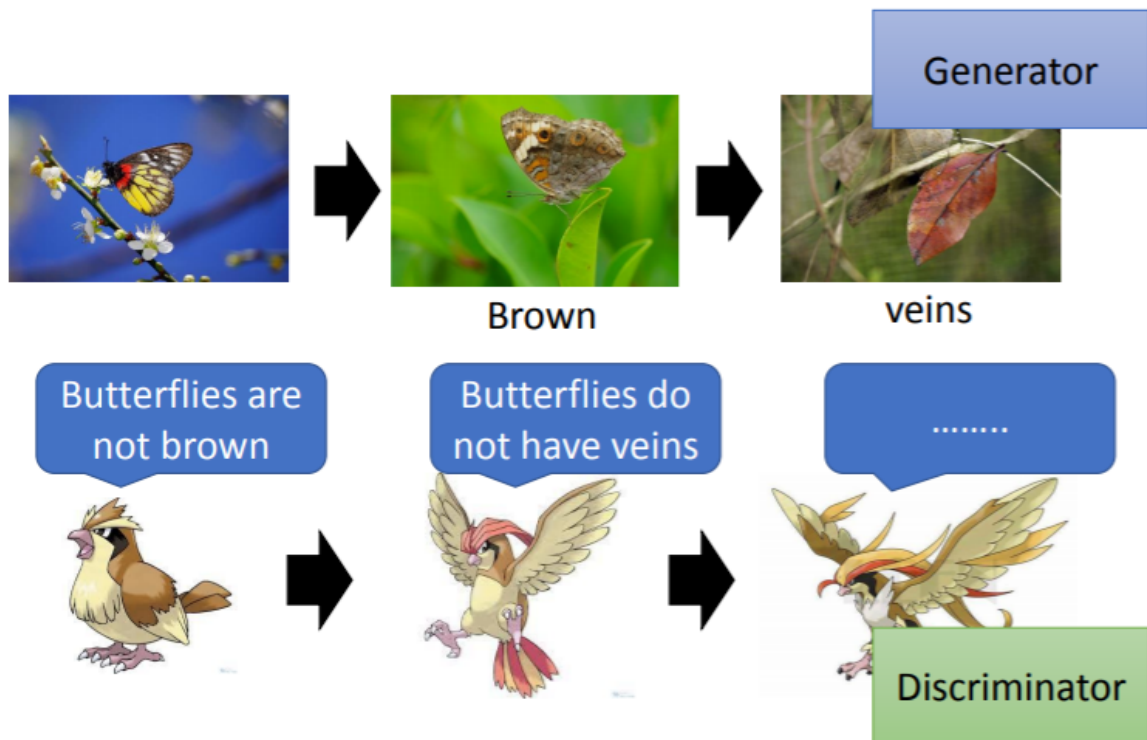
GAN在训练Generator的同时会训练Discriminator，它也是一个神经网络或者一个函数，以图片为输入（假设是图片任务），输出一个数值（scalar），代表产生出来的图片（输入）的质量。数值越大，表面输入图片的结果越好（越真实）。



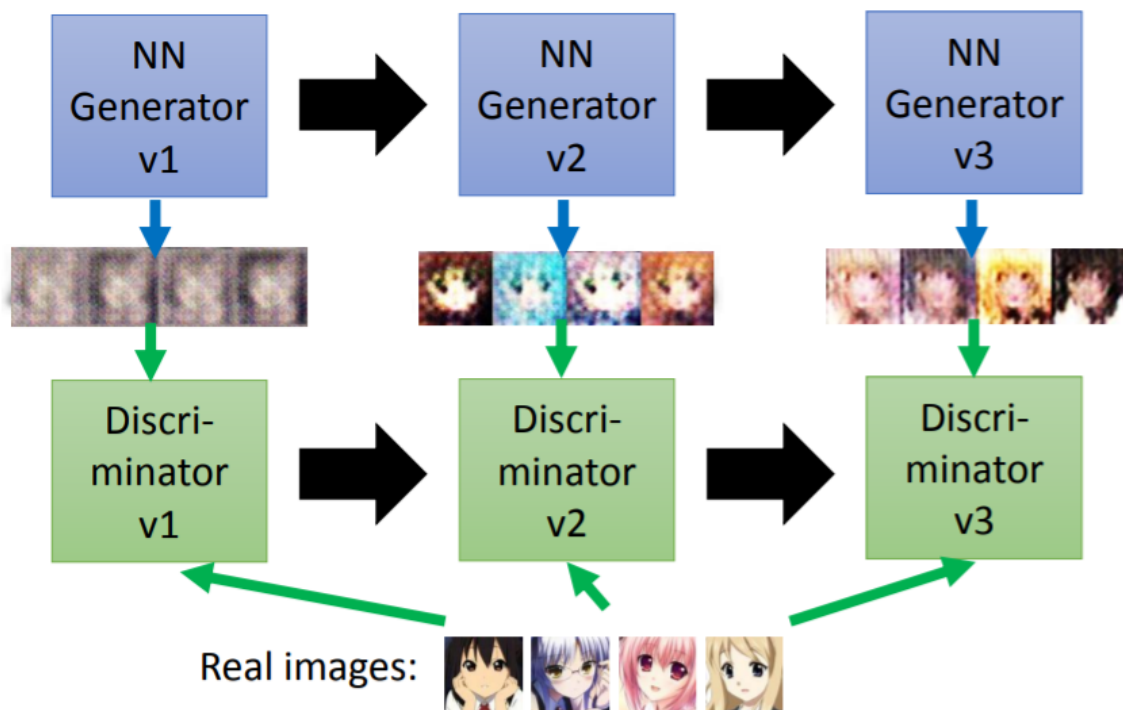
Relation Between Generator and Discriminator

在课程当中李老师以枯叶蝶和波波（假设波波会吃枯叶蝶）为例，说明生成器和判别器是“敌对”关系。生成器就像枯叶蝶，在与天敌的对抗过程中不断进化来隐藏自己。判别器就像波波，不断进化来增强对枯叶蝶的识别能力。所以二者都在对抗中越来越强。

Basic Idea of GAN



下图就是生成器和判别器之间的对抗。生成器V1生成图片，然后判别器V1根据真实图片，分辨生成器产生的图片，然后生成器V2进化产生更接近真实图片的结果，同时判别器也在进化。二者不断对抗，能力越来越强，生成器生成的图片也越来越接近于真实图片。这就是GAN中Adversarial（对抗）的由来。



如果换一个比较和平的比喻，生成器和判别器就像学生和老师，学生在老师指导下不断进步。

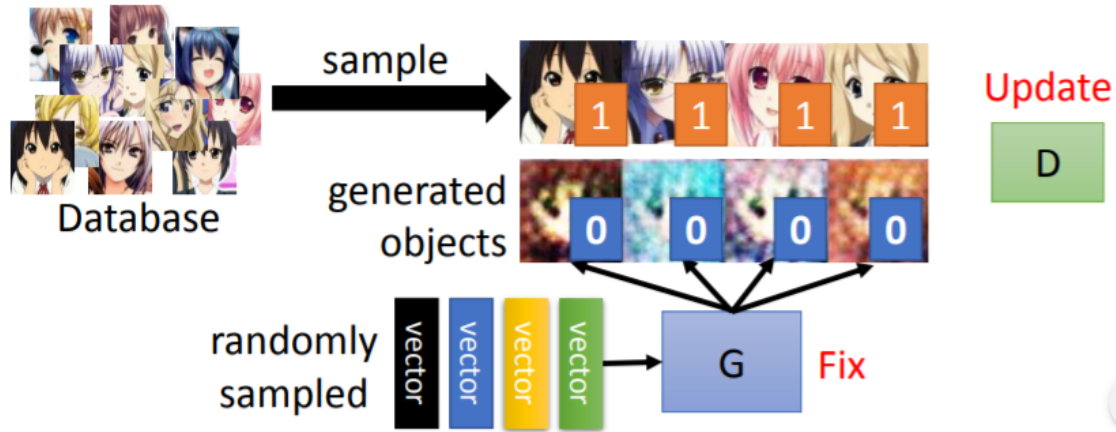
Algorithm

1.初始化生成器和判别器的参数

2.在每次迭代当中:

(1) 固定生成器参数, 训练判别器: 判别器学会给真实图片高分, 而给生成器生成的图片低分;

Step 1: Fix generator G, and update discriminator D

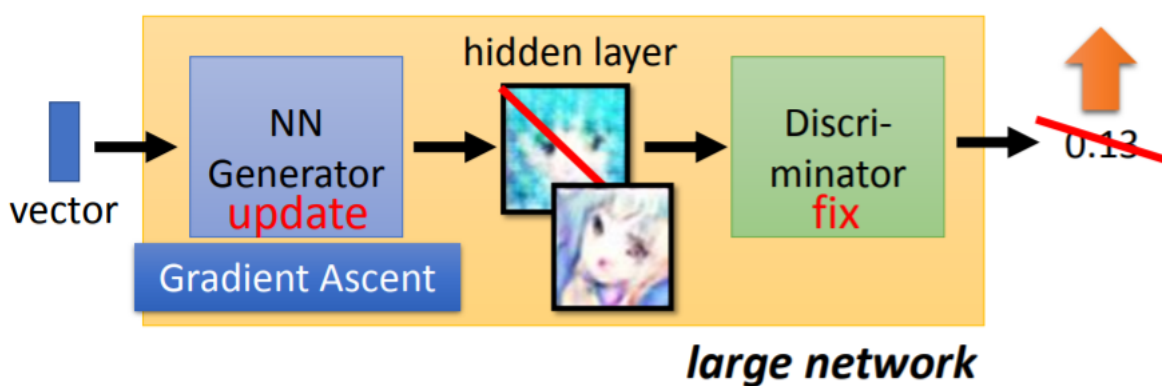


Discriminator learns to assign high scores to real objects and low scores to generated objects.

(2) 固定判别器的参数, 训练生成器: 让生成器学会“骗过”判别器, 生成的结果尽量给高分, 于是为了得到高分, 生成器的生成结果会越来越接近天生的高分结果——真实图片。

Step 2: Fix discriminator D, and update generator G

Generator learns to “fool” the discriminator



实际操作的时候会把生成器和判别器连接在一起当作一个大的网络, 其中有一个很宽的隐藏层是生成器的图片。训练的时候分别固定对应的参数, 调整其他的参数。

形式化表示:

Algorithm Initialize θ_d for D and θ_g for G

- In each training iteration:

Learning
D

- Sample m examples $\{x^1, x^2, \dots, x^m\}$ from database
- Sample m noise samples $\{z^1, z^2, \dots, z^m\}$ from a distribution
- Obtaining generated data $\{\tilde{x}^1, \tilde{x}^2, \dots, \tilde{x}^m\}, \tilde{x}^i = G(z^i)$
- Update discriminator parameters θ_d to maximize
 - $\tilde{V} = \frac{1}{m} \sum_{i=1}^m \log D(x^i) + \frac{1}{m} \sum_{i=1}^m \log (1 - D(\tilde{x}^i))$
 - $\theta_d \leftarrow \theta_d + \eta \nabla \tilde{V}(\theta_d)$

Learning
G

- Sample m noise samples $\{z^1, z^2, \dots, z^m\}$ from a distribution
- Update generator parameters θ_g to maximize
 - $\tilde{V} = \frac{1}{m} \sum_{i=1}^m \log (D(G(z^i)))$
 - $\theta_g \leftarrow \theta_g - \eta \nabla \tilde{V}(\theta_g)$

learning D中的函数仅为一个范例。前半部分意思是真实图片的得分取log求平均，这一部分越大越好（真实图片得分越高越好）。后半部分是1-生成图片得分取log求平均，这一部分越大越好，也就是生成图片得分越低越好。总的来看，maximize目标函数的目标就是使判别器“进化”，真实得分越大越好，生成得分越低越好。

GAN as Structured Learning

Structured Learning

结构化学习的输入和输出都是序列、列表、树、矩阵等等模型，下图是一些结构化学习的例子。

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)

结构化学习是一个特别具有挑战性的问题，首先结构化学习可以看作One-shot/Zero-shot Learning。假设有一个语句翻译的模型，输入和输出都是句子，很可能所有样本里没有重复的句子。如果我们把翻译的每个结果视为一个分类，可能每一个分类的样本就出现一次。如果输出的分类很多，有一些分类甚至可能没有训练样本。所以要让模型处理从来没有见过的句子是一个难题。

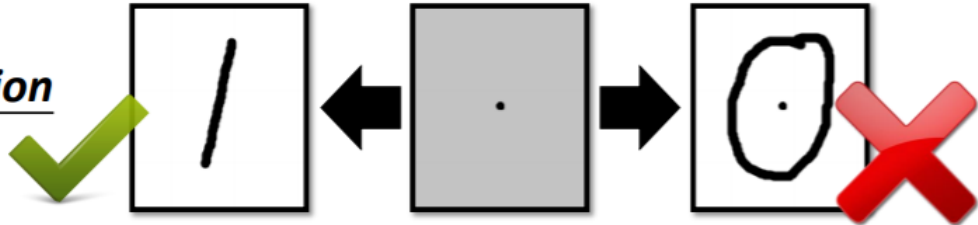
• One-shot/Zero-shot Learning:

- In classification, each class has some examples.
- In structured learning,
 - If you consider each possible output as a “class”
 - Since the output space is huge, most “classes” do not have any training data.
 - Machine has to create new stuff during testing.
 - Need more intelligence


其次机器必须要有整体规划（“大局观”），以图像生成为例，生成一系列的pixels，但是pixels之间要存在关系，必须要能够组成一张合理图片，要从全局上考虑。


- 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.

Image
Generation



Sentence
Generation

這個婆娘不是人 

九天玄女下凡塵 

Structured Learning Approach

在结构化学习中有两种方法：自底向上和自顶向下，自底向上是从组成元素的级别来生成目标，缺点是：很容易失去“大局观”。自顶向下即从整体上来评估序列，找到最优结果，缺点是这个方法很难进行生成。

在GAN中，自底向上的思路和Generator一致，自顶向下的思路和Discriminator一致。

Generator

Learn to generate the object at the component level

Bottom
Up



Discriminator

Evaluating the whole object, and find the best one

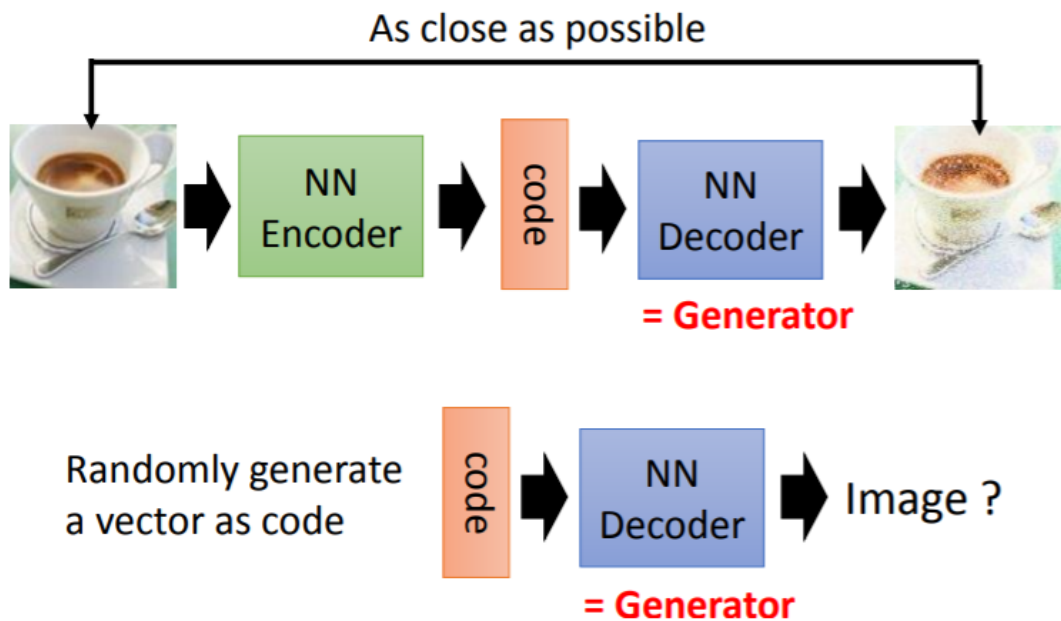
Top
Down

Can Generator learn by itself?

生成器其实是可以自己学习的，但是问题在于输入的向量如何获取。为此可以再通过学习得到一个Encoder，输入图片可以得到对应的向量。

在Auto-encoder当中，Decoder其实就相当于GAN当中的生成器。

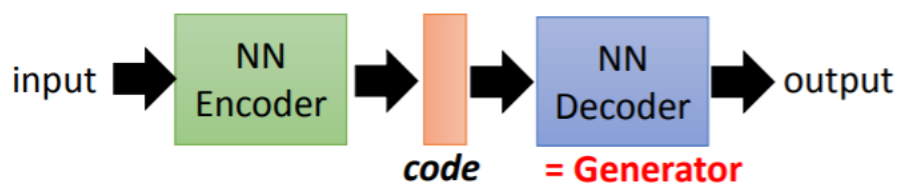
Auto-encoder



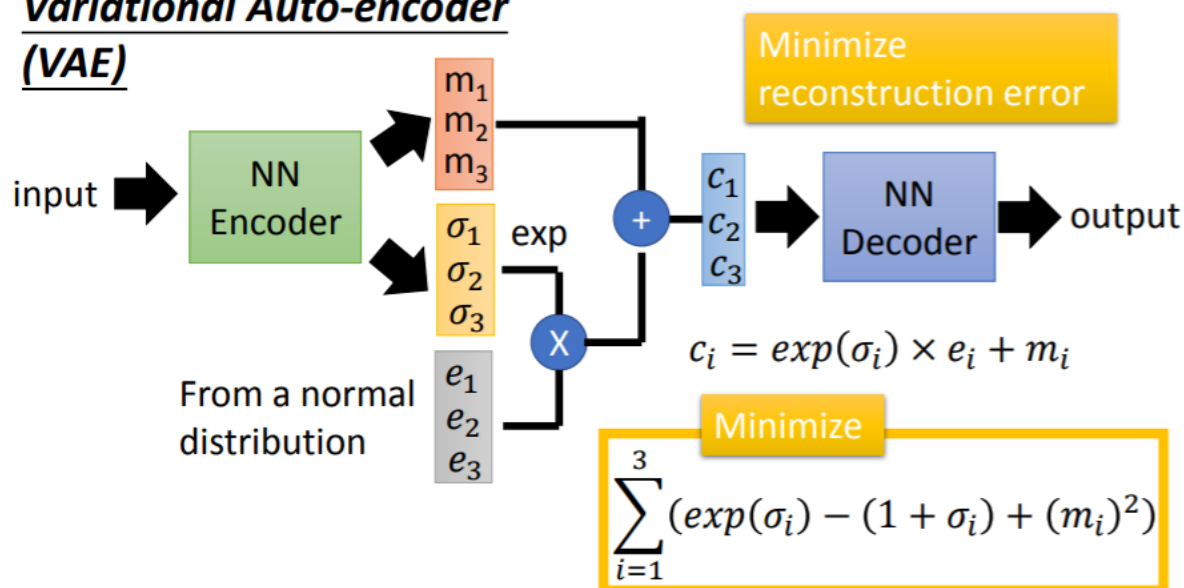
以下的部分没有完全听懂，只记录部分。

VAE模型的提出是为了让Decoder更加稳定。

Auto-encoder

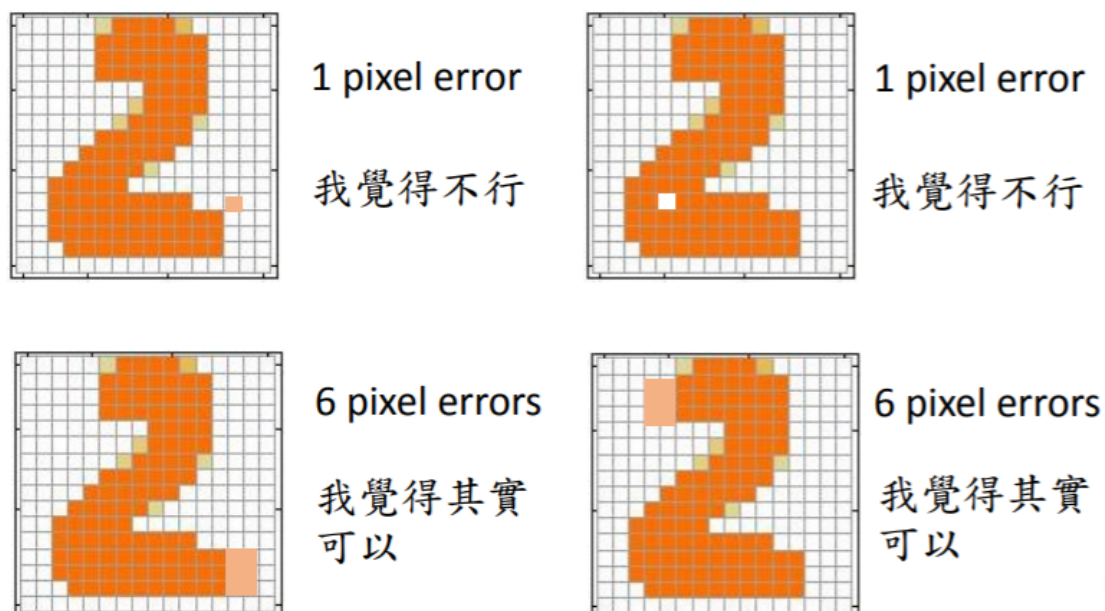


Variational Auto-encoder (VAE)

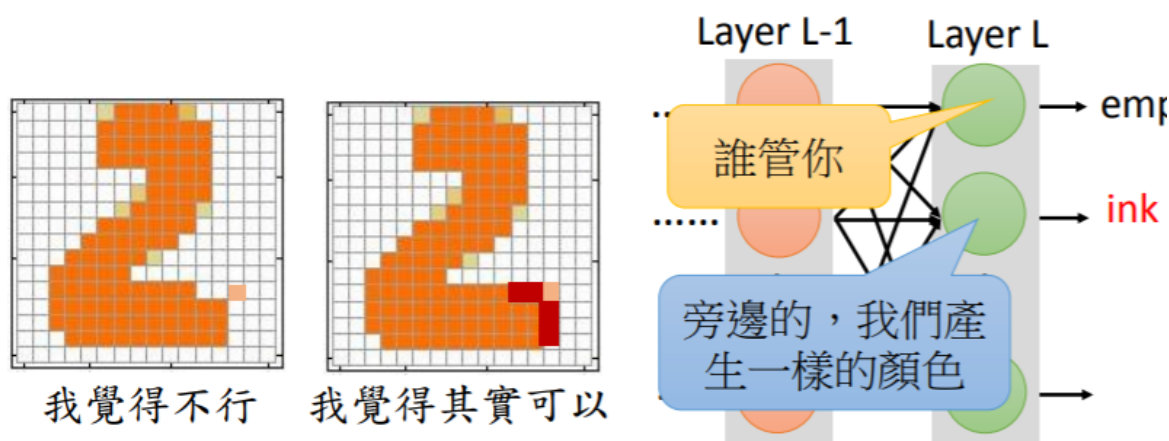


auto-encoder中decoder的目标是，输出和目标越相似越好，衡量的指标通常是通过像素之间的距离计算。而decoder在学习过程中不能完全cover target，所以会做出一些妥协。

如下图，按照decoder来说，上面一行更可能是产生的结果，但按照人类的想法，下边一行虽然差异更大，但效果是更好的。所以单纯地让output和target越像越好不可取。



本身多一个pixel没有错，但是是相邻的pixels空缺，因此在structure learning里面，component间的correlation很重要；单纯学一个generator困难在于train一个network的时候很难将component和component间的correlation直接考虑进去，虽然可以通过增大网络的规模来实现差不多的效果。



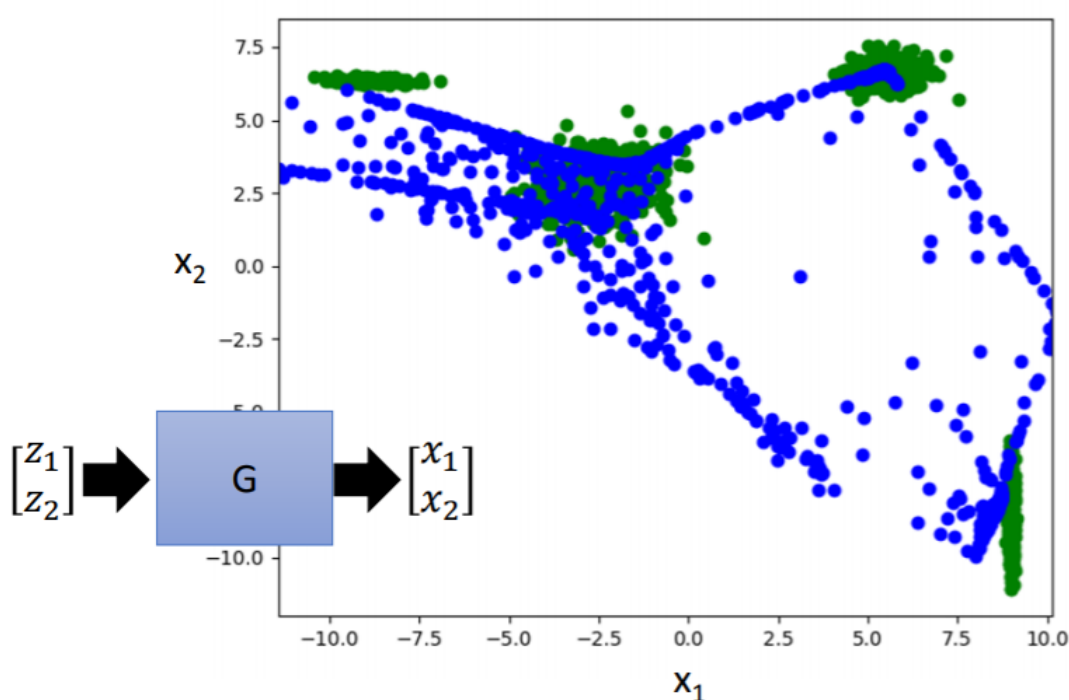
The relation between the components are critical.

Although highly correlated, they cannot influence each other

Need deep structure to catch the relation between components.

如下图，蓝色是生成器学习的输出，绿色是真实分布，VAE/AE难以意识到component之间的关系。比如“X1,X2都很大或者都很小时较好，中间较差”，所以输出在中间的也有很多值。

(Variational) Auto-encoder



Can Discriminator generate?

判别器在不同的文献中也被叫做Evaluation function, Potential function, Energy function. 其实判别器是可以生成的。判别器相较于生成器的优势是：生成器独立生成每个component，比较难衡量correlation，而判别器的输入是完整的图片，在这个基础上去catch components之间的correlation相对比较容易。

生成方法：穷举所有X，输入到判别器当中是否得到高分。

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

Inference

- Generate object \tilde{x} that

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

Enumerate all possible x !!!

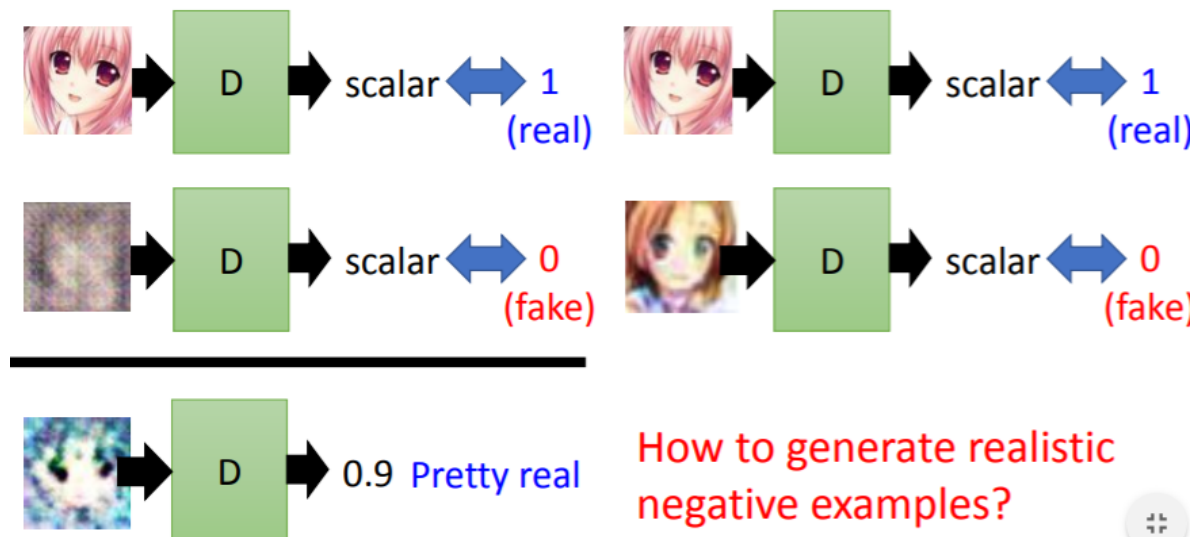
It is feasible ???

How to learn the discriminator?

问题是我们的数据只有positive example (好的图像)，用这种数据学习的判别器只会给出高分。所以要想办法生成negative example,但是这样的数据也是有要求的。如果negative example非常的差（显然不是正常图像的，比如一堆噪声），那么通过这样的学习，如果有一个相对较好，但依然很差的样本进行测试，判别器还是会给出高分。（下图左列）

所以需要相对更加真实的Negative Example，但是问题是这种数据如何产生。所以就变成了鸡生蛋蛋生鸡的问题。

- Negative examples are critical.



训练算法：每一次迭代都用上一次产生的Negative Example

• General Algorithm

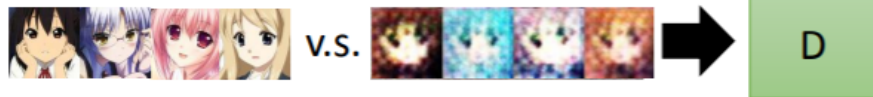


- Given a set of **positive examples**, randomly generate a set of **negative examples**.

- In each iteration



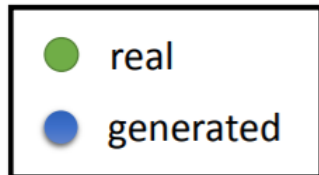
- Learn a discriminator D that can discriminate positive and negative examples.



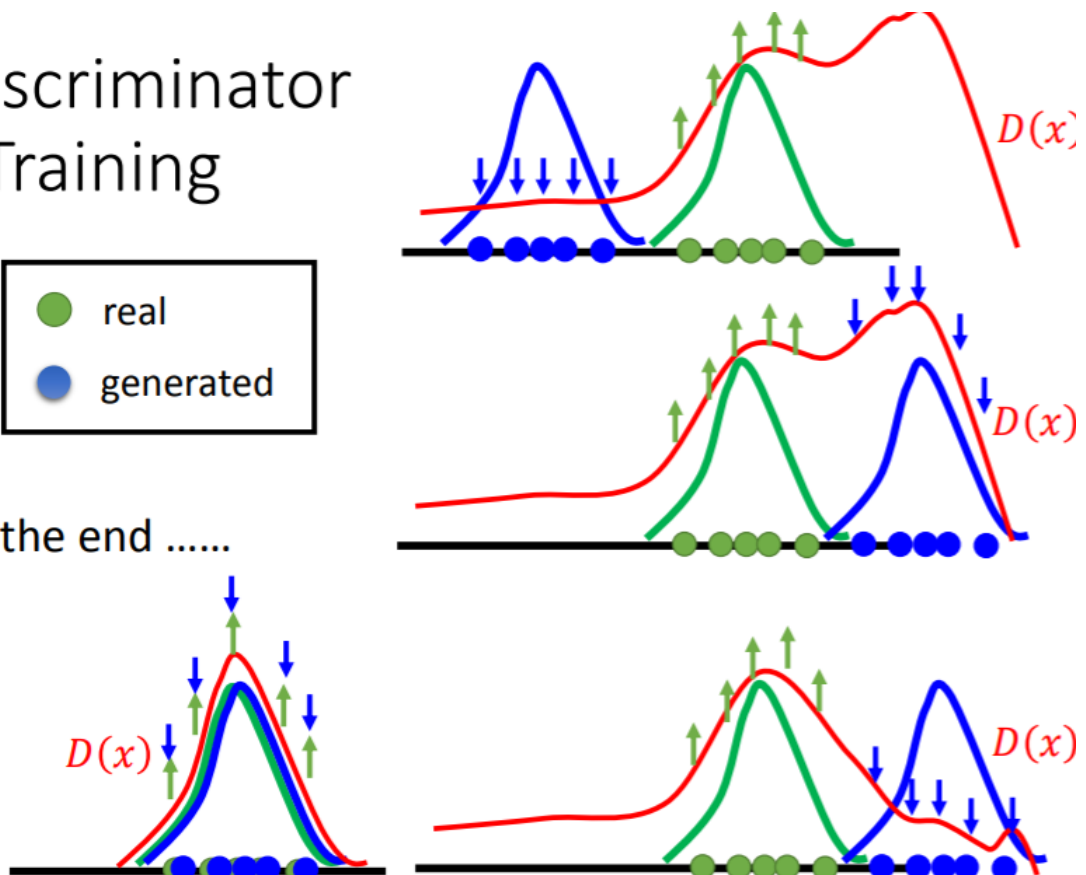
- Generate negative examples by discriminator D

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

Discriminator - Training



In the end



生成器和判别器对比:

Generator v.s. Discriminator

- **Generator**

- Pros:
 - Easy to generate even with deep model
- Cons:
 - Imitate the appearance
 - Hard to learn the correlation between components

- **Discriminator**

- Pros:
 - Considering the big picture
- Cons:
 - Generation is not always feasible
 - Especially when your model is deep
 - How to do negative sampling?

GAN的优势:

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

$$\boxed{\begin{array}{c} \text{G} \end{array} \rightarrow \tilde{x}} = \boxed{\tilde{x} = \arg \max_{x \in X} D(x)}$$

efficient

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

