GAN

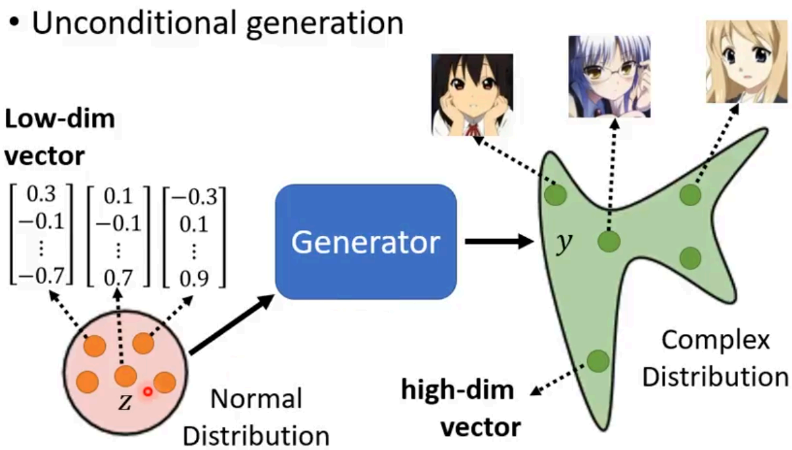
Network as Generator

- when the same input has different outputs
  - e.g. drawing, chatbot...
- input: a simple distribution + (condition information)
- output: a complex distribution

GAN, Generative adversarial network, add a discriminator (another network) to measure how "realistic" the output of generator is. Therefore, the network is self-supervised.
[All kinds of GAN]

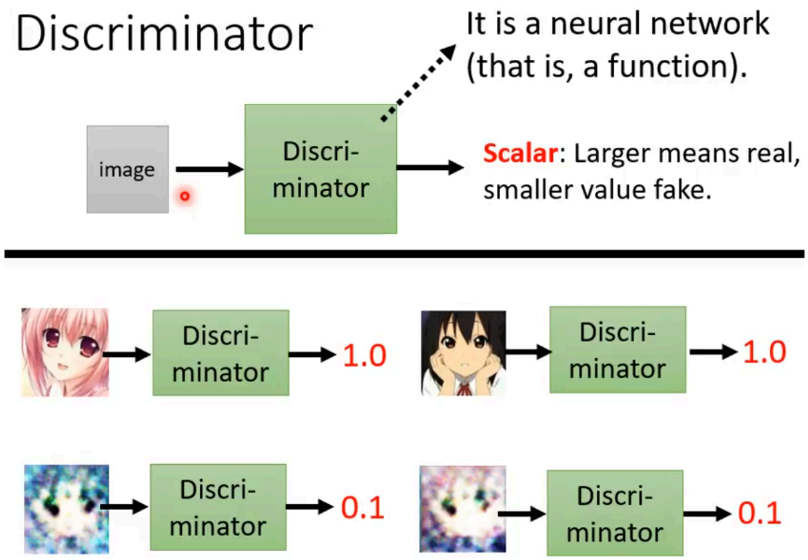
Example network (task: generate anime girl faces). The training process optimizes the discriminator and generator repeatedly

- Generator



  - input: vectors sampled from a simple distribution
  - ouput: vectors that from a complex distribution P_G, whicih ideally is close to the real data distribution P_{data}
  - optimization: we aim to have P_G be as close as possible to P_{data}
    $G^* = arg\min_G Div(P_G, P_{data})$
    The Div represents the **divergence** between the distribution generated and the real data distribution. The divergence can be measured using **[Jensen-Shannon (JS) divergnece]**.

- Discriminator



  - input: data points (vector sequence) from P_G and P_{data}
  - output: a probability that the input data point is real
  - optimization: make the discriminator D to become as good as possible at distinguishing between P_G and P_{data}
    $D^* = arg\max_D V(D,G) = arg\max_D(E_{y\sim P_{data}}[\log D(y)] + E_{y\sim P_G}[\log{(1-D(y)]})$
    the objective function is composed by two parts:
    - $E_{y\sim P_{data}}[\log D(y)]$: the probability that D gives to the data points from P_{data} being real
    - $E_{y\sim P_G}[\log{(1-D(y)]}$: the probability that D gives to the data points from P_G being fake

WGAN

From Wikipedia:

> The discriminator is mainly as a critic to provide feedback for the generator about "how far it is from perfection", where "far" is defined as Jensen–Shannon divergence.

In WGAN, the "farness" is defined as Wasserstein Distance.
...

cGAN

Conditional GAN
input condition information apart from a simple distribution
e.g. descriptive texts in text-to-image generation
the generator should generate images that correctly reflect the content described in the text

the discriminator should determine whether the condition information of the generated data matches the real data

CycleGAN

CycleGAN is usually used in **unsupervised** image-to-image translation tasks like picture style transfer.

Consists of two generators (G1, G2), two discriminators (D1, D2) and Cycle Consistency Loss

- G1 transforms an image X from domain A to domain B
- G2 transforms X' back to domain A.
- In this cycle, we guarantees the similarity of X and $G_2(G_1(X))$ by using a special Loss function -- Cycle Consistency Loss.