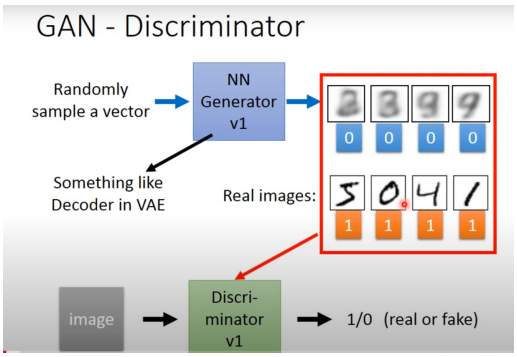
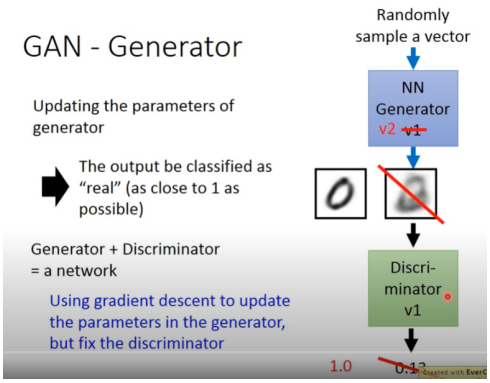
GAN

How to train Discriminator?



How to train Generator?



理论依据:

Maximum Likelihood Estimation

- Give a data distribution $P_{data}(x)$
- We have a distribution $P_G(x;\theta)$ parameterized by θ
  - E.g. $P_G(x;\theta)$ is a GMM, θ are means and variances of Gaussian
  - We want to find θ such that $P_G(x;\theta)$ close to $P_{data}(x)$

- Sample $\{x^1, x^2, \cdots, x^m\}$ from $P_{data}(x)$
- We can compute $P_G(x^i;\theta)$
- Likelihood of generating the samples: $L = \prod_{i=1}^{m} P_G(x^i;\theta)$
- Find θ^* maximizing the Likelihood.

$$\theta^* = \arg\max_\theta \prod_{i=1}^{m} P_G(x^i;\theta)$$

$$= \arg\max_\theta \sum_{i=1}^{m} \log P_G(x^i;\theta) \quad \{x^1, x^2, \cdots, x^n\} \text{ from } P_{data}(x)$$

$$\approx \arg\max_\theta E_{x\sim P_{data}}[\log P_G(x^i;\theta)]$$

$$= \arg\max_\theta \int_x P_{data}(x) \log P_G(x^i;\theta)\, dx$$

θ 和 x 无关，可以移到外面

$$= \arg\max_\theta \int_x P_{data}(x) \log P_G(x^i;\theta)\, dx - \int_x P_{data}(x) \log P_{data}(x)\, dx$$

$$= \arg\max_\theta \int_x \left[P_{data}(x) \log \frac{P_G(x^i;\theta)}{P_{data}(x)}\right] dx$$

$$= \arg\min_{\theta} \sum_{x} \left[\cancel{P_{data}(x)} \log \frac{P_{data}(x)}{P_G(x^i;\theta)} \right] dx$$

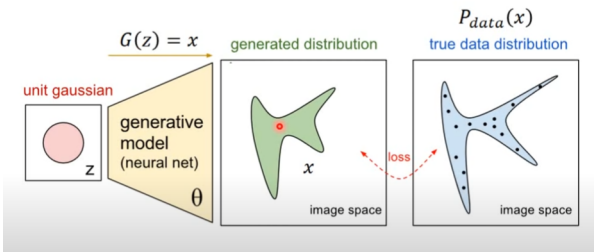
(与θ无关)

$$= \arg\min_{\theta} KL\left(P_{data}(x) \,||\, P_G(x^i;\theta)\right)$$

KL散度越小，代表这两个分布越相近

Question: How to have a very general $P_G(x;\theta)$?

Now $P_G(x;\theta)$ is a NN



negative cross entropy

最小：min cross entropy

= 最大 max negative cross entropy

Basic Idea of GAN

- Generator G — Hard to learn by maximum likelihood
  - G is a function, input z, output x
  - Given a prior distribution $P_{prior}(z)$, a probability distribution $P_G(x)$ is defined by function G
- Discriminator D
  - D is a function, input x, output scalar
  - Evaluate the "difference" between $P_G(x)$ and $P_{data}(x)$
- There is a function V(G,D).

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

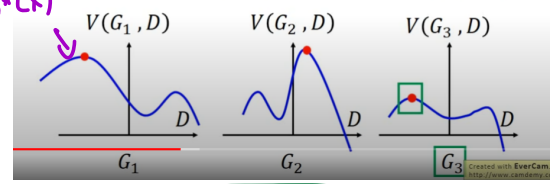
Basic Idea

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

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

Given a generator G, $\max_D V(G,D)$ evaluate the "difference" between P_G and P_{data}

Pick the G defining P_G most similar to P_{data}



given G, maximize D. $G^* = \arg\min_G \max_D V(G,D)$

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

$$= \int P_{data}(x) \log D(x)\, dx + \int P(G) \log(1-D(x))\, dx$$

$$\Rightarrow \int \left(P_{data}(x) \log D(x) + P(G) \log(1-D(x)) \right) dx$$

找 D(x)，令值最大

$$\Rightarrow a \log D(x) + b \log(1-D(x))$$

求导置零 $D^*(x) = \frac{P_{data}(x)}{P_{data}(x) + P_G(x)}$

Given D^*, minimize G:

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

$$= E_{x\sim P_{data}}\left[\log \frac{P_{data}(x)}{P_{data}(x)+P_G(x)}\right] + E_{x\sim P_G}\left[\log\left(1-\frac{P_{data}(x)}{P_{data}(x)+P_G(x)}\right)\right]$$

$$= \int_x P_{data}(x)\left[\log \frac{P_{data}(x)}{P_{data}(x)+P_G(x)}\right] + \int_x P(G)\left[\log\left(1-\frac{P_{data}(x)}{P_{data}(x)+P_G(x)}\right)\right]dx$$

$$= \int_x P_{data}(x)\left[\log \frac{P_{data}(x)}{(P_{data}(x)+P_G(x))/2}\right]dx + \int_x P(G)\left[\log\left(1-\frac{P_{data}(x)}{(P_{data}(x)+P_G(x))/2}\right)\right]dx - 2\log 2$$

$$= -2\log 2 + KL\left(P_{data}(x) \,\|\, \frac{P_{data}(x)+P_G(x)}{2}\right) + KL\left(P_G(x) \,\|\, \frac{P_{data}(x)+P_G(x)}{2}\right)$$

Remark:

$JSD(P\|Q) = \frac{1}{2}D(P\|M) + \frac{1}{2}D(Q\|M)$ where $M = \frac{1}{2}(P+Q)$

$= -2\log 2 + 2JSD\left(P_{data}(x) \,\|\, P_G(x)\right)$

In the end .....

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

- Generator G, Discriminator D
- Looking for G* such that

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

- Given G, $\max_D V(G,D)$ $= -2\log 2 + 2JSD\left(P_{data}(x)\|P_G(x)\right)$ (0 < … < log 2)
- What is the optimal G?

$$P_G(x) = P_{data}(x)$$

the value is related to JS divergence

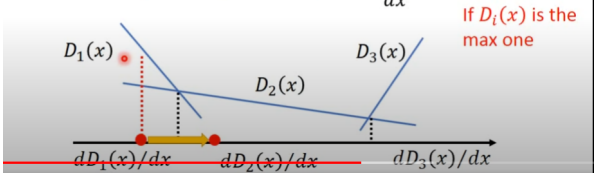
Question: 怎么解?

Algorithm $G^* = \arg\min_G \max_D V(G,D)$, $L(G)$

- To find the best G minimizing the loss function $L(G)$,

$\theta_G \leftarrow \theta_G - \eta\, \partial L(G)/\partial\theta_G$ (θ_G defines G)

$f(x) = \max\{D_1(x), D_2(x), D_3(x)\}$ $\frac{df(x)}{dx} = ?\ dD_i(x)/dx$ If $D_i(x)$ is the max one



Algorithm $G^* = \arg\min_G \max_D V(G,D)$, $L(G)$

- Given G_0
- Find D_0^* maximizing $V(G_0, D)$

$V(G_0, D_0^*)$ is the JS divergence between $P_{data}(x)$ and $P_{G_0}(x)$

- $\theta_G \leftarrow \theta_G - \eta\, \partial V(G, D_0^*)/\partial\theta_G$ → Obtain G_1 (Decrease JS divergence(?))
- Find D_1^* maximizing $V(G_1, D)$

$V(G_1, D_1^*)$ is the JS divergence between $P_{data}(x)$ and $P_{G_1}(x)$

- $\theta_G \leftarrow \theta_G - \eta\, \partial V(G, D_1^*)/\partial\theta_G$ → Obtain G_2 (Decrease JS divergence(?))
- .....

In practice ...

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

- Given G, how to compute $\max_D V(G,D)$
  - Sample $\{x^1, x^2, \dots, x^m\}$ from $P_{data}(x)$, sample $\{\tilde{x}^1, \tilde{x}^2, \dots, \tilde{x}^m\}$ from generator $P_G(x)$

Maximize $\tilde{V} = \frac{1}{m}\sum_{i=1}^{m} logD(x^i) + \frac{1}{m}\sum_{i=1}^{m} log\left(1-D(\tilde{x}^i)\right)$

Binary Classifier

Output is D(x) Minimize Cross-entropy

If x is a positive example ➡ Minimize –log D(x)

If x is a negative example ➡ Minimize –log(1-D(x))

Binary Classifier

Output is f(x) Minimize Cross-entropy

If x is a positive example ➡ Minimize –log f(x)

If x is a negative example ➡ Minimize –log(1-f(x))

D is a binary classifier (can be deep) with parameters θ_d

$\{x^1, x^2, \dots, x^m\}$ from $P_{data}(x)$ ➡ Positive examples

$\{\tilde{x}^1, \tilde{x}^2, \dots, \tilde{x}^m\}$ from $P_G(x)$ ➡ Negative examples

Minimize $L = \frac{1}{m}\sum_{i=1}^{m} logD(x^i) + \frac{1}{m}\sum_{i=1}^{m} log\left(1-D(\tilde{x}^i)\right)$

=

Maximize $\tilde{V} = \frac{1}{m}\sum_{i=1}^{m} logD(x^i) + \frac{1}{m}\sum_{i=1}^{m} log\left(1-D(\tilde{x}^i)\right)$

Algorithm Initialize θ_d for D and θ_g for G

- In each training iteration: Can only find lower found of $\max_D V(G,D)$

Learning D — Repeat k times:

- Sample m examples $\{x^1, x^2, \dots, x^m\}$ from data distribution $P_{data}(x)$
- Sample m noise samples $\{z^1, z^2, \dots, z^m\}$ from the prior $P_{prior}(z)$
- 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
  - $\tilde{V} = \frac{1}{m}\sum_{i=1}^{m} logD(x^i) + \frac{1}{m}\sum_{i=1}^{m} log\left(1-D(\tilde{x}^i)\right)$
  - $\theta_d \leftarrow \theta_d + \eta\nabla\tilde{V}(\theta_d)$

Learning G — Only Once:

- Sample another m noise samples $\{z^1, z^2, \dots, z^m\}$ from the prior $P_{prior}(z)$
- Update generator parameters θ_g to minimize
  - $\tilde{V} = \frac{1}{m}\sum_{i=1}^{m} logD(x^i) + \frac{1}{m}\sum_{i=1}^{m} log\left(1-D\left(G(z^i)\right)\right)$
  - $\theta_g \leftarrow \theta_g - \eta\nabla\tilde{V}(\theta_g)$

讲讲 GAN 的小技巧:

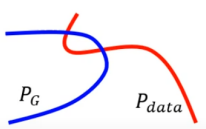
Background:

JS divergence is not suitable

- In most cases, P_G and P_{data} are not overlapped.
- 1. The nature of data

  Both P_{data} and P_G are low-dim manifold in high-dim space.

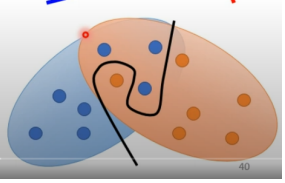
  The overlap can be ignored.

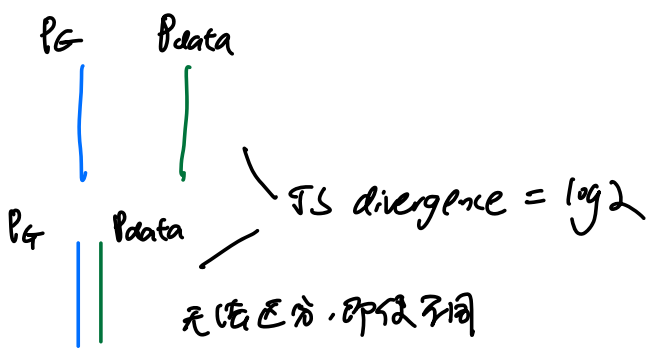


- 2. Sampling

  Even though P_{data} and P_G have overlap.

  If you do not have enough sampling .....





What's the problem of JS divergence?

JS divergence is always log2 if two distribution do not overlapped!

解决方法：

WGAN

WGAN

Evaluate Wasserstein distance between P_{data} and P_G

spectral normalization

$$\max_{D \in 1\text{-}Lipschitz} \{E_{x \sim P_{data}}[D(x)] - E_{x \sim P_G}[D(x)]\}$$

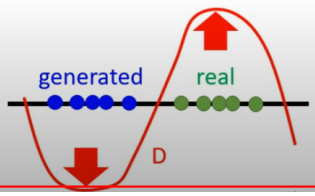
D has to be smooth enough.

平滑限制3 两个波峰相差不会很大

Without the constraint, the training of D will not converge.

Keeping the D smooth forces D(x) become ∞ and $-\infty$

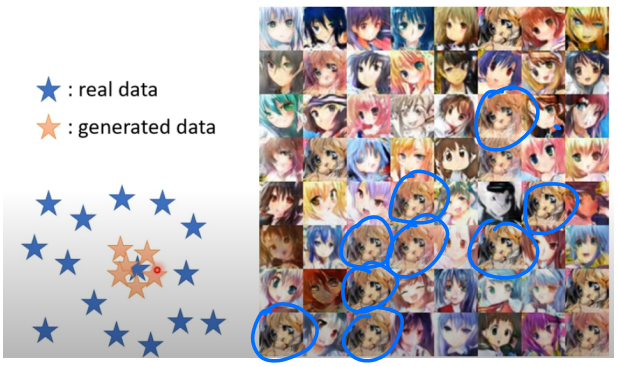
doesn't



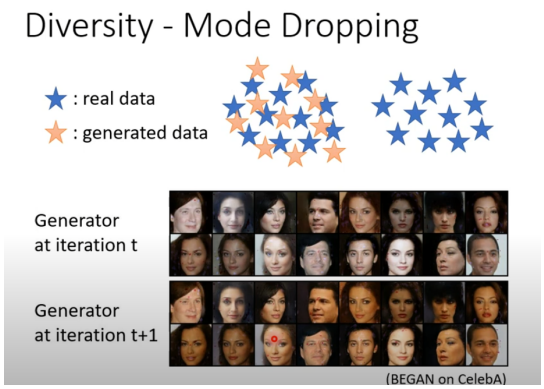
Evaluation of Generator (takeways: 很多，但用的没有最全)

1. 用图片分类/识别系统操作生成的图像.

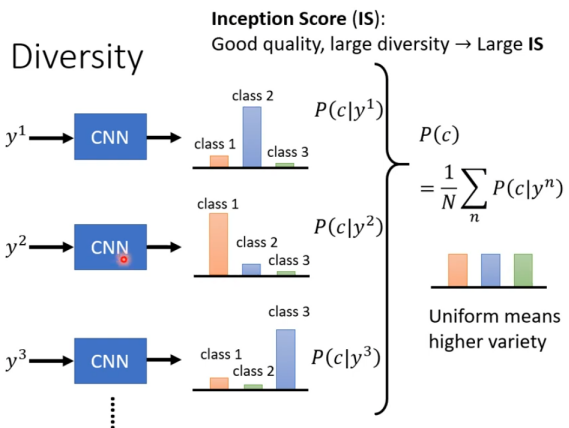
1. Drawback: Mode Collapse



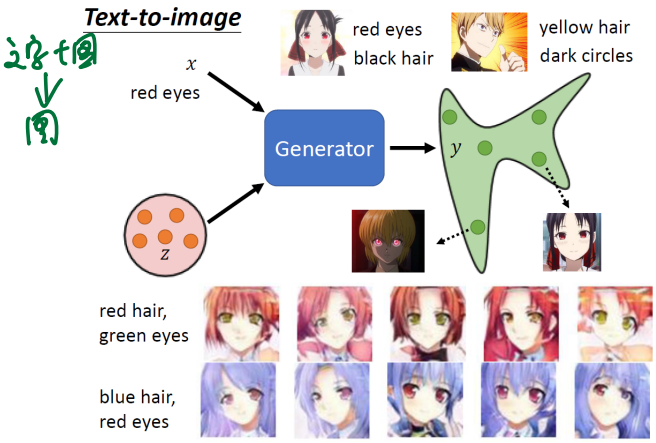
Mode Dropping

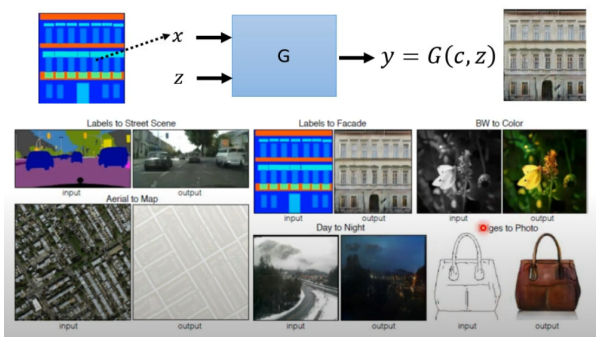


2. 如何判断是否 diversity ?



Conditional Generation (Conditional GAN)





Learning from Unpaired Data

Cycle GAN