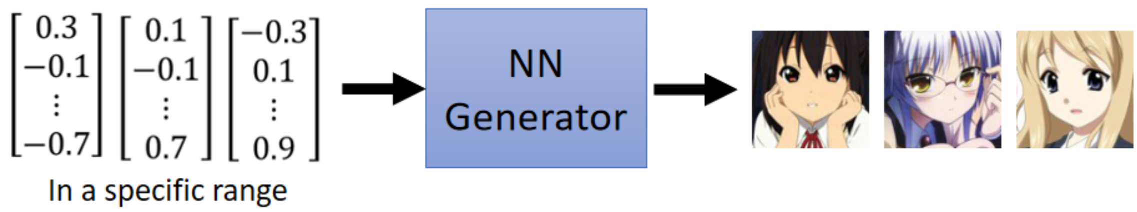
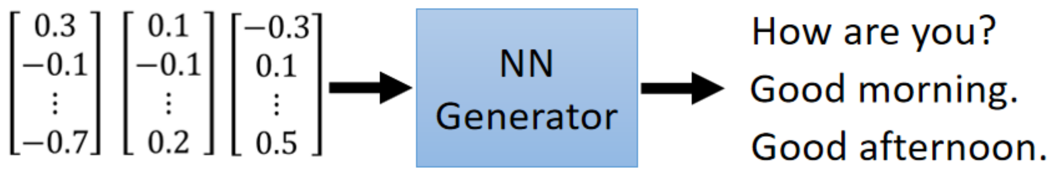
# GAN (11.30)

## 1、Generation

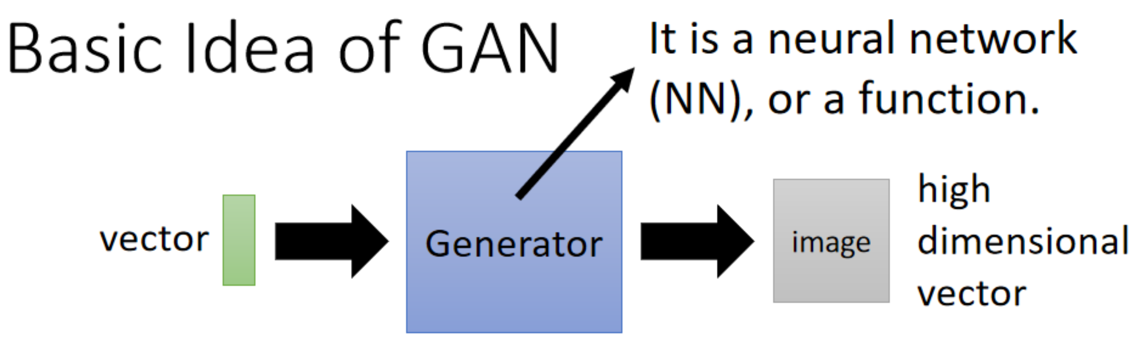
***Image Generation***

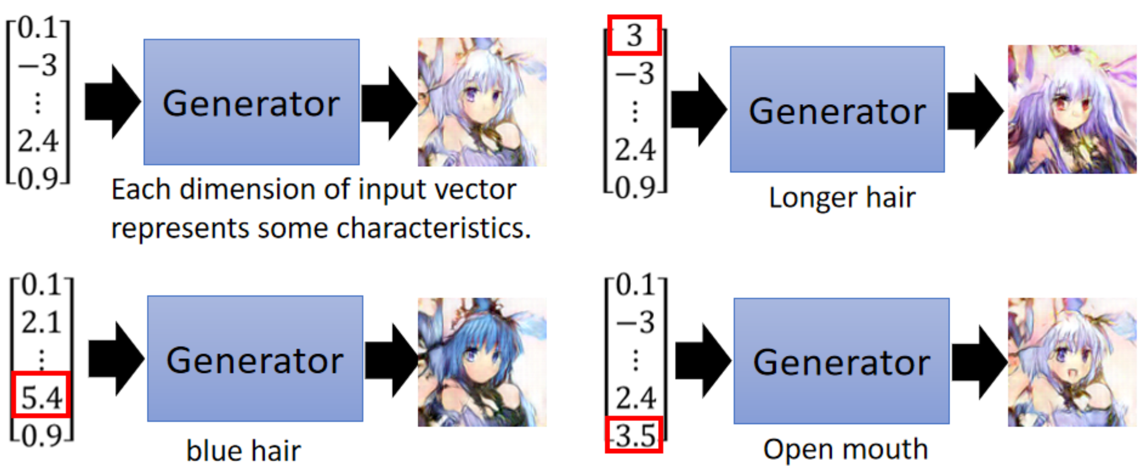



***Sentence Generation***

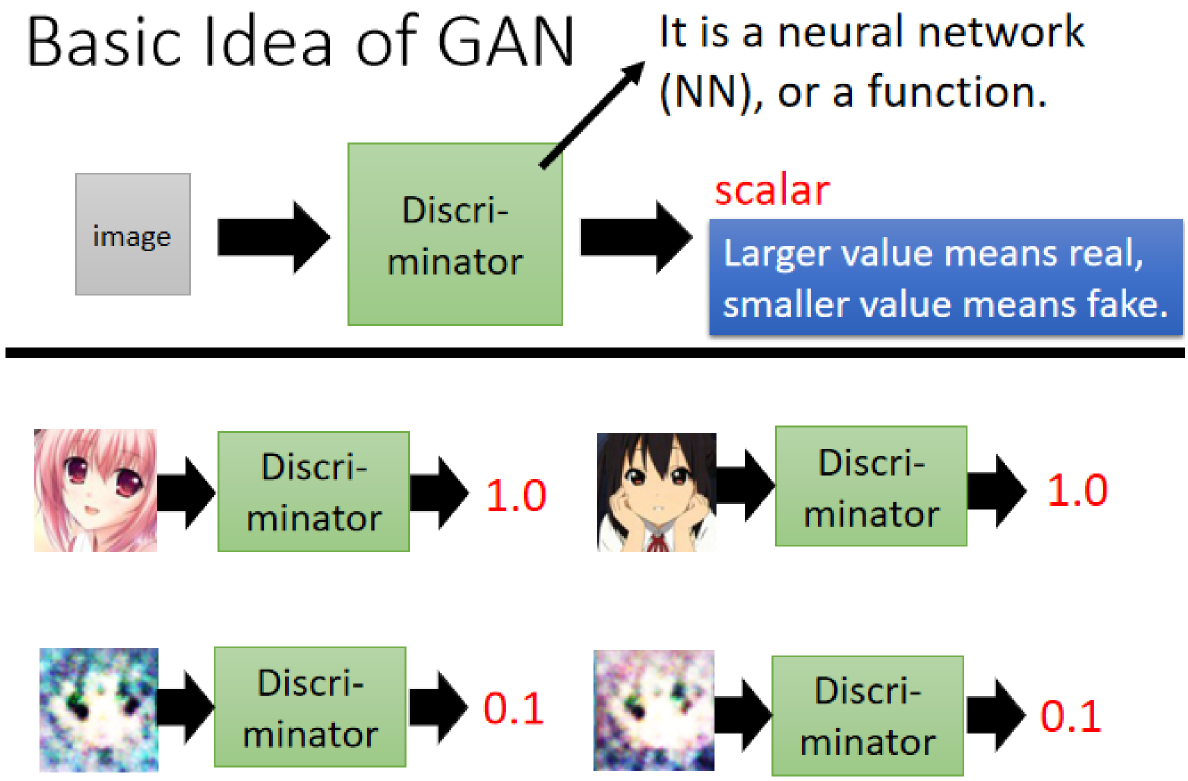



## 2、GAN的基本概念



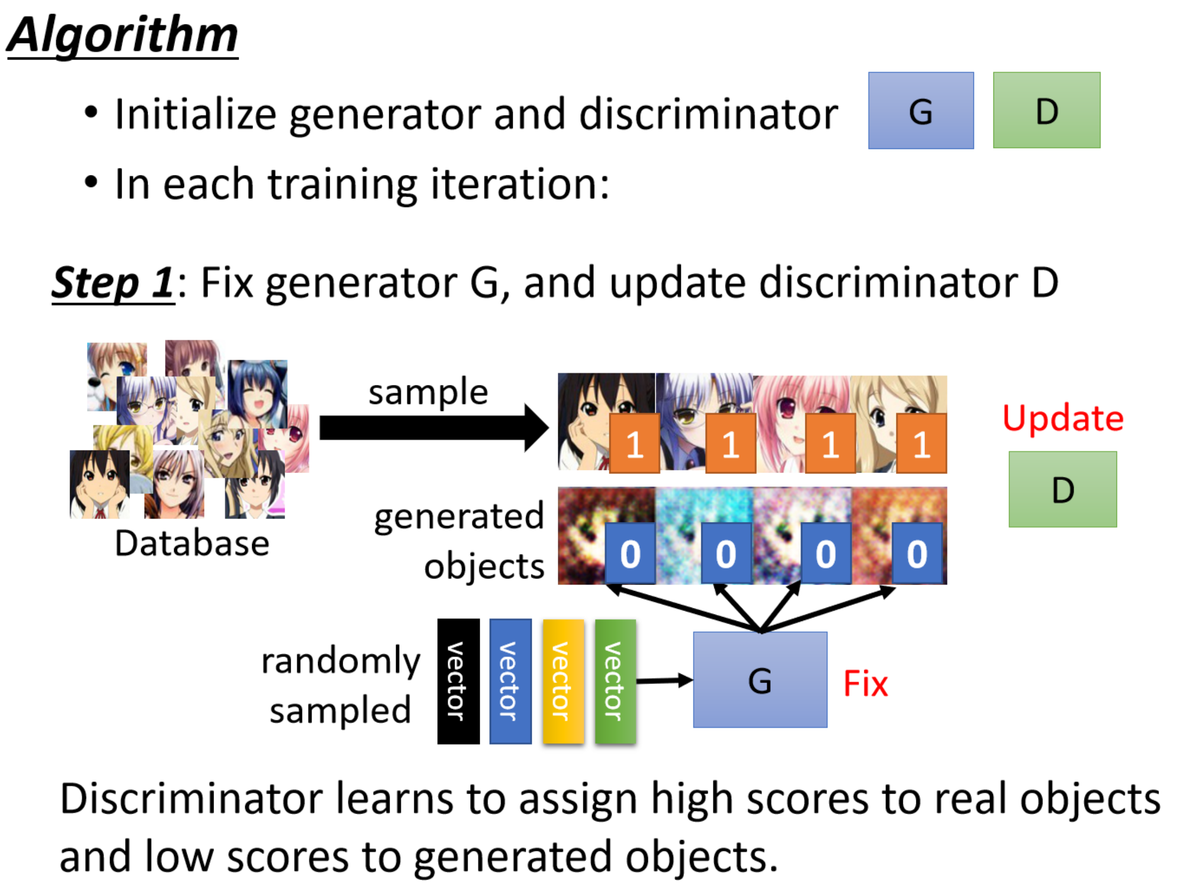



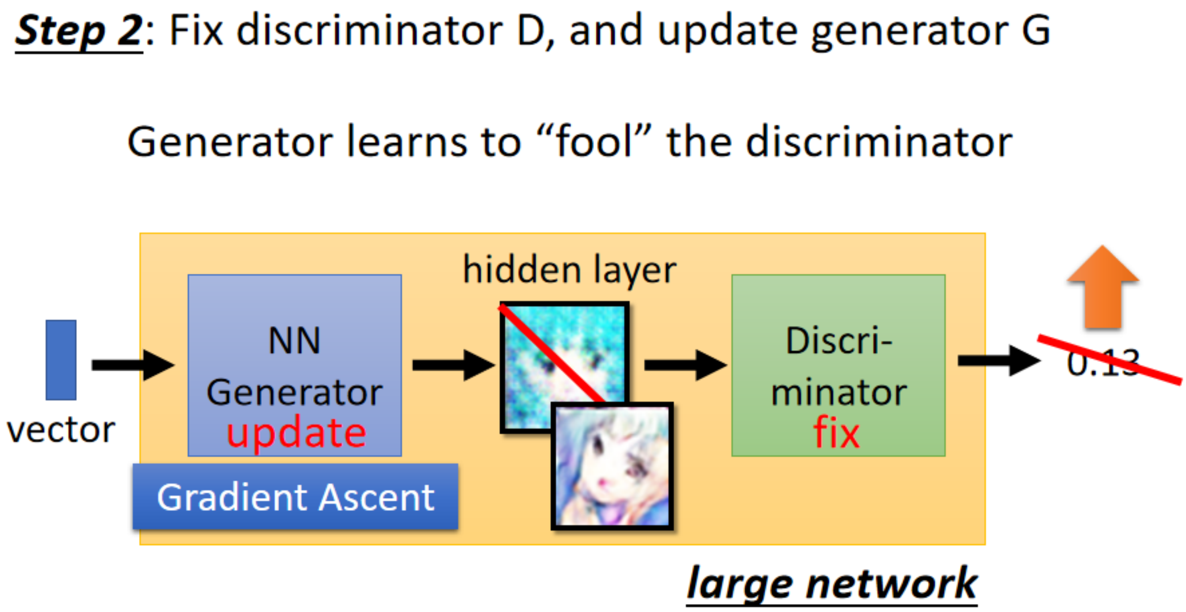

**生成器：输入向量---->输出向量，所以首先要有个东西做这件事**


Basic Idea of GAN
It is a neural network (NN), or a function.
image
Discri-minator
scalar
Larger value means real, smaller value means fake.
Discri-minator
1.0
Discri-minator
1.0
Discri-minator
0.1
Discri-minator
0.1

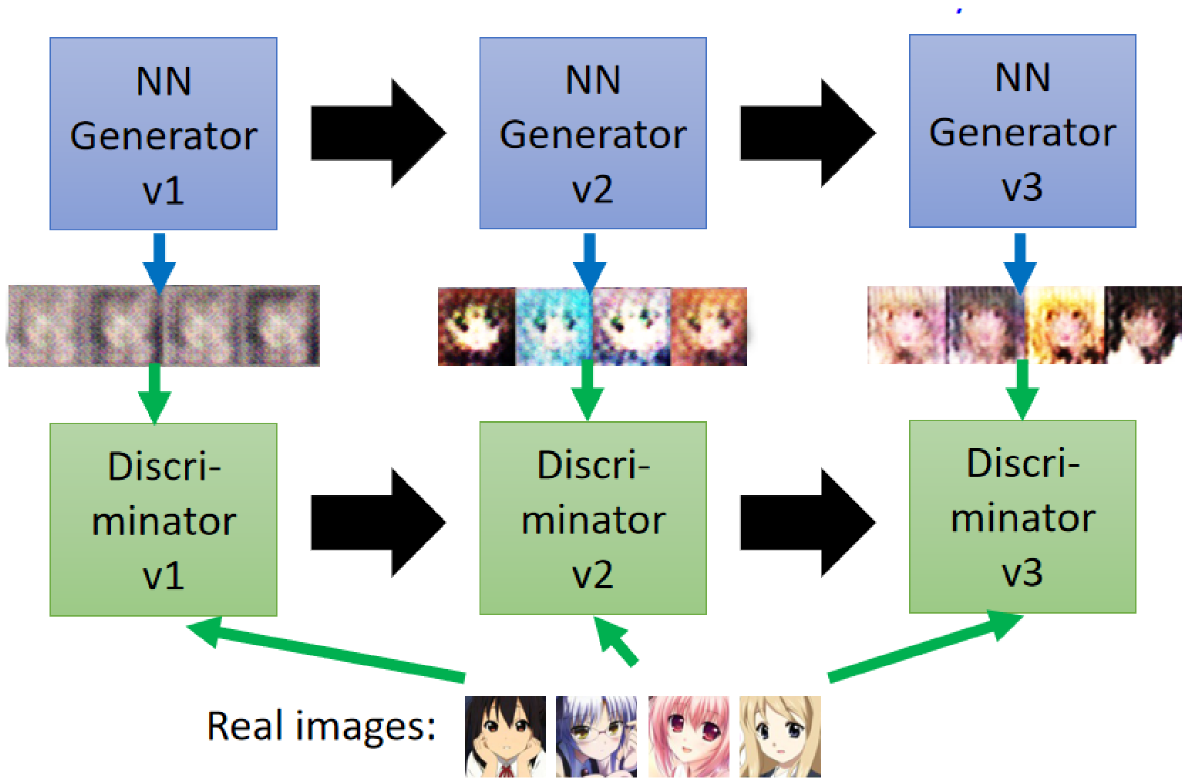
判别器：向量--->向量

3、算法


Algorithm
• Initialize generator and discriminator
G
D
• In each training iteration:
Step 1: Fix generator G, and update discriminator D
Database
sample
1
1
1
1
generated objects
0
0
0
0
Update
D
randomly sampled
vector
vector
vector
vector
G
Fix
Discriminator learns to assign high scores to real objects and low scores to generated objects.




**对抗：个人更喜欢进化/演化的解释，例如：**




- 第一代时，Step1: 固定生成器，训练判别器，训练的目标是让判别器能够对fake和real图片正确区分，这时判别器比较弱，因为生成器生成的图片和真实图片很容易区分。Step2: 固定判别器，训练生成器，训练的目标是让生成器生成的图片，判别器不能正确区分。这时生成器开始进化，它要生成能够欺骗第一代判别器的图片。
- 第二代时：判别器也开始进化，因为这时生成的图片和真实图片更难区分了。后面不断交替进化。

**Algorithm 1** Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, $k$, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

**for** number of training iterations **do**

  **for** $k$ steps **do**

- Sample minibatch of $m$ noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of $m$ examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[ \log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

  **end for**

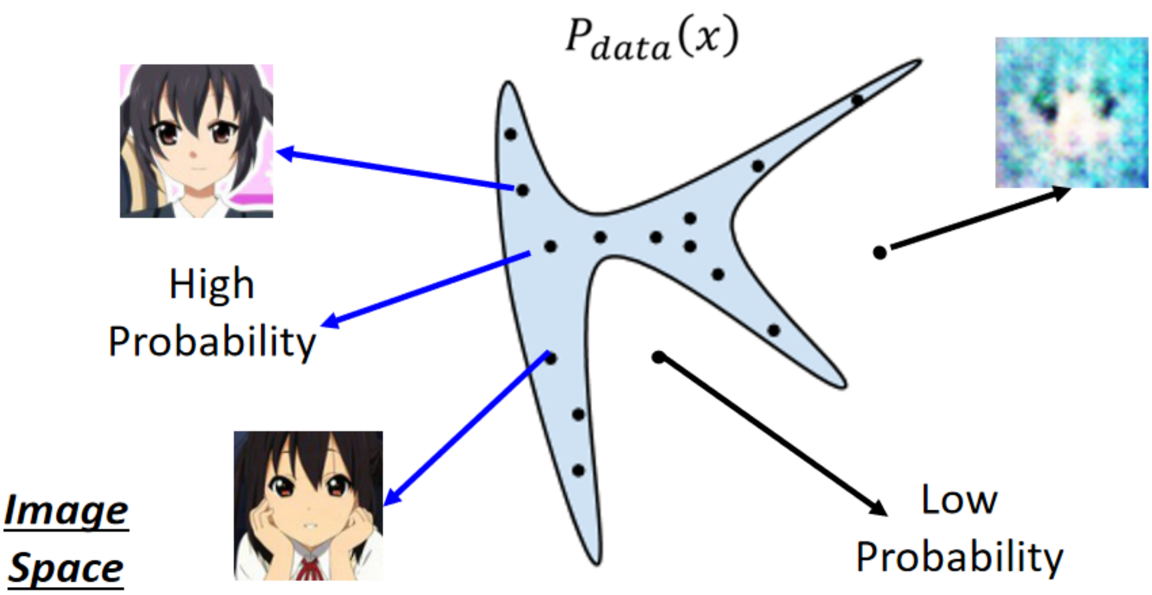
- Sample minibatch of $m$ noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

**end for**

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

## 4、理论




- <u>**图像数据分布在高维空间中，现在看下特殊点的情况，假设有个高斯分布，要估计它的最优参数，采用MLE**</u>

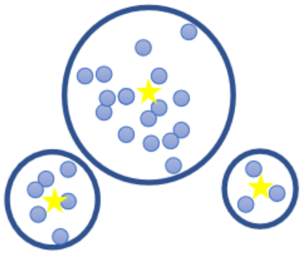
- Given a data distribution $P_{data}(x)$ (We can sample from it.)
- We have a distribution $P_G(x;\theta)$ parameterized by $\theta$
  - We want to find $\theta$ such that $P_G(x;\theta)$ close to $P_{data}(x)$
  - E.g. $P_G(x;\theta)$ is a Gaussian Mixture Model, $\theta$ are means and variances of the Gaussians

Sample $\{x^1, x^2, \ldots, x^m\}$ 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 $\theta^*$ maximizing the likelihood

- **MLE其实是在做最小化KL散度**

# Maximum Likelihood Estimation = Minimize KL Divergence

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

$$= arg\max_\theta \sum_{i=1}^{m} logP_G(x^i;\theta)$$

$\{x^1, x^2, \ldots, x^m\}$ from $P_{data}(x)$

$$\approx arg\max_\theta E_{x\sim P_{data}}[logP_G(x;\theta)]$$

$$= arg\max_\theta \int_x P_{data}(x)logP_G(x;\theta)dx \; - \int_x P_{data}(x)logP_{data}(x)dx$$

$$= arg\min_\theta KL(P_{data}||P_G)$$

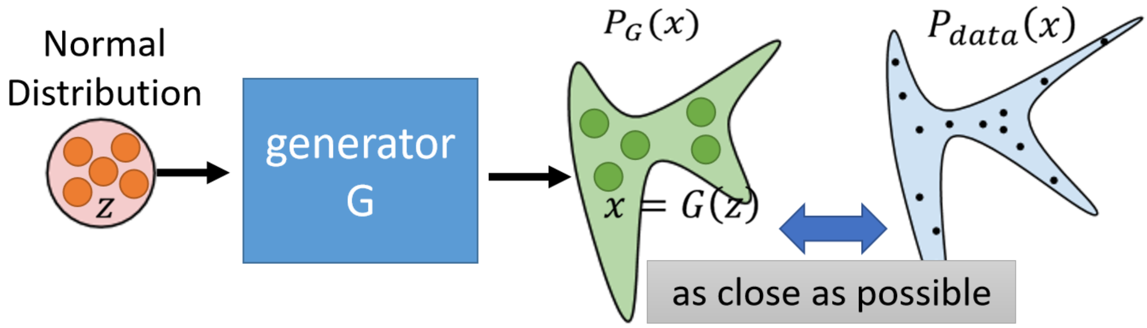
How to define a general $P_G$?

- 当成器生成的分布不知道时，通过采样

## 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?

先回到判别器的训练，采样：

**Example** Objective Function for D

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

(G is fixed)

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

The maximum objective value is related to JS divergence.

[Goodfellow, et al., NIPS, 2014]

求解D：

- Given G, what is the optimal D* maximizing

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

$$= \int_x P_{data}(x)\log D(x)\,dx + \int_x P_G(x)\log\left(1 - D(x)\right)dx$$

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

Assume that D(x) can be any function

- Given x, the optimal D* maximizing

$$P_{data}(x)\log D(x) + P_G(x)\log\left(1 - D(x)\right)$$

**推导:**

- Given x, the optimal D* maximizing

$$\underset{a}{P_{data}(x)}\log \underset{D}{D(x)} + \underset{b}{P_G(x)}\log\left(1 - \underset{D}{D(x)}\right)$$

- Find D* maximizing: $f(D) = a\log(D) + b\log(1 - D)$

$$\frac{df(D)}{dD} = a \times \frac{1}{D} + b \times \frac{1}{1-D} \times (-1) = 0$$

$$a \times \frac{1}{D^*} = b \times \frac{1}{1-D^*} \qquad a \times (1 - D^*) = b \times D^*$$

$$a - aD^* = bD^* \qquad a = (a+b)D^*$$

$$D^* = \frac{a}{a+b} \Rightarrow 0 < D^*(x) = \frac{P_{data}(x)}{P_{data}(x) + P_G(x)} < 1$$

$$\max_D V(G,D) = V(G,D^*) \qquad D^*(x) = \frac{P_{data}(x)}{P_{data}(x)+P_G(x)}$$

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

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

$$= \int_x P_{data}(x) log \frac{\frac{1}{2} P_{data}(x)}{\frac{P_{data}(x)+P_G(x)}{2}} dx$$

$$+ \int_x P_G(x) log \frac{\frac{1}{2} P_G(x)}{\frac{P_{data}(x)+P_G(x)}{2}} dx$$

$$+2log\frac{1}{2} \qquad -2log2$$

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

$$M = \frac{1}{2}(P+Q)$$

$$\max_D V(G,D)$$

$$\max_D V(G,D) = V(G,D^*) \qquad D^*(x) = \frac{P_{data}(x)}{P_{data}(x)+P_G(x)}$$

$$= -2log2 + \int_x P_{data}(x) log \frac{P_{data}(x)}{(P_{data}(x)+P_G(x))/2} dx$$

$$+ \int_x P_G(x) log \frac{P_G(x)}{(P_{data}(x)+P_G(x))/2} dx$$

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

$$= -2log2 + 2JSD(P_{data} || P_G)$$

Jensen-Shannon divergence

**结论：其实是找有一个G，是JS最小**

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

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

The maximum objective value is related to JS divergence.