假如要生成人脸，可能有个固定区域的分布，在区域内采样几率高，区域外采样几率低
我们希望通过GAN找出这个distribution

过去的方法：最大似然估计

给一组数据分布 $P_{data}(x)$

$P_G(x;\theta)$

Goal：找到 θ 使 $P_G(x;\theta)$ 接近 $P_{data}(x)$，其中 $P_G(x;\theta)$ 是高斯模型，θ 为 mean & variance

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

代入

我们可以计算 $P_G(x^i;\theta)$ 变量

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

maximum L, Find θ^*

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

$\boxed{\{x^1, x^2 \cdots x^m\} \text{ from } P_{data}(x)}$

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

这一项与 θ 无关
不影响 argmax

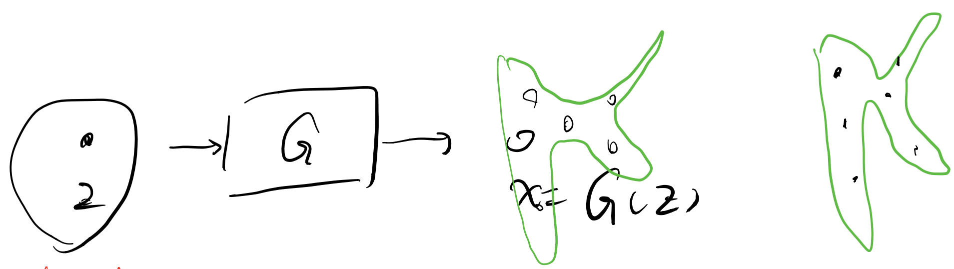
$$= \arg\max_\theta \int_x P_{data}(x)\log P_G(x;\theta)\,dx - \int_x P_{data}(x)\log P_{data}(x)\,dx.$$

加一项

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

$$\boxed{KL(P\|Q) = \sum_i P(i) \log\left(\frac{P(i)}{Q(i)}\right)}$$

A generator G is a network, defines a probability distribution P_G.



Normal Distribution

另一个 distribution

$P_G(x)$ $\longleftrightarrow$ $P_{data}(x)$.

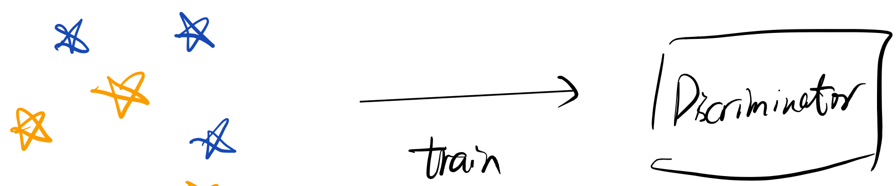
as close as possible.

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

How to compute divergence?

我们不知道 P_G and P_{data}，但是可以 sample.

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



sigmoid output

objective function for D:

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

fix住

Training: $D^* = \arg\max_D V(D,G)$ 和逻辑回归一样

与 JS div 有关

$\boxed{\begin{aligned} &\text{JS 散度:} \\ &JS(P_1\|P_2) = \tfrac{1}{2}KL\left(P_1\middle\|\tfrac{P_1+P_2}{2}\right) \\ &\qquad + \tfrac{1}{2}KL\left(P_2\middle\|\tfrac{P_1+P_2}{2}\right) \end{aligned}}$

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

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

Assume P(x) can be any function

• Given x, the optimal D^* maximizing

$$\overset{a}{P_{data}(x)}\log \overset{D}{D(x)} + \overset{b}{P_G(x)}\log(1-\overset{D}{D(x)}).$$

所有x分开来算，分开找最好的

$$f(D) = a\log D + b\log(1-D)$$

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

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

$$a - aD^* = bD^*$$

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

回代 V

$$V = E_{x\sim P_{data}}\left[\log \frac{\frac{1}{2}P_{data}(x)}{\frac{P_{data}(x)+P_G(x)}{2}}\right] + E_{x\sim P_G}\left[\log \frac{\frac{1}{2}P_G(x)}{\frac{P_{data}(x)+P_G(x)}{2}}\right]$$

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

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

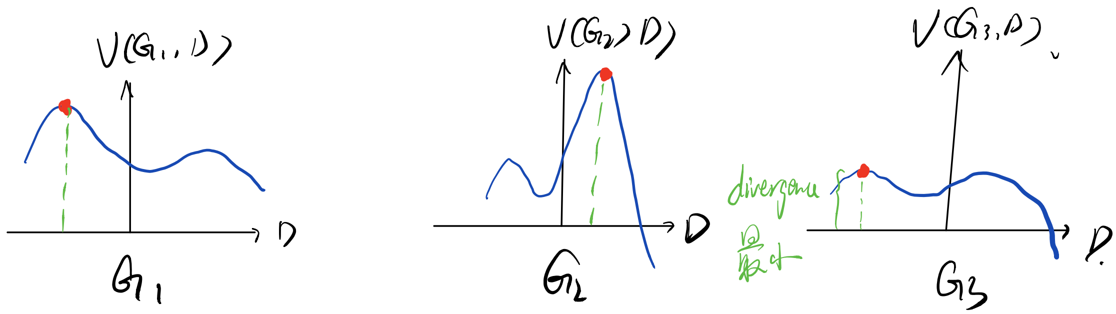
$$= -2\log 2 + 2JSD(P_{data}\,\|\,P_G)$$

解释：当我们 train discriminator 时，我们在 evaluate P_{data} 和 P_G 的 JS divergence

$$G^* = \arg\min_G Div(P_G, P_{data}) \rightarrow \text{无法计算}$$

$$\xrightarrow{\text{替换}} \arg\min_G \max_D V(G, D).$$

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



因此选择 $G^* = G_3$

- Initialize generator and discriminator.
- In each training iteration.

  step 1: Fix G, update D.

  step 2: Fix D, update G.

为什么上述 alg 是在解 minmax 问题？

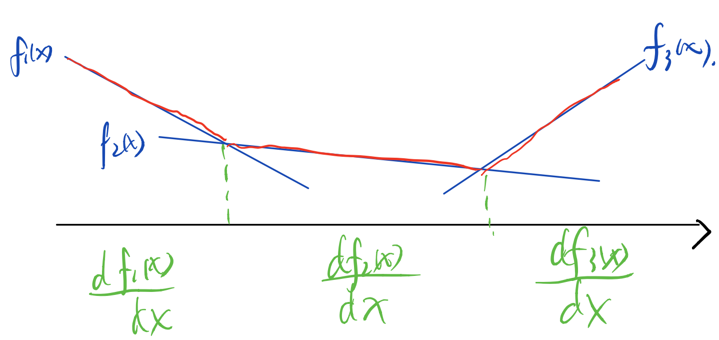
$$G^* = \arg\min_G \boxed{\max_D V(G,D)} \quad L(G)$$

- To find the best G minimizing the loss func L(G)

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

然而 L(G) 里有 max，可以求偏导吗？

$$f(x) = \max\{f_1(x), f_2(x), f_3(x)\}$$



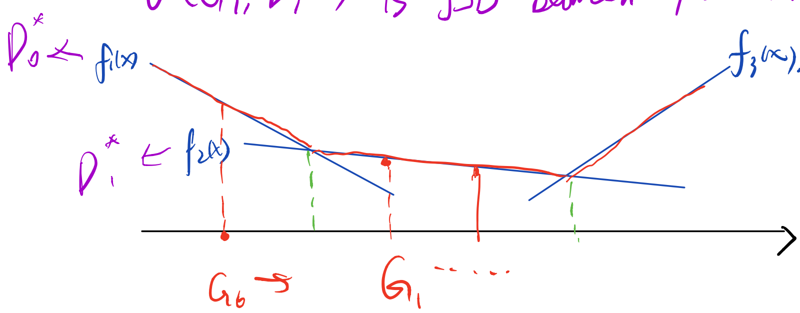
$\frac{df(x)}{dx} = ?$ $\frac{df_i(x)}{dx}$ if $f_i(x)$ is the max one

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

  $V(G_0, D_0^*)$ is JSD between $P_{data}(x)$ and $P_{G_0}(x)$
- $\theta_G \leftarrow \theta_G - \eta \, \partial V(G, D_0^*) / \partial \theta_G \implies$ obtain G_1

  $\boxed{\text{Decrease JSD}}$
- Find D_1^* maximizing $V(G_1, D)$

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



- $\theta_G \leftarrow \theta_G - \eta \, \partial V(G, D_1^*) / \partial \theta_G \rightarrow$ obtain G_2

  $\boxed{\text{Decrease JSD}}$

...

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

实际中我们无法计算 E，因此

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

sample $\{x_1, x_2 \dots x_m\}$ from $P_{data}(x)$

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

等同于 train binary classifier with sigmoid output

$\{x^1, x^2 \dots x^m\} \Longrightarrow$ positive examples

$\{\tilde{x}^1, \tilde{x}^2 \dots \tilde{x}^m\} \Longrightarrow$ negative examples

Minimize Cross-entropy

Algorithm Initialize θ_d for D and θ_g for G

Can only find lower bound of $\max_D V(G, D)$

- In each training iteration:

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
  - $\tilde{V} = \frac{1}{m}\sum_{i=1}^{m} \log D(x^i) + \frac{1}{m}\sum_{i=1}^{m} \log\left(1 - D(\tilde{x}^i)\right)$
  - $\theta_d \leftarrow \theta_d + \eta \nabla \tilde{V}(\theta_d)$

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
  - $\tilde{V} = \cancel{\frac{1}{m}\sum_{i=1}^{m} \log D(x^i)} + \frac{1}{m}\sum_{i=1}^{m} \log\left(1 - D\left(G(z^i)\right)\right)$
  - $\theta_g \leftarrow \theta_g - \eta \nabla \tilde{V}(\theta_g)$

Intuition

Discriminator
Data (target) distribution
Generated distribution

- Discriminator leads the generator