Improved GAN

王宗威

回顾

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

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

Algorithm

Initialize θ_d for D and θ_g for G

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

- In each training iteration:

Learning D

$\boxed{\text{Repeat k times}}$

- Sample m examples $\{x^1, x^2, \ldots, x^m\}$ from data distribution $P_{data}(x)$
- Sample m noise samples $\{z^1, z^2, \ldots, z^m\}$ from the prior $P_{prior}(z)$
- Obtaining generated data $\{\tilde{x}^1, \tilde{x}^2, \ldots, \tilde{x}^m\}$, $\tilde{x}^i = G(z^i)$
- 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

$\boxed{\text{Only Once}}$

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

Original GAN的缺点

问题：现在的GAN是否能指定生成想要的图片呢？

比如，

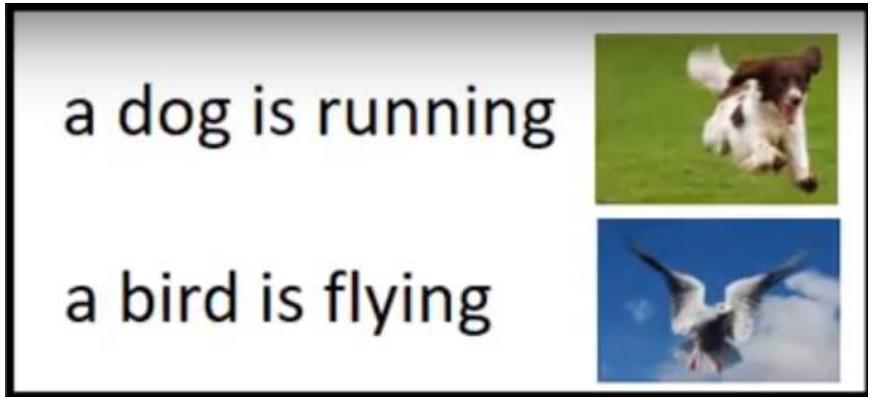
（1）case one：

我们现在的任务是只需要9这个数字

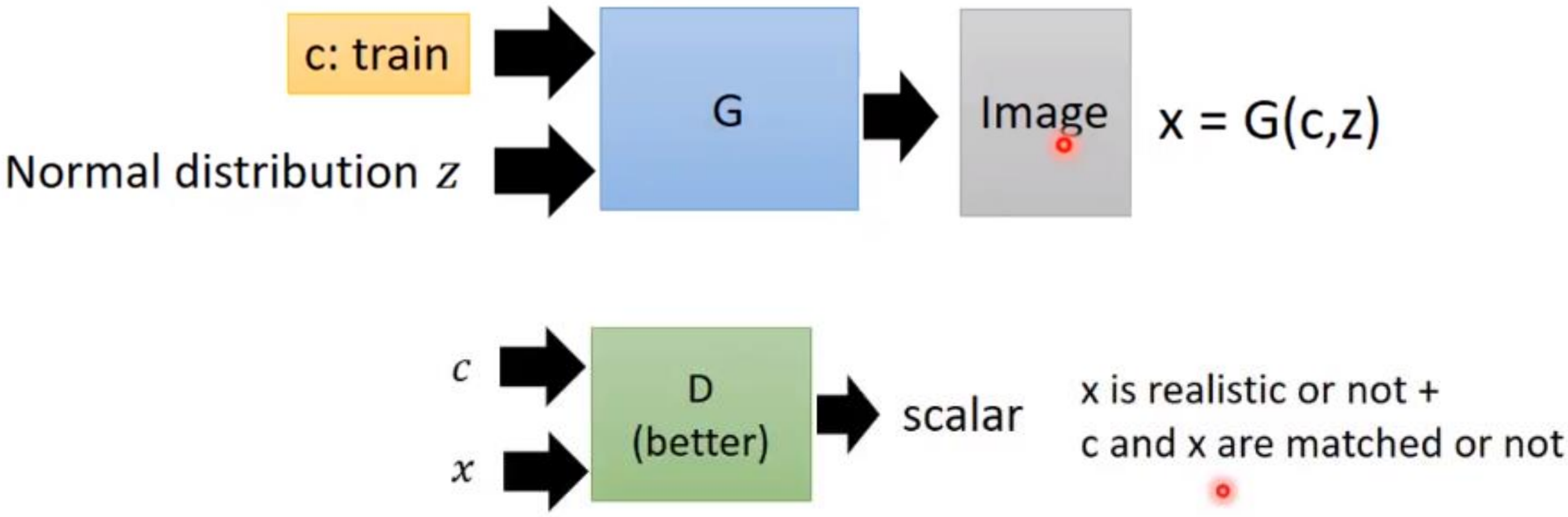
（2）case two：

我们给一句话， a dog is running

希望他生成一只小狗



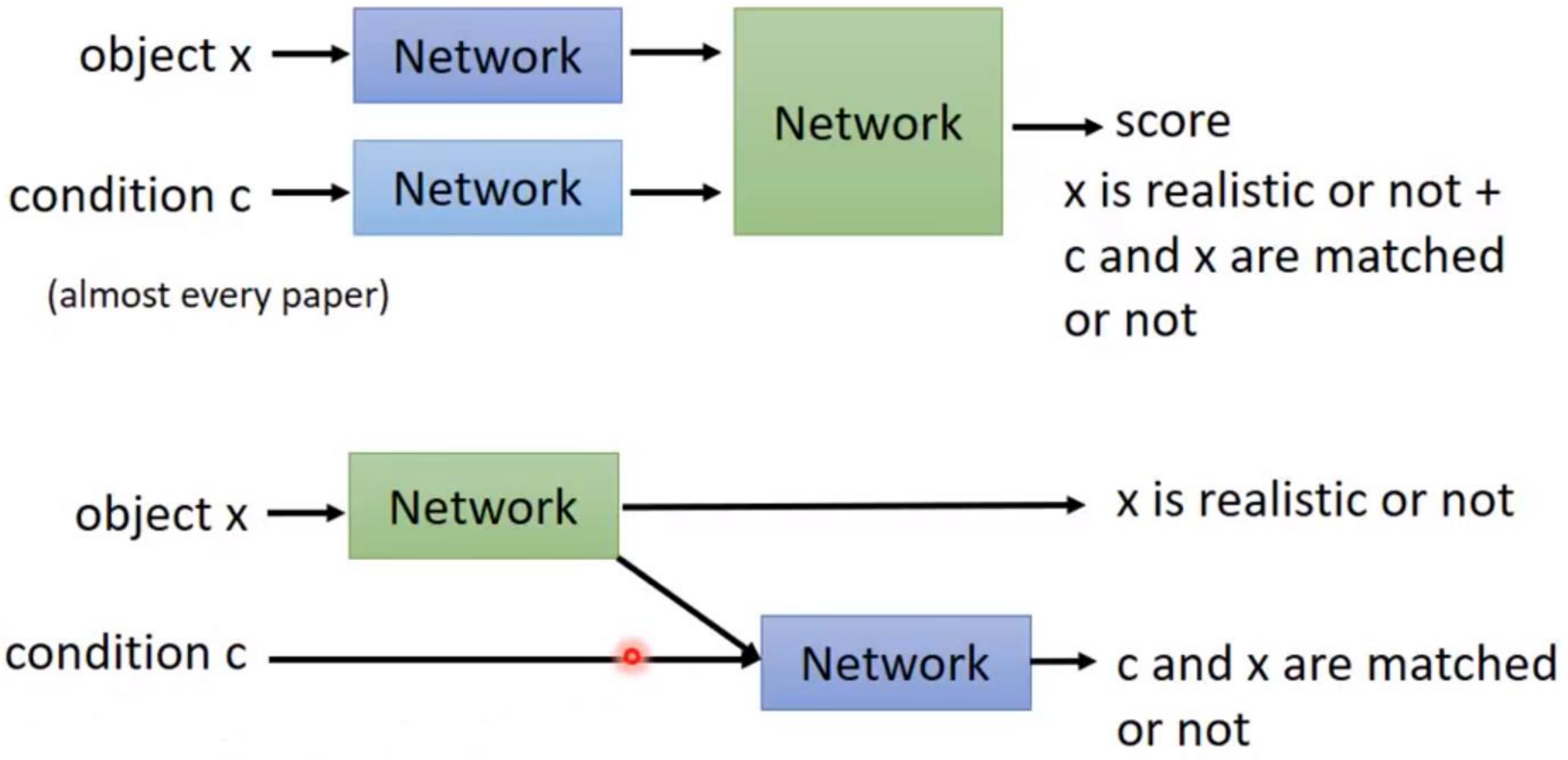
Conditional GAN



生成器
输入：随机噪声Z，标签向量c
输出：生成数据X1

判别器
输入：生成数据X1,真实数据X2，标签向量c
输出：标量Scalar

- In each training iteration:
  - Sample m positive examples $\{(c^1, x^1), (c^2, x^2), \dots, (c^m, x^m)\}$ from database
  - Sample m noise samples $\{z^1, z^2, \dots, z^m\}$ from a distribution
  - Obtaining generated data $\{\tilde{x}^1, \tilde{x}^2, \dots, \tilde{x}^m\}$, $\tilde{x}^i = G(c^i, z^i)$
  - Sample m objects $\{\hat{x}^1, \hat{x}^2, \dots, \hat{x}^m\}$ from database
  - Update discriminator parameters θ_d to maximize
    - $\tilde{V} = \frac{1}{m}\sum_{i=1}^{m} logD(c^i, x^i)$
      $+\frac{1}{m}\sum_{i=1}^{m} log\left(1 - D(c^i, \tilde{x}^i)\right) + \frac{1}{m}\sum_{i=1}^{m} log\left(1 - D(c^i, \hat{x}^i)\right)$
    - $\theta_d \leftarrow \theta_d + \eta\nabla\tilde{V}(\theta_d)$
  - Sample m noise samples $\{z^1, z^2, \dots, z^m\}$ from a distribution
  - Sample m conditions $\{c^1, c^2, \dots, c^m\}$ from a database
  - Update generator parameters θ_g to maximize
    - $\tilde{V} = \frac{1}{m}\sum_{i=1}^{m} log\left(D\left(G(c^i, z^i)\right)\right)$, $\theta_g \leftarrow \theta_g - \eta\nabla\tilde{V}(\theta_g)$


object x
Network
Network
score
condition c
Network
x is realistic or not +
c and x are matched
or not
(almost every paper)
object x
Network
x is realistic or not
condition c
Network
c and x are matched
or not

paired data

blue eyes
red hair
short hair

Collecting anime faces
and the description of its
characteristics

red hair,
green eyes

blue hair,
red eyes

f-GAN

为什么出现f-GAN？

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

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

$$G^* = arg\min_G D_f(P_{data}||P_G)$$

$$= arg\min_G \max_D \{E_{x\sim P_{data}}[D(x)] - E_{x\sim P_G}\left[f^*\left(D(x)\right)\right]$$

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

f-divergence

用来衡量P，Q两分布之间的差异

$$D_f(P||Q) = \int_x q(x) f\left(\frac{p(x)}{q(x)}\right) dx$$

性质：凸函数且f(1)=0

共轭函数

若一个函数f为凸函数，则它存在一个共轭函数

$$f^*(t) = \max_{x \in dom(f)} \{xt - f(x)\}$$

此处省略一万个字！！！！！！！！！！！！！！！！！

$$f^*(t) = \sup_{x \in dom(f)} \{xt - f(x)\} \longleftrightarrow f(x) = \max_{t \in dom(f^*)} \{xt - f^*(t)\}$$

$$f^*(t)=\sup_{x\in dom(f)}\{xt-f(x)\} \longleftrightarrow f(x)=\max_{t\in dom(f^*)}\{xt-f^*(t)\}$$

$$\begin{aligned}
D_f(P||Q) &= \int_x q(x)f\left(\frac{p(x)}{q(x)}\right)dx \\
&= \int_x q(x)\left(\max_{t\in dom(f^*)}\left\{\frac{p(x)}{q(x)}t-f^*(t)\right\}\right)dx \\
&\approx \max_D \int_x p(x)D(x)dx-\int_x q(x)f^*(D(x))dx
\end{aligned}$$

假设有一个D，输入是x，输出是t

$$\begin{aligned}
D_f(P||Q) &\geq \int_x q(x)\left(\frac{p(x)}{q(x)}\underline{D(x)}-f^*(\underline{D(x)})\right)dx \\
&= \int_x p(x)D(x)dx-\int_x q(x)f^*(D(x))dx
\end{aligned}$$

$$D_f(P||Q) \approx \max_{\mathrm{D}} \int_x p(x)D(x)dx - \int_x q(x)f^*(D(x))dx$$

$$= \max_{\mathrm{D}} \{E_{x\sim P}[D(x)] - E_{x\sim Q}[f^*(D(x))]\}$$

Samples from P Samples from Q

$$D_f(P_{data}||P_G) = \max_{\mathrm{D}} \{E_{x\sim P_{data}}[D(x)] - E_{x\sim P_G}[f^*(D(x))]\}$$

$$G^* = arg \min_G D_f(P_{data}||P_G)$$

$$= arg \min_G \max_D \{E_{x\sim P_{data}}[D(x)] - E_{x\sim P_G}[f^*(D(x))]\}$$

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

$$D_f(P_{data}||P_G) = \max_{D}\{E_{x\sim P_{data}}[D(x)] - E_{x\sim P_G}[f^*(D(x))]\}$$

Name	$D_f(P\|Q)$	Generator $f(u)$
Total variation	$\frac{1}{2}\int \lvert p(x)-q(x)\rvert \,\mathrm{d}x$	$\frac{1}{2}\lvert u-1\rvert$
Kullback-Leibler	$\int p(x)\log\frac{p(x)}{q(x)}\,\mathrm{d}x$	$u\log u$
Reverse Kullback-Leibler	$\int q(x)\log\frac{q(x)}{p(x)}\,\mathrm{d}x$	$-\log u$
Pearson χ^2	$\int \frac{(q(x)-p(x))^2}{p(x)}\,\mathrm{d}x$	$(u-1)^2$
Neyman χ^2	$\int \frac{(p(x)-q(x))^2}{q(x)}\,\mathrm{d}x$	$\frac{(1-u)^2}{u}$
Squared Hellinger	$\int \left(\sqrt{p(x)}-\sqrt{q(x)}\right)^2\,\mathrm{d}x$	$(\sqrt{u}-1)^2$
Jeffrey	$\int (p(x)-q(x))\log\left(\frac{p(x)}{q(x)}\right)\mathrm{d}x$	$(u-1)\log u$
Jensen-Shannon	$\frac{1}{2}\int p(x)\log\frac{2p(x)}{p(x)+q(x)} + q(x)\log\frac{2q(x)}{p(x)+q(x)}\,\mathrm{d}x$	$-(u+1)\log\frac{1+u}{2} + u\log u$
Jensen-Shannon-weighted	$\int p(x)\pi\log\frac{p(x)}{\pi p(x)+(1-\pi)q(x)} + (1-\pi)q(x)\log\frac{q(x)}{\pi p(x)+(1-\pi)q(x)}\,\mathrm{d}x$	$\pi u\log u - (1-\pi+\pi u)\log(1-\pi+\pi u)$
GAN	$\int p(x)\log\frac{2p(x)}{p(x)+q(x)} + q(x)\log\frac{2q(x)}{p(x)+q(x)}\,\mathrm{d}x - \log(4)$	$u\log u - (u+1)\log(u+1)$

Using the f-divergence you like ☺

https://arxiv.org/pdf/1606.00709.pdf

Name	Conjugate $f^*(t)$
Total variation	t
Kullback-Leibler (KL)	$\exp(t-1)$
Reverse KL	$-1-\log(-t)$
Pearson χ^2	$\frac{1}{4}t^2+t$
Neyman χ^2	$2-2\sqrt{1-t}$
Squared Hellinger	$\frac{t}{1-t}$
Jeffrey	$W(e^{1-t}) + \frac{1}{W(e^{1-t})} + t - 2$
Jensen-Shannon	$-\log(2-\exp(t))$
Jensen-Shannon-weighted	$(1-\pi)\log\frac{1-\pi}{1-\pi e^{t/\pi}}$
GAN	$-\log(1-\exp(t))$

Wasserstein GAN

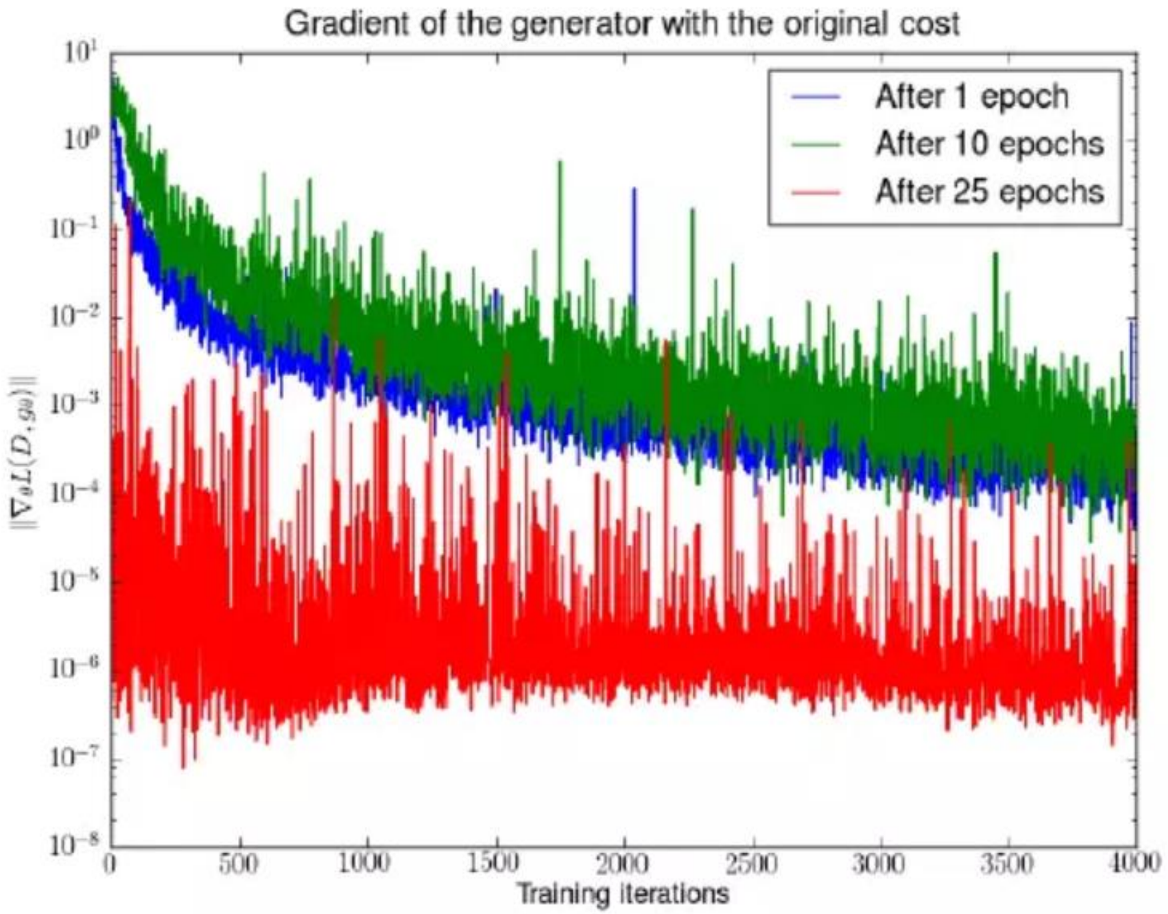
实质是什么？

将JS散度变成EM距离（ Wasserstein 距离）

JS散度的问题

$$\max_D V(G,D) = -2log2 + 2JSD\left(P_{data}(x)||P_G(x)\right)$$

梯度消失的问题



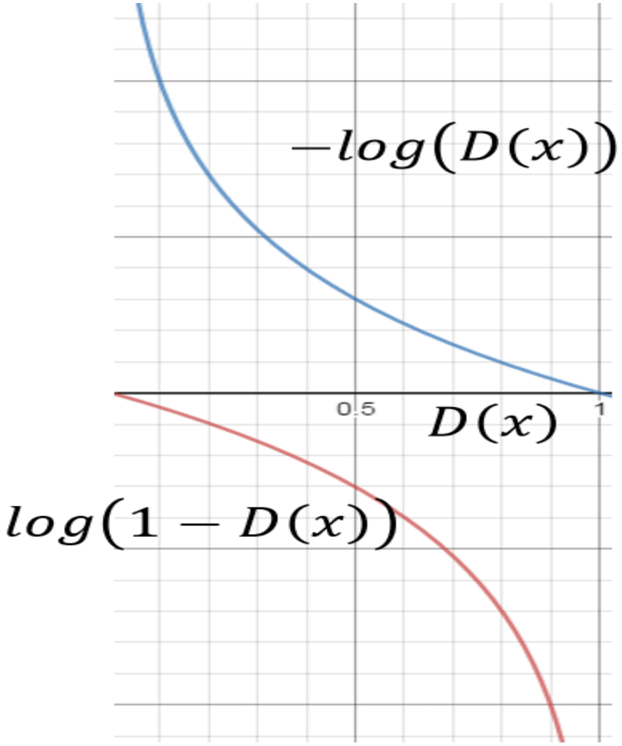
$$V = \cancel{E_{x\sim P_{data}}[logD(x)]}$$

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

Slow at the beginning

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

Real implementation:
label x from P_G as positive



$$\mathbb{E}_{z\sim p(z)}\left[-\nabla_\theta \log D^*(g_\theta(z))|_{\theta=\theta_0}\right] = \nabla_\theta\left[KL(\mathbb{P}_{g_\theta}\|\mathbb{P}_r) - 2JSD(\mathbb{P}_{g_\theta}\|\mathbb{P}_r)\right]|_{\theta=\theta_0}$$

1.第一项的KL散度会被最小化，这会带来严重的mode collapse问题。

2.第二项意味着最大化真实数据分布和生成数据分布之间的JS散度，也就是让两者差异化更大，这显然违背了最初的优化目标，算是一种缺陷

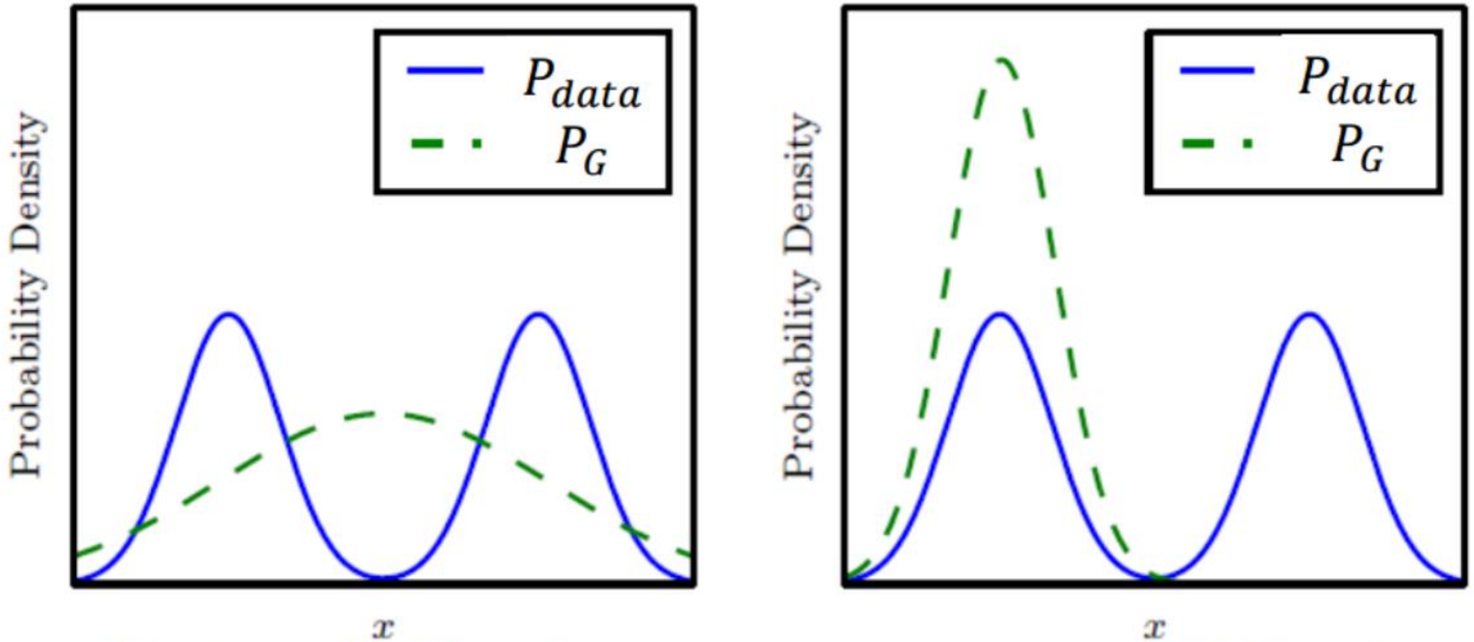
3.训练不稳定

Mode Collapse

$$\nabla_\theta [KL(\mathbb{P}_{g_\theta}\|\mathbb{P}_r)$$

Generator会生成大量高质量却缺乏多样性的样本

$$KL = \int P_{data} \log \frac{P_{data}}{P_G} dx \qquad \text{Reverse } KL = \int P_G \log \frac{P_G}{P_{data}} dx$$

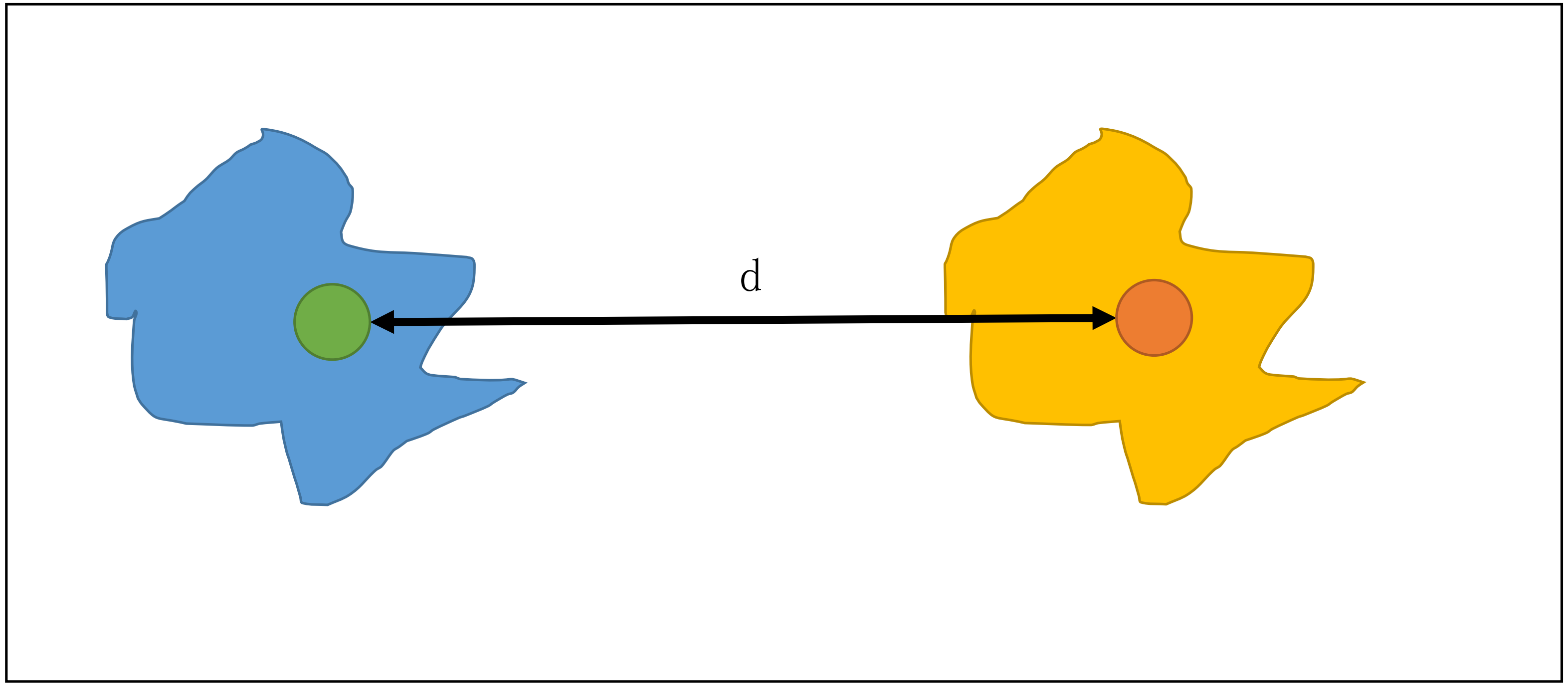


Maximum likelihood
(minimize KL($P_{data}||P_G$))

Minimize KL($P_G||P_{data}$)
(reverse KL)

EM距离 earth mover （Wasserstein 距离）

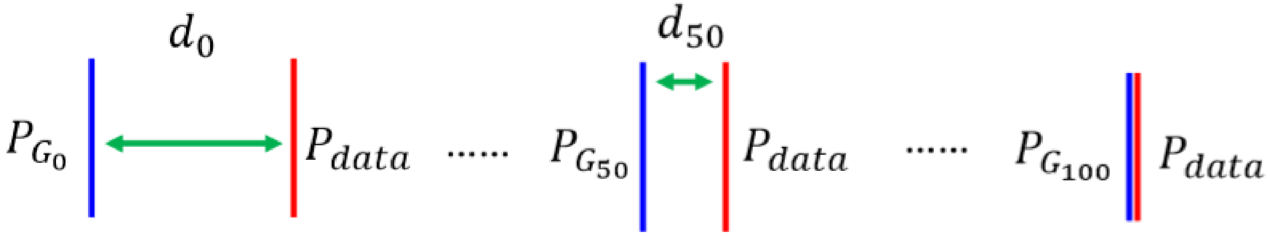
P, Q分布很远几乎无重叠的情况，仍能反映两个分布的远近



$$D_f(P_{data}||P_G)$$

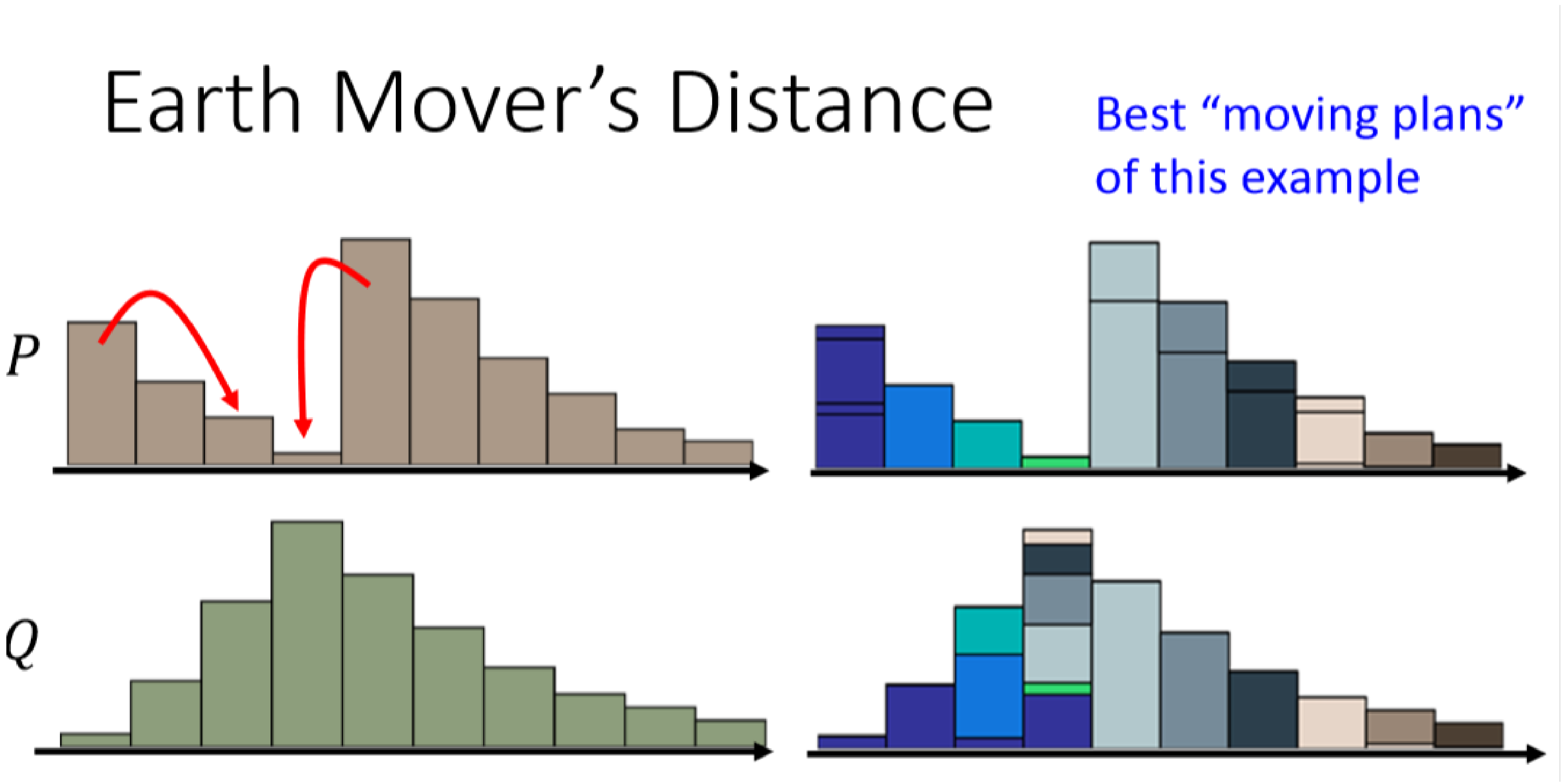
$$\Downarrow$$

$$W(P_{data}, P_G)$$



$JS(P_{G_0}, P_{data}) = log2$	$JS(P_{G_{50}}, P_{data}) = log2$	$JS(P_{G_{100}}, P_{data}) = 0$
$W(P_{G_0}, P_{data}) = d_0$	$W(P_{G_{50}}, P_{data}) = d_{50}$	$W(P_{G_{100}}, P_{data}) = 0$

形象表示



定义式

$$\mathcal{W}[p,q] = \inf_{\gamma\in\Pi[p,q]} \iint \gamma(\boldsymbol{x},\boldsymbol{y})\underline{d(\boldsymbol{x},\boldsymbol{y})}\,d\boldsymbol{x}\,d\boldsymbol{y}$$

d(x, y) ，它不一定是距离，其准确含义应该是一个成本函数，代表着从 x 运输到 y 的成本

定义式

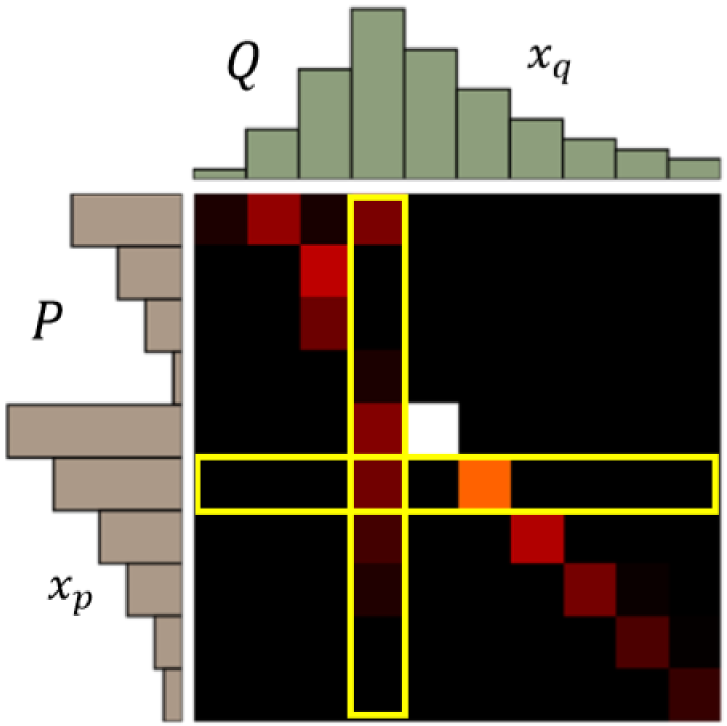
$$\mathcal{W}[p,q] = \inf_{\gamma\in\Pi[p,q]} \iint \gamma(\boldsymbol{x},\boldsymbol{y})\,d(\boldsymbol{x},\boldsymbol{y})\,d\boldsymbol{x}\,d\boldsymbol{y}$$

$$\int \gamma(\boldsymbol{x},\boldsymbol{y})d\boldsymbol{y} = p(\boldsymbol{x}) \quad \text{且} \quad \int \gamma(\boldsymbol{x},\boldsymbol{y})d\boldsymbol{x} = q(\boldsymbol{y})$$

γ 是一个联合分布，它的边缘分布就是原来的 p 和 q

$$\mathcal{W}[p,q] = \inf_{\gamma \in \Pi[p,q]} \iint \gamma(\boldsymbol{x},\boldsymbol{y}) d(\boldsymbol{x},\boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y}$$

下确界，简单来说就是取最小，也就是说，要从所有的运输方案中，找出总运输成本 ∬γ(x,y)d(x,y)dxdy 最小的方案，这个方案的成本，就是我们要算的 W[p,q]



moving plan γ
All possible plan Π

A "moving plan" is a matrix

The value of the element is the amount of earth from one position to another.

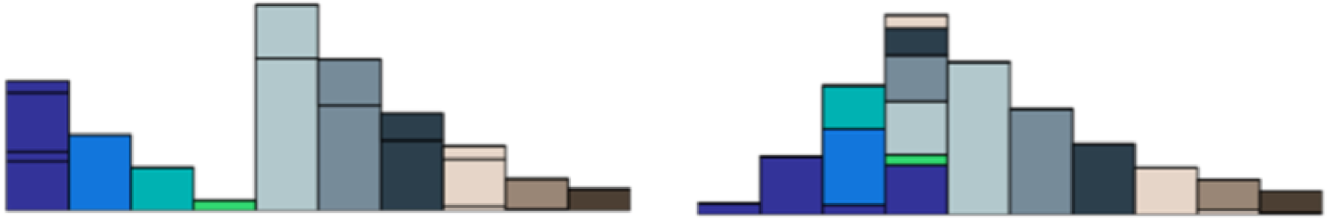
Average distance of a plan γ:

$$B(\gamma) = \sum_{x_p, x_q} \gamma(x_p, x_q) \|x_p - x_q\|$$

Earth Mover's Distance:

$$W(P, Q) = \min_{\gamma \in \Pi} B(\gamma)$$

The best plan



$$\mathcal{W}[p,q]=\inf_{\gamma\in\Pi[p,q]}\iint \gamma(\boldsymbol{x},\boldsymbol{y})d(\boldsymbol{x},\boldsymbol{y})d\boldsymbol{x}d\boldsymbol{y}$$

其实就是最小化 $$\iint \gamma(\boldsymbol{x},\boldsymbol{y})\underline{d(\boldsymbol{x},\boldsymbol{y})}d\boldsymbol{x}d\boldsymbol{y}$$

d（x, y）是定值

需要满足约束：$\int \gamma(\boldsymbol{x},\boldsymbol{y})d\boldsymbol{y}=p(\boldsymbol{x}),\quad \int \gamma(\boldsymbol{x},\boldsymbol{y})d\boldsymbol{x}=q(\boldsymbol{y}),\quad \gamma(\boldsymbol{x},\boldsymbol{y})\geq 0$

积分只是求和的极限形式，所以我们可以把 γ(x,y) 和 d(x,y) 离散化

相当于就是将 Γ 和 D 对应位置相乘，然后求和，这不就是内积 ⟨Γ,D⟩ 了吗

$$\boldsymbol{\Gamma}=\begin{pmatrix}\gamma(\boldsymbol{x}_1,\boldsymbol{y}_1)\\ \gamma(\boldsymbol{x}_1,\boldsymbol{y}_2)\\ \vdots\\ \hline \gamma(\boldsymbol{x}_2,\boldsymbol{y}_1)\\ \gamma(\boldsymbol{x}_2,\boldsymbol{y}_2)\\ \vdots\\ \hline \vdots\\ \hline \gamma(\boldsymbol{x}_n,\boldsymbol{y}_1)\\ \gamma(\boldsymbol{x}_n,\boldsymbol{y}_2)\\ \vdots\\ \hline \vdots\end{pmatrix},\quad \boldsymbol{D}=\begin{pmatrix}d(\boldsymbol{x}_1,\boldsymbol{y}_1)\\ d(\boldsymbol{x}_1,\boldsymbol{y}_2)\\ \vdots\\ \hline d(\boldsymbol{x}_2,\boldsymbol{y}_1)\\ d(\boldsymbol{x}_2,\boldsymbol{y}_2)\\ \vdots\\ \hline \vdots\\ \hline d(\boldsymbol{x}_n,\boldsymbol{y}_1)\\ d(\boldsymbol{x}_n,\boldsymbol{y}_2)\\ \vdots\\ \hline \vdots\end{pmatrix}$$

$$\int \gamma(\boldsymbol{x},\boldsymbol{y})d\boldsymbol{y} = p(\boldsymbol{x}), \quad \int \gamma(\boldsymbol{x},\boldsymbol{y})d\boldsymbol{x} = q(\boldsymbol{y}), \quad \gamma(\boldsymbol{x},\boldsymbol{y}) \geq 0$$

约束条件也可以写成矩阵形式 AΓ=b

$$\underbrace{\left(\begin{array}{ccc|ccc|c|ccc|c}
1 & 1 & \cdots & 0 & 0 & \cdots & \cdots & 0 & 0 & \cdots & \cdots \\
0 & 0 & \cdots & 1 & 1 & \cdots & \cdots & 0 & 0 & \cdots & \cdots \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots \\
0 & 0 & \cdots & 0 & 0 & \cdots & \cdots & 1 & 1 & \cdots & \cdots \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots \\
\hline
1 & 0 & \cdots & 1 & 0 & \cdots & \cdots & 1 & 0 & \cdots & \cdots \\
0 & 1 & \cdots & 0 & 1 & \cdots & \cdots & 0 & 1 & \cdots & \cdots \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots \\
0 & 0 & \cdots & 0 & 0 & \cdots & \cdots & 0 & 0 & \cdots & \cdots \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots
\end{array}\right)}_{\boldsymbol{A}}
\underbrace{\left(\begin{array}{c}
\gamma(\boldsymbol{x}_1,\boldsymbol{y}_1) \\
\gamma(\boldsymbol{x}_1,\boldsymbol{y}_2) \\
\vdots \\
\hline
\gamma(\boldsymbol{x}_2,\boldsymbol{y}_1) \\
\gamma(\boldsymbol{x}_2,\boldsymbol{y}_2) \\
\vdots \\
\hline
\vdots \\
\hline
\gamma(\boldsymbol{x}_n,\boldsymbol{y}_1) \\
\gamma(\boldsymbol{x}_n,\boldsymbol{y}_2) \\
\vdots \\
\hline
\vdots
\end{array}\right)}_{\boldsymbol{\Gamma}}
=
\underbrace{\left(\begin{array}{c}
p(\boldsymbol{x}_1) \\
p(\boldsymbol{x}_2) \\
\vdots \\
p(\boldsymbol{x}_n) \\
\vdots \\
\hline
q(\boldsymbol{y}_1) \\
q(\boldsymbol{y}_2) \\
\vdots \\
q(\boldsymbol{y}_n) \\
\vdots
\end{array}\right)}_{\boldsymbol{b}}
\Longrightarrow
\min_{\boldsymbol{\Gamma}} \{ \langle \boldsymbol{\Gamma}, \boldsymbol{D} \rangle \mid \boldsymbol{A\Gamma} = \boldsymbol{b}, \boldsymbol{\Gamma} \geq 0 \}$$

线性约束下的线性函数最小值

线性约束下的线性函数最小值

$$\min_{\boldsymbol{x}} \{\boldsymbol{c}^\top \boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},\, \boldsymbol{x} \geq 0\}$$

弱对偶形式

设置最小值在 x^* 取到

两边乘以一个 $y^\top \in R^m$ $\longrightarrow$ $y^\top A x^* \in y^\top b$

此时假设 $y^\top A \leq c^\top$ $\longrightarrow$ $y^\top A x^* \leq c^\top x^*$ 因为 $x^* > 0$

$$y^\top b \leq c^\top x^*$$

在条件下 $y^\top A \leq c^\top$ $\longrightarrow$ $\max_{\boldsymbol{y}} \{\boldsymbol{b}^\top \boldsymbol{y} \mid \boldsymbol{A}^\top \boldsymbol{y} \leq \boldsymbol{c}\} \leq \min_{\boldsymbol{x}} \{\boldsymbol{c}^\top \boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},\, \boldsymbol{x} \geq 0\}$

现在我们将原来的最小值问题变成了一个最大值问题，这便有了对偶的味道。
从应用角度，其实弱对偶形式给出的下界都已经够用了，
因为深度学习中的问题都很复杂，能有一个近似的目标去优化都已经很不错了。

$$\max_{\boldsymbol{y}} \{\boldsymbol{b}^\top \boldsymbol{y} \mid \boldsymbol{A}^\top \boldsymbol{y} \le \boldsymbol{c}\} \le \min_{\boldsymbol{x}} \{\boldsymbol{c}^\top \boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},\, \boldsymbol{x} \ge 0\}$$

$$\min_{\boldsymbol{\Gamma}} \{\langle \boldsymbol{\Gamma}, \boldsymbol{D}\rangle \mid \boldsymbol{A}\boldsymbol{\Gamma} = \boldsymbol{b},\, \boldsymbol{\Gamma} \ge 0\} = \max_{\boldsymbol{F}} \{\langle \boldsymbol{b}, \boldsymbol{F}\rangle \mid \boldsymbol{A}^\top \boldsymbol{F} \le \boldsymbol{D}\}$$

b 是由两部分拼起来的，所以我们也可以把 F 类似地写成

$$\underbrace{\left(\begin{array}{ccc|ccc|c|ccc|c}
1 & 1 & \cdots & 0 & 0 & \cdots & \cdots & 0 & 0 & \cdots & \cdots \\
0 & 0 & \cdots & 1 & 1 & \cdots & \cdots & 0 & 0 & \cdots & \cdots \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots \\
0 & 0 & \cdots & 0 & 0 & \cdots & \cdots & 1 & 1 & \cdots & \cdots \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots \\
\hline
1 & 0 & \cdots & 1 & 0 & \cdots & \cdots & 1 & 0 & \cdots & \cdots \\
0 & 1 & \cdots & 0 & 1 & \cdots & \cdots & 0 & 1 & \cdots & \cdots \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots \\
0 & 0 & \cdots & 0 & 0 & \cdots & \cdots & 0 & 0 & \cdots & \cdots \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots
\end{array}\right)}_{\boldsymbol{A}}
\underbrace{\left(\begin{array}{c}
\gamma(\boldsymbol{x}_1, \boldsymbol{y}_1) \\
\gamma(\boldsymbol{x}_1, \boldsymbol{y}_2) \\
\vdots \\
\hline
\gamma(\boldsymbol{x}_2, \boldsymbol{y}_1) \\
\gamma(\boldsymbol{x}_2, \boldsymbol{y}_2) \\
\vdots \\
\hline
\vdots \\
\hline
\gamma(\boldsymbol{x}_n, \boldsymbol{y}_1) \\
\gamma(\boldsymbol{x}_n, \boldsymbol{y}_2) \\
\vdots \\
\hline
\vdots
\end{array}\right)}_{\boldsymbol{\Gamma}}
=
\underbrace{\left(\begin{array}{c}
p(\boldsymbol{x}_1) \\
p(\boldsymbol{x}_2) \\
\vdots \\
p(\boldsymbol{x}_n) \\
\vdots \\
\hline
q(\boldsymbol{y}_1) \\
q(\boldsymbol{y}_2) \\
\vdots \\
q(\boldsymbol{y}_n) \\
\vdots
\end{array}\right)}_{\boldsymbol{b}}$$

$$\boldsymbol{F} = \left(\begin{array}{c}
f(\boldsymbol{x}_1) \\
f(\boldsymbol{x}_2) \\
\vdots \\
f(\boldsymbol{x}_n) \\
\vdots \\
\hline
g(\boldsymbol{y}_1) \\
g(\boldsymbol{y}_2) \\
\vdots \\
g(\boldsymbol{y}_n) \\
\vdots
\end{array}\right)$$

现在<b , F>
是什么?

$$\langle \boldsymbol{b}, \boldsymbol{F}\rangle = \sum_n p(\boldsymbol{x}_n) f(\boldsymbol{x}_n) + \sum_n q(\boldsymbol{x}_n) g(\boldsymbol{x}_n)$$

$$\langle \boldsymbol{b}, \boldsymbol{F} \rangle = \sum_n p(\boldsymbol{x}_n) f(\boldsymbol{x}_n) + \sum_n q(\boldsymbol{x}_n) g(\boldsymbol{x}_n) \longrightarrow \langle \boldsymbol{b}, \boldsymbol{F} \rangle = \int \big[p(\boldsymbol{x}) f(\boldsymbol{x}) + q(\boldsymbol{x}) g(\boldsymbol{x})\big] d\boldsymbol{x}$$

约束条件：

$$\underbrace{\left(\begin{array}{ccccc|ccccc}
1 & 0 & \dots & 0 & \dots & 1 & 0 & \dots & 0 & \dots \\
1 & 0 & \dots & 0 & \dots & 0 & 1 & \dots & 0 & \dots \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots \\
\hline
0 & 1 & \dots & 0 & \dots & 1 & 0 & \dots & 0 & \dots \\
0 & 1 & \dots & 0 & \dots & 0 & 1 & \dots & 0 & \dots \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots \\
\hline
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots \\
\hline
0 & 0 & \dots & 1 & \dots & 1 & 0 & \dots & 0 & \dots \\
0 & 0 & \ddots & 1 & \ddots & 0 & 1 & \ddots & 0 & \ddots \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots \\
\hline
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots
\end{array}\right)}_{\boldsymbol{A}^\top}
\underbrace{\left(\begin{array}{c}
f(\boldsymbol{x}_1) \\ f(\boldsymbol{x}_2) \\ \vdots \\ f(\boldsymbol{x}_n) \\ \vdots \\
\hline
g(\boldsymbol{y}_1) \\ g(\boldsymbol{y}_2) \\ \vdots \\ g(\boldsymbol{y}_n) \\ \vdots
\end{array}\right)}_{\boldsymbol{F}}
\leq
\underbrace{\left(\begin{array}{c}
d(\boldsymbol{x}_1, \boldsymbol{y}_1) \\ d(\boldsymbol{x}_1, \boldsymbol{y}_2) \\ \vdots \\
\hline
d(\boldsymbol{x}_2, \boldsymbol{y}_1) \\ d(\boldsymbol{x}_2, \boldsymbol{y}_2) \\ \vdots \\
\hline
\vdots \\
\hline
d(\boldsymbol{x}_n, \boldsymbol{y}_1) \\ d(\boldsymbol{x}_n, \boldsymbol{y}_2) \\ \vdots \\
\hline
\vdots
\end{array}\right)}_{\boldsymbol{D}}$$

实际上就是： $\forall i, j,\ f(\boldsymbol{x}_i) + g(\boldsymbol{y}_j) \leq d(\boldsymbol{x}_i, \boldsymbol{y}_j) \longrightarrow \forall \boldsymbol{x}, \boldsymbol{y},\ f(\boldsymbol{x}) + g(\boldsymbol{y}) \leq d(\boldsymbol{x}, \boldsymbol{y})$

$$\mathcal{W}[p,q] = \inf_{\gamma \in \Pi[p,q]} \iint \gamma(\boldsymbol{x},\boldsymbol{y}) d(\boldsymbol{x},\boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y}$$

$$\Downarrow$$

$$\mathcal{W}[p,q] = \max_{f,g} \left\{ \int [p(\boldsymbol{x})f(\boldsymbol{x}) + q(\boldsymbol{x})g(\boldsymbol{x})]\, d\boldsymbol{x} \,\middle|\, f(\boldsymbol{x}) + g(\boldsymbol{y}) \leq d(\boldsymbol{x},\boldsymbol{y}) \right\}$$

$$\mathcal{W}[p,q] = \max_{f,g}\left\{\int [p(\boldsymbol{x})f(\boldsymbol{x}) + q(\boldsymbol{x})g(\boldsymbol{x})]d\boldsymbol{x} \,\middle|\, f(\boldsymbol{x}) + g(\boldsymbol{y}) \le d(\boldsymbol{x},\boldsymbol{y})\right\}$$

$$f(\boldsymbol{x}) + g(\boldsymbol{x}) \le d(\boldsymbol{x},\boldsymbol{x}) = 0$$

$$\begin{aligned} p(\boldsymbol{x})f(\boldsymbol{x}) + q(\boldsymbol{x})g(\boldsymbol{x}) &\le p(\boldsymbol{x})f(\boldsymbol{x}) + q(\boldsymbol{x})[-f(\boldsymbol{x})] \\ &= p(\boldsymbol{x})f(\boldsymbol{x}) - q(\boldsymbol{x})f(\boldsymbol{x}) \end{aligned}$$

$$\mathcal{W}[p,q] = \max_{f}\left\{\int [p(\boldsymbol{x})f(\boldsymbol{x}) - q(\boldsymbol{x})f(\boldsymbol{x})]d\boldsymbol{x} \,\middle|\, f(\boldsymbol{x}) - f(\boldsymbol{y}) \le d(\boldsymbol{x},\boldsymbol{y})\right\}$$

$$\mathcal{W}[p,q] = \max_{f,\,\|f\|_L\le 1} \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}[f(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}[f(\boldsymbol{x})] \Longrightarrow \min_{G}\max_{f,\,\|f\|_L\le 1} \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}[f(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z})}[f(G(\boldsymbol{z}))]$$

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

Lipschitz Function

$$\|f(x_1) - f(x_2)\| \leq K\|x_1 - x_2\|$$

Output change | Input change

K=1 for "$1 - Lipschitz$"

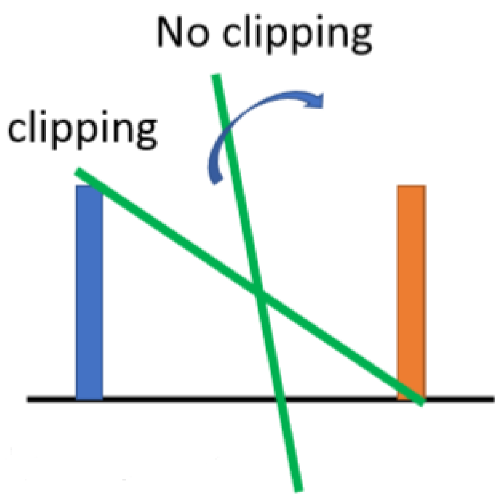
Do not change fast

问题

在计算梯度的时候，怎么解决约束问题？

（1）weight clipping

强制把更新后的参数固定在[-c , c]



if w > c, then w=c; if w<-c, then w=-c

Algorithm of WGAN

- In each training iteration: **No sigmoid for the output of D**

Learning D

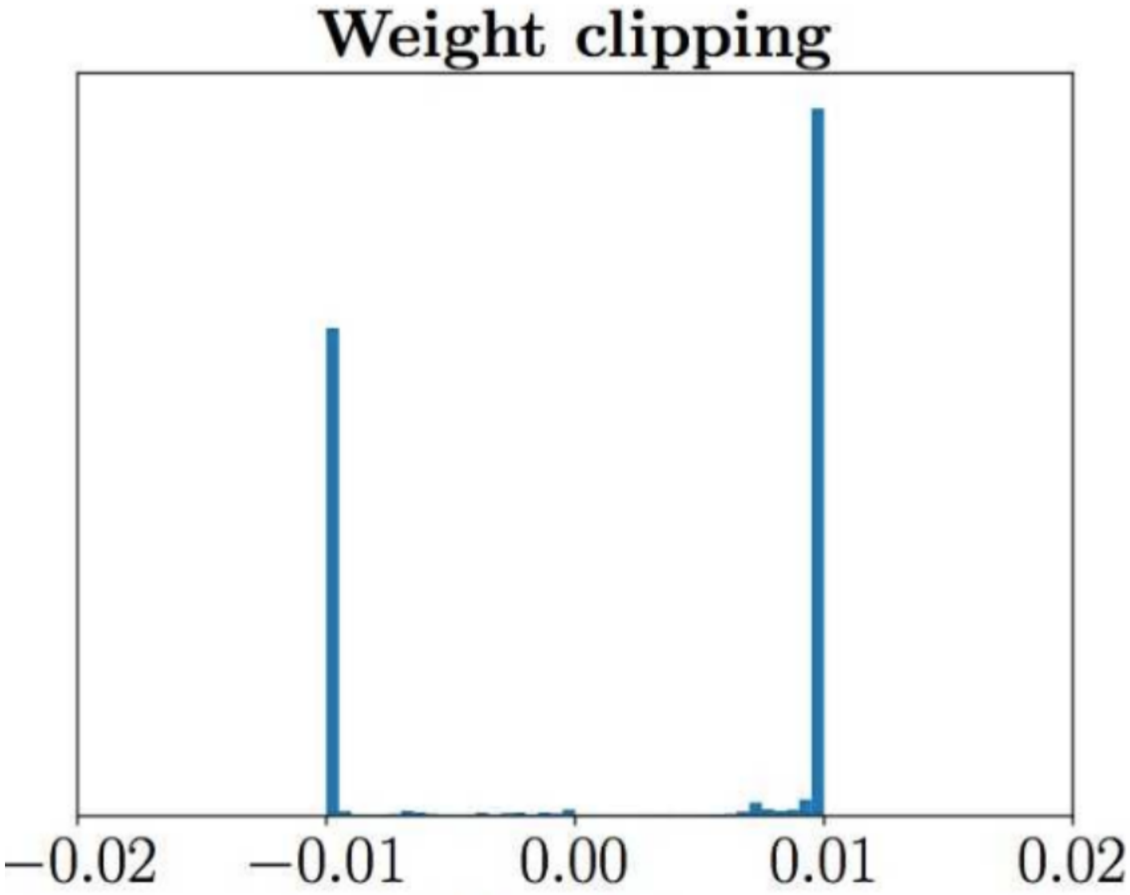
Repeat k times

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

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



很容易一不小心就梯度消失或者梯度爆炸。原因是判别器是一个多层网络，如果把clipping threshold设得稍微小了一 点，每经过一层网络，梯度就变小一点点，多层之后就会指数衰减；反之，如果设得稍微大了一点，每经过一层网络，梯度变大一点点，多层之后就会指数爆 炸。只有设得不大不小，才能让生成器获得恰到好处的回传梯度，然而在实际应用中这个平衡区域可能很狭窄，就会给调参工作带来麻烦。

（2）WGAN-GP（gradient penalty）

$$D \in 1 - Lipschitz \iff \|\nabla_x D(x)\| \leq 1 \text{ for all x}$$

$$\Downarrow$$

$$W(P_{data}, P_G) \approx \max_D \{E_{x\sim P_{data}}[D(x)] - E_{x\sim P_G}[D(x)]$$

$$-\lambda \int_x max(0, \|\nabla_x D(x)\| - 1)dx\}$$

让梯度尽可能的接近1

$$\Downarrow$$

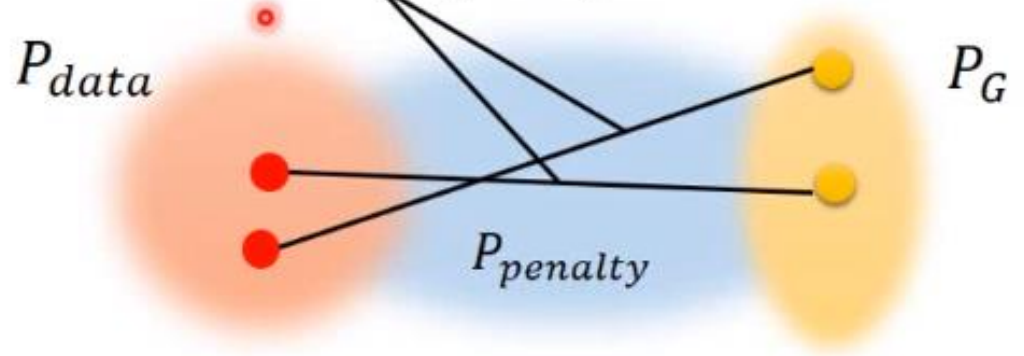
不能全部x都算，要采样

$$-\lambda E_{x\sim P_{penalty}}[\underline{max(0, \|\nabla_x D(x)\| - 1)}]\}$$

$$\Downarrow$$

$$(\|\nabla_x D(x)\| - 1)^2$$

$$W(P_{data}, P_G) \approx \max_D \{E_{x\sim P_{data}}[D(x)] - E_{x\sim P_G}[D(x)]$$
$$-\lambda E_{x\sim P_{penalty}}[max(0, \|\nabla_x D(x)\| - 1)]\}$$



谢谢！