生成对抗网络(GAN与W-GAN)

作者：凯鲁嘎吉 - 博客园 http://www.cnblogs.com/kailugaji/

通过阅读《神经网络与深度学习》，了解生成对抗网络(Generative Adversarial Networks，GAN)的来龙去脉，并介绍GAN与Wasserstein GAN。

1．基础知识

KL散度 (Kullback–Leibler Divergence)、JS散度 (Jensen–Shannon Divergence)、推土机距离 (Wasserstein Distance, or Earth-Mover's Distance)以及Lipschitz连续函数.

➢ 基础知识

- KL散度 (Kullback–Leibler Divergence)

$$KL(p,q)=\sum_{i=1}^{N}p(x_i)\log\frac{p(x_i)}{q(x_i)}$$

- JS散度 (Jensen–Shannon Divergence)

$$JS(p,q)=\frac{1}{2}KL\left(p,\frac{p+q}{2}\right)+\frac{1}{2}KL\left(q,\frac{p+q}{2}\right)$$

- 推土机距离 (Wasserstein Distance, or Earth-Mover's Distance)

Wasserstein-ℓ:

$$W_l[p,q]=\left(\inf_{\gamma\in\Pi(p,q)}\mathbb{E}_{(x,y)\sim\gamma(x,y)}[d(x,y)^l]\right)^{\frac{1}{l}}$$

其中$\Pi(p,q)$是边际分布为p和q的所有可能的联合分布集合，$d(x,y)$为x和y的距离，比如ℓ_2距离等.

- Lipschitz连续函数

在数学中，对于一个实数函数$f:\mathbb{R}\to\mathbb{R}$，如果满足函数曲线上任意两点连线的斜率一致有界，即任意两点的斜率都小于常数$K>0$，

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

则函数f就称为K-Lipschitz连续函数，K称为Lipschitz常数.
Lipschitz 连续要求函数在无限的区间上不能有超过线性的增长．如果一个函数可导，并满足Lipschitz连续，那么导数有界．如果一个函数可导，并且导数有界，那么函数为Lipschitz连续.

1

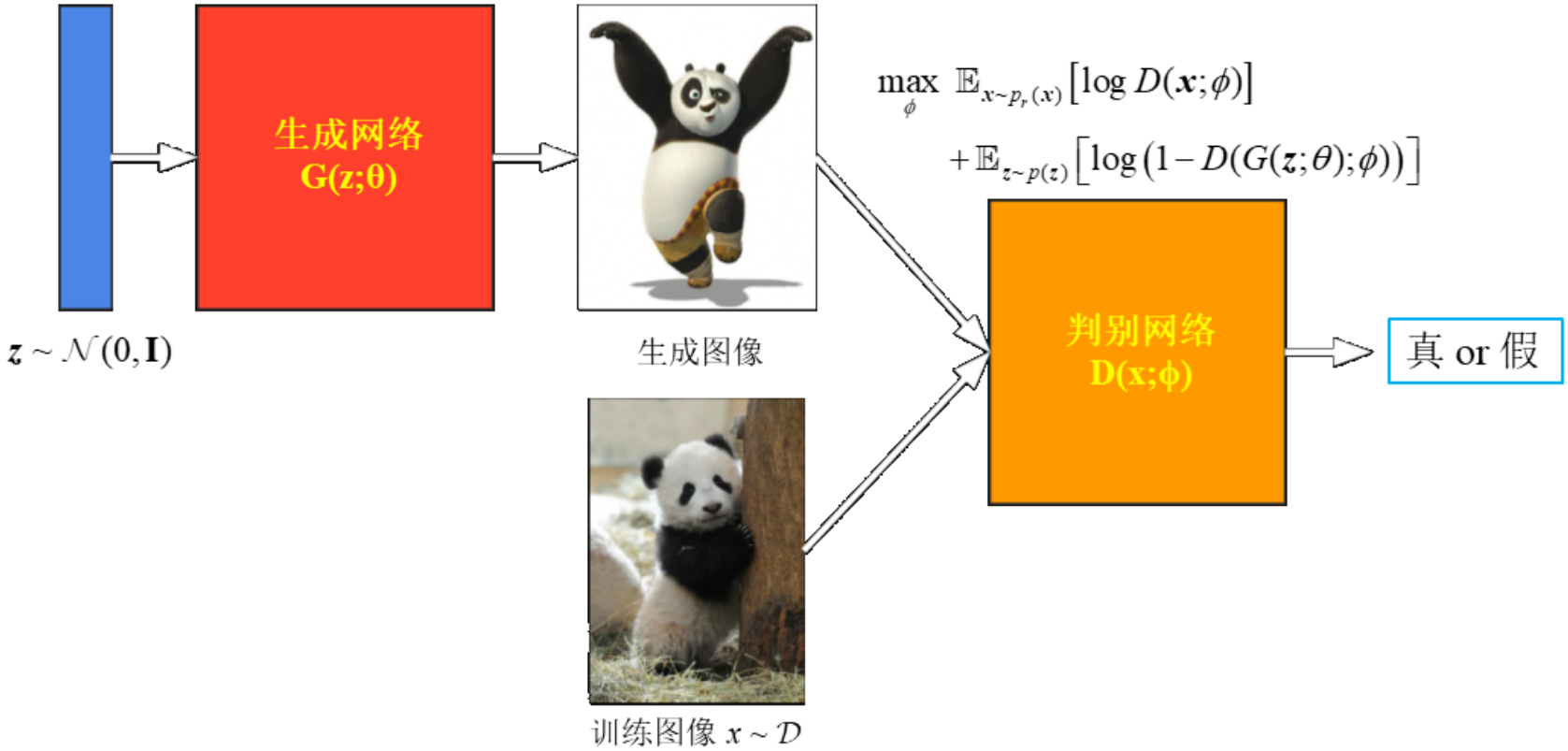
2. Vanilla GAN (标准GAN/原始GAN)

Vanilla GAN

$$\min_{\theta}\max_{\phi}\ \mathbb{E}_{x\sim p_r(x)}\left[\log D(\boldsymbol{x};\phi)\right]+\mathbb{E}_{z\sim p(z)}\left[\log\left(1-D(G(\boldsymbol{z};\theta);\phi)\right)\right]$$

$$\max_{\theta}\ \mathbb{E}_{z\sim p(z)}\left[\log\left(D(G(\boldsymbol{z};\theta);\phi)\right)\right]$$

$$\Leftrightarrow \min_{\theta}\ \mathbb{E}_{z\sim p(z)}\left[\log\left(1-D(G(\boldsymbol{z};\theta);\phi)\right)\right]$$



判别网络，生成网络，以及总体目标函数

➢ Vanilla GAN

- 判别网络(Discriminator Network)
  - ✓ 判别网络$D(\boldsymbol{x};\phi)$的目标是区分出一个样本$\boldsymbol{x}$是来自于真实分布$p_r(\boldsymbol{x})$还是来自于生成模型$p_\theta(\boldsymbol{x})$. 用标签y=1来表示样本来自真实分布，y=0表示样本来自生成模型，判别网络$D(\boldsymbol{x};\phi)$的输出为$\boldsymbol{x}$属于真实数据分布的概率，即$p(y=1|\boldsymbol{x})=D(\boldsymbol{x};\phi)$, 则样本来自生成模型的概率为$p(y=0|\boldsymbol{x})=1-D(\boldsymbol{x};\phi)$.
  - ✓ 给定一个样本$(\boldsymbol{x}, y)$，$y=\{1, 0\}$表示其来自于$p_r(\boldsymbol{x})$还是$p_\theta(\boldsymbol{x})$, 判别网络的目标函数为最小化交叉熵，即

$$\min_{\phi} -\left(\mathbb{E}_x\left[y\log p(y=1|\boldsymbol{x})+(1-y)\log p(y=0|\boldsymbol{x})\right]\right)$$

  - ✓ 假设分布$p(\boldsymbol{x})$是由分布$p_r(\boldsymbol{x})$和分布$p_\theta(\boldsymbol{x})$等比例混合而成，即$p(\boldsymbol{x})=(p_r(\boldsymbol{x})+p_\theta(\boldsymbol{x}))/2$，则上式等价于

$$\max_{\phi}\ \mathbb{E}_{x\sim p_r(x)}\left[\log D(\boldsymbol{x};\phi)\right]+\mathbb{E}_{x'\sim p_\theta(x')}\left[\log\left(1-D(\boldsymbol{x}';\phi)\right)\right]\Leftrightarrow\max_{\phi}\ \mathbb{E}_{x\sim p_r(x)}\left[\log D(\boldsymbol{x};\phi)\right]+\mathbb{E}_{z\sim p(z)}\left[\log\left(1-D(G(\boldsymbol{z};\theta);\phi)\right)\right]$$

  其中θ和ϕ分别是生成网络和判别网络的参数.
- 生成网络(Generator Network)
  - ✓ 生成网络$G(\boldsymbol{z};\theta)$的目标是让判别网络将自己生成的样本判别为真实样本.

$$\max_{\theta}\ \mathbb{E}_{z\sim p(z)}\left[\log\left(D(G(\boldsymbol{z};\theta);\phi)\right)\right]\Leftrightarrow\min_{\theta}\ \mathbb{E}_{z\sim p(z)}\left[\log\left(1-D(G(\boldsymbol{z};\theta);\phi)\right)\right]$$

- 总体目标函数看做最小化最大化游戏(Minimax Game)

$$\min_{\theta}\max_{\phi}\ \left(\mathbb{E}_{x\sim p_r(x)}\left[\log D(\boldsymbol{x};\phi)\right]+\mathbb{E}_{x\sim p_\theta(x)}\left[\log\left(1-D(\boldsymbol{x};\phi)\right)\right]\right)$$

$$\Leftrightarrow\min_{\theta}\max_{\phi}\ \left(\mathbb{E}_{x\sim p_r(x)}\left[\log D(\boldsymbol{x};\phi)\right]+\mathbb{E}_{z\sim p(z)}\left[\log\left(1-D(G(\boldsymbol{z};\theta);\phi)\right)\right]\right)$$

最小化交叉熵就是极大似然估计，使其期望最大化。

Vanilla GAN训练过程

➢ Vanilla GAN训练过程

- 对于生成网络，在实际训练时一般使用下面公式，因为其梯度性质更好.

$$\max_{\theta} \mathbb{E}_{z\sim p(z)}\left[\log\left(D(G(z;\theta);\phi)\right)\right]$$

- 函数$\log(x)$, $x\in(0,1)$在x 接近1时的梯度要比接近0时的梯度小很多，接近“饱和”区间．这样，当判别网络D以很高的概率认为生成网络G产生的样本是“假”样本，即$(1-D(G(\boldsymbol{z};\ \theta);\ \phi))\to 1$，这时目标函数关于$\theta$的梯度反而很小，从而不利于优化.
- 还有一种改进生成网络的梯度的方法是将真实样本和生成样本的标签互换，即生成样本的标签为1.
- 每次迭代时，判别网络更新K次而生成网络更新一次，即首先要保证判别网络足够强才能开始训练生成网络.
- 在实践中K是个超参数，其取值一般取决于具体任务.

输入: 训练集$\mathcal{D}$,对抗训练迭代次数T,每次判别网络的训练迭代次数K,小批量样本数量M

随机初始化θ,ϕ;

for $t\leftarrow 1$ **to** T **do**

　// 训练判别网络$D(\boldsymbol{x};\phi)$

　for $k\leftarrow 1$ **to** K **do**

　　// 采集小批量训练样本

　　从训练集$\mathcal{D}$中采集M个样本$\{\boldsymbol{x}^{(m)}\}, 1\le m\le M$;

　　从分布$\mathcal{N}(0,\boldsymbol{I})$中采集$M$个样本$\{\boldsymbol{z}^{(m)}\}, 1\le m\le M$;

　　使用随机梯度上升更新ϕ,梯度为

$$\frac{\partial}{\partial\phi}\left[\frac{1}{M}\sum_{m=1}^{M}\left(\log D(\boldsymbol{x}^{(m)};\phi)+\log\left(1-D(G(\boldsymbol{z}^{(m)};\theta);\phi)\right)\right)\right];$$

　end

　// 训练生成网络$G(\boldsymbol{z};\theta)$

　从分布$\mathcal{N}(0,\boldsymbol{I})$中采集$M$个样本$\{\boldsymbol{z}^{(m)}\}, 1\le m\le M$;

　使用随机梯度上升更新θ,梯度为

$$\frac{\partial}{\partial\theta}\left[\frac{1}{M}\sum_{m=1}^{M}D(G(\boldsymbol{z}^{(m)};\theta),\phi)\right];$$

end

输出: 生成网络$G(\boldsymbol{z};\theta)$

Vanilla GAN进一步分析——梯度消失，分布不重叠时，JS散度恒为log2

➢ Vanilla GAN进一步分析

- 假设$p_r(\boldsymbol{x})$和$p_\theta(\boldsymbol{x})$已知，则最优的判别器为 $D^*(\boldsymbol{x})=\frac{p_r(\boldsymbol{x})}{p_r(\boldsymbol{x})+p_\theta(\boldsymbol{x})}$
- 将最优的判别器$D^*(\boldsymbol{x})$代入公式 $\min_\theta \max_\phi \left(\mathbb{E}_{x\sim p_r(x)}\left[\log D(\boldsymbol{x};\phi)\right]+\mathbb{E}_{x\sim p_\theta(x)}\left[\log\left(1-D(\boldsymbol{x};\phi)\right)\right]\right)$，其目标函数变为

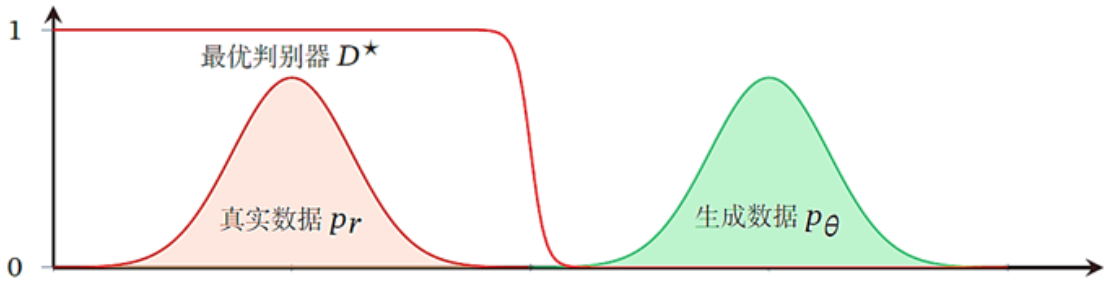
$$\mathcal{L}(G\mid D^*)=\mathbb{E}_{x\sim p_r(x)}\left[\log D^*(\boldsymbol{x})\right]+\mathbb{E}_{x\sim p_\theta(x)}\left[\log\left(1-D^*(\boldsymbol{x})\right)\right]=\mathbb{E}_{x\sim p_r(x)}\left[\log\frac{p_r(\boldsymbol{x})}{p_r(\boldsymbol{x})+p_\theta(\boldsymbol{x})}\right]+\mathbb{E}_{x\sim p_\theta(x)}\left[\log\frac{p_\theta(\boldsymbol{x})}{p_r(\boldsymbol{x})+p_\theta(\boldsymbol{x})}\right]$$

$$=\mathbb{E}_{x\sim p_r(x)}\left[\log\frac{p_r(\boldsymbol{x})}{2p_a(\boldsymbol{x})}\right]+\mathbb{E}_{x\sim p_\theta(x)}\left[\log\frac{p_\theta(\boldsymbol{x})}{2p_a(\boldsymbol{x})}\right]=KL(p_r,p_a)+KL(p_\theta,p_a)-2\log 2=2JS(p_r,p_\theta)-2\log 2$$

其中$p_a(\boldsymbol{x})=\frac{p_r(\boldsymbol{x})+p_\theta(\boldsymbol{x})}{2}$为一个“平均”分布.

- 在Vanilla GAN中，当判别网络最优时，生成网络的优化目标是最小化真实分布p_r和模型分布p_θ之间的JS散度．当两个分布相同时，JS散度为0，最优生成网络G^*对应的损失为$\mathcal{L}(G^*|D^*)=-2\log 2$.
- 使用JS散度来训练生成对抗网络的一个问题是当真实分布p_r和模型分布p_θ没有重叠时，它们之间的JS散度恒等于常数log2, 最优的判别器D^*对所有生成数据的输出都为0，即$D^*(G(\boldsymbol{z};\theta))=0, \forall \boldsymbol{z}$. 对生成网络来说，目标函数关于参数的梯度为0，即 $\frac{\partial\mathcal{L}(G\mid D^*)}{\partial\theta}=0.$ 因此，生成网络的梯度消失.

判别器越好，生成器梯度消失越严重



➢ Vanilla GAN进一步分析

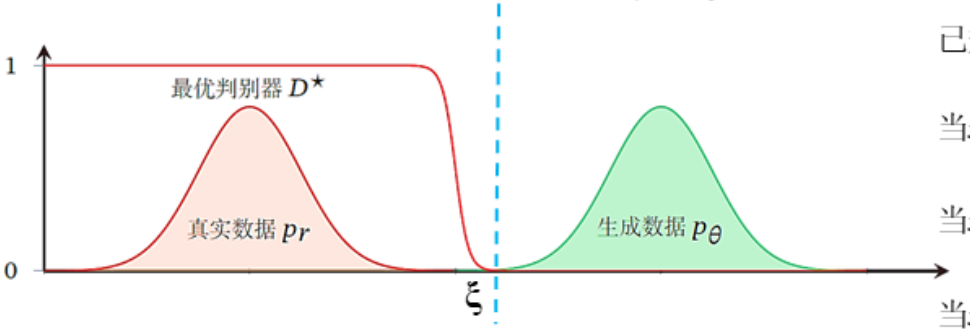
- 当生成器G固定，对于样本$\boldsymbol{x}$，它的目标函数为 $F = p_r(\boldsymbol{x})\log D(\boldsymbol{x}) + p_\theta(\boldsymbol{x})\log[1-D(\boldsymbol{x})]$

$$\frac{\partial F}{\partial D} = \frac{p_r(\boldsymbol{x})}{D(\boldsymbol{x})} - \frac{p_\theta(\boldsymbol{x})}{1-D(\boldsymbol{x})} = 0,\ \therefore D^*(\boldsymbol{x}) = \frac{p_r(\boldsymbol{x})}{p_r(\boldsymbol{x}) + p_\theta(\boldsymbol{x})}$$

- 假设$p_r(\boldsymbol{x})$和$p_\theta(\boldsymbol{x})$已知且互不重叠，$p_a(\boldsymbol{x}) = \frac{p_r(\boldsymbol{x}) + p_\theta(\boldsymbol{x})}{2}$.

$$2JS(p_r, p_\theta) = KL(p_r, p_a) + KL(p_\theta, p_a) = KL(p_r, \frac{p_r + p_\theta}{2}) + KL(p_\theta, \frac{p_r + p_\theta}{2}) = \sum p_r \log\frac{2p_r}{p_r + p_\theta} + \sum p_\theta \log\frac{2p_\theta}{p_r + p_\theta}$$

$$= \sum p_r \log 2 + \sum p_r \log\frac{p_r}{p_r + p_\theta} + \sum p_\theta \log 2 + \sum p_\theta \log\frac{p_\theta}{p_r + p_\theta} = 2\log 2 + \sum p_r \log\frac{p_r}{p_r + p_\theta} + \sum p_\theta \log\frac{p_\theta}{p_r + p_\theta}$$



- 假设$p_r(\boldsymbol{x})$和$p_\theta(\boldsymbol{x})$两个分布不重叠，则必存在一点$\boldsymbol{x}_0 = \xi$，使得$p_r(\boldsymbol{x}_0)=0$且$p_\theta(\boldsymbol{x}_0)=0$.
- 当$\boldsymbol{x}<\xi$时，$p_r(\boldsymbol{x})=0$; 当$\boldsymbol{x}>\xi$时，$p_\theta(\boldsymbol{x})=0$.

已规定$0\log 0 = 0,\ 0\log\frac{0}{0} = 0.$

当$x<\xi$时，$\sum p_r \log\frac{p_r}{p_r+p_\theta} + \sum p_\theta \log\frac{p_\theta}{p_r+p_\theta} = \sum 0\log\frac{0}{0+p_\theta} + \sum p_\theta \log\frac{p_\theta}{p_\theta} = 0;$

当$x>\xi$时，$\sum p_r \log\frac{p_r}{p_r+p_\theta} + \sum p_\theta \log\frac{p_\theta}{p_r+p_\theta} = \sum p_r \log\frac{p_r}{p_r} + \sum 0\log\frac{0}{p_r+0} = 0;$

当$x=\xi$时，$\sum p_r \log\frac{p_r}{p_r+p_\theta} + \sum p_\theta \log\frac{p_\theta}{p_r+p_\theta} = 0.$

因此，当p_r与p_θ互不重叠时，在整个数域上，$JS(p_r, p_\theta) \triangleq \log 2$

Vanilla GAN进一步分析——逆向KL散度导致模型坍塌

➢ Vanilla GAN进一步分析

- 将最优判别器D^*代入生成网络的目标函数 $\max_\theta \mathbb{E}_{z\sim p(z)}\left[\log\left(D(G(\boldsymbol{z};\theta);\phi)\right)\right] \Leftrightarrow \max_\theta \mathbb{E}_{x\sim p_\theta(x)}\left[\log\left(D(\boldsymbol{x};\phi)\right)\right]$，得到

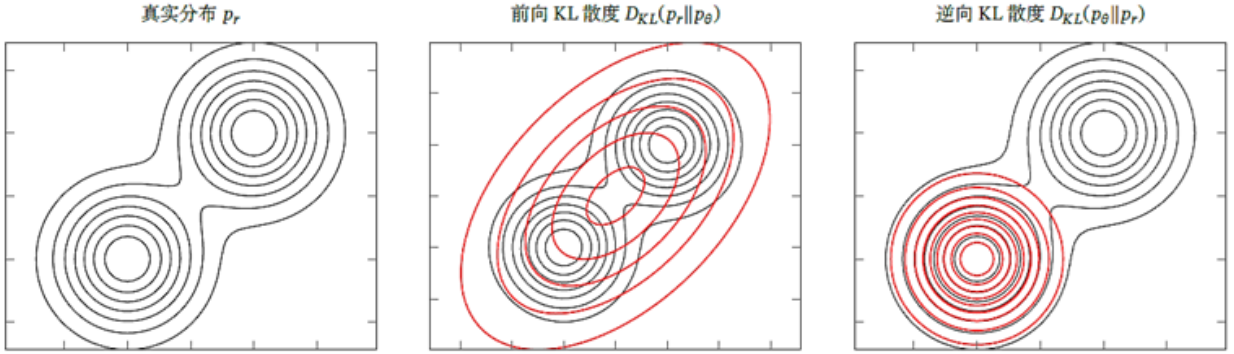
$$\mathcal{J}(G\mid D^*)=\mathbb{E}_{x\sim p_\theta(x)}\left[\log D^*(\boldsymbol{x})\right]=\mathbb{E}_{x\sim p_\theta(x)}\left[\log\left(\frac{p_r(\boldsymbol{x})}{p_r(\boldsymbol{x})+p_\theta(\boldsymbol{x})}\cdot\frac{p_\theta(\boldsymbol{x})}{p_\theta(\boldsymbol{x})}\right)\right]=\mathbb{E}_{x\sim p_\theta(x)}\left[\log\left(\frac{p_r(\boldsymbol{x})}{p_\theta(\boldsymbol{x})}\cdot\frac{p_\theta(\boldsymbol{x})}{p_r(\boldsymbol{x})+p_\theta(\boldsymbol{x})}\right)\right]$$

$$=-\mathbb{E}_{x\sim p_\theta(x)}\left[\log\frac{p_\theta(\boldsymbol{x})}{p_r(\boldsymbol{x})}\right]+\mathbb{E}_{x\sim p_\theta(x)}\left[\log\frac{p_\theta(\boldsymbol{x})}{p_r(\boldsymbol{x})+p_\theta(\boldsymbol{x})}\right]=-KL(p_\theta,p_r)+\mathbb{E}_{x\sim p_\theta(x)}\left[\log\left(1-D^*(\boldsymbol{x})\right)\right]$$

$$=-KL(p_\theta,p_r)+2JS(p_r,p_\theta)\boxed{-2\log 2-\mathbb{E}_{x\sim p_r(x)}\left[\log D^*(\boldsymbol{x})\right]}$$

与θ无关

- 因此 $\arg\max_\theta \mathcal{J}(G\mid D^*)=\arg\min_\theta KL(p_\theta,p_r)-2JS(p_r,p_\theta)$
- 其中JS散度$JS(p_r, p_\theta)\in[0, \log 2]$为有界函数，因此生成网络的目标更多的是受逆向KL散度$KL(p_\theta, p_r)$影响，使得生成网络更倾向于生成一些更“安全”的样本，从而造成模型坍塌(Model Collapse)问题.



- ✓ 前向KL散度会鼓励模型分布$p_\theta(\boldsymbol{x})$尽可能覆盖所有真实分布$p_r(\boldsymbol{x})>0$的点，而不用回避$p_r(\boldsymbol{x})\approx 0$的点.
- ✓ 逆向KL散度会鼓励模型分布$p_\theta(\boldsymbol{x})$尽可能避开所有真实分布$p_r(\boldsymbol{x})\approx 0$的点，而不需要考虑是否覆盖所有真实分布$p_r(\boldsymbol{x})>0$的点. 生成器没生成真实样本，惩罚微小；生成器生成不真实样本，惩罚巨大. 这导致生成器宁可多生成一些重复但是很“安全”的样本，也不愿意去生成多样性的样本.

3. Wasserstein GAN

➢ W-GAN

- 当两个分布没有重叠或者重叠非常少时，它们之间的KL散度为+∞，JS散度为log2，并不随着两个分布之间的距离而变化．而Wasserstein-1距离依然可以衡量两个没有重叠分布之间的距离.
- W-GAN是一种通过用Wasserstein距离替代JS散度来优化训练的生成对抗网络.

Wasserstein-1：

$$W_1[p_r, p_\theta] = \inf_{\gamma \in \Pi(p_r, p_\theta)} \mathbb{E}_{(x,y)\sim\gamma}\left[\|x - y\|\right]$$

其中$\Pi(p_r, p_\theta)$是边际分布为p_r与p_θ的所有可能的联合分布集合.

- 两个分布p_r与p_θ的Wasserstein-1距离通常难以直接计算，但是两个分布的Wasserstein-1距离有一个对偶形式：

$$W_1[p_r, p_\theta] = \sup_{\|f\|_L \le 1} \left(\mathbb{E}_{x\sim p_r}\left[f(x)\right] - \mathbb{E}_{x\sim p_\theta}\left[f(x)\right]\right)$$

Kantorovich-Rubinstein 对偶定理

其中$f: \mathbb{R}^d \to \mathbb{R}$为1-Lipschitz函数，满足 $\|f\|_L \triangleq \sup_{x\neq y} \frac{|f(x) - f(y)|}{|x - y|} \le 1.$

- 通常情况下，1-Lipschitz 连续的约束可以宽松为K-Lipschitz 连续．这样分布p_r与p_θ之间的Wasserstein-1距离为

$$W_1[p_r, p_\theta] = \frac{1}{K} \sup_{\|f\|_L \le K} \left(\mathbb{E}_{x\sim p_r}\left[f(x)\right] - \mathbb{E}_{x\sim p_\theta}\left[f(x)\right]\right)$$

➢ W-GAN

- 评价网络(Critic Network)

令$f(\boldsymbol{x};\phi)$为一个神经网络，假设存在参数集合Φ，对于所有的$\phi\in\Phi$，$f(\boldsymbol{x};\ \phi)$为K-Lipschitz连续函数，那么分布p_r与p_θ之间的Wasserstein-1距离公式的上界可以近似转换为

$$\max_{\phi\in\Phi}\left(\mathbb{E}_{x\sim p_r}\left[f(\boldsymbol{x};\phi)\right]-\mathbb{E}_{x\sim p_\theta}\left[f(\boldsymbol{x};\phi)\right]\right)\Leftrightarrow\max_{\phi\in\Phi}\left(\mathbb{E}_{x\sim p_r}\left[f(\boldsymbol{x};\phi)\right]-\mathbb{E}_{z\sim p(z)}\left[f\left(G(\boldsymbol{z};\theta);\phi\right)\right]\right)$$

其中$f(\boldsymbol{x};\ \phi)$称为评价网络.

和标准GAN中的判别网络的值域为[0, 1]不同，评价网络$f(\boldsymbol{x};\ \phi)$最后一层为线性层，其值域没有限制．这样只需要找到一个网络$f(\boldsymbol{x};\ \phi)$使其在两个分布p_r与p_θ下的期望的差最大．即对于真实样本，$f(\boldsymbol{x};\ \phi)$的打分要尽可能高；对于模型生成的样本，$f(\boldsymbol{x};\ \phi)$ 的打分要尽可能低.

- 生成网络(Generator Network)

生成网络的目标是使得评价网络$f(\boldsymbol{x};\ \phi)$对其生成样本的打分尽可能高，即

$$\max_{\theta}\ \mathbb{E}_{z\sim p(z)}\left[f\left(G(\boldsymbol{z};\theta);\phi\right)\right]\Leftrightarrow\min_{\theta}\ -\mathbb{E}_{z\sim p(z)}\left[f\left(G(\boldsymbol{z};\theta);\phi\right)\right]$$

由于$f(\boldsymbol{x};\ \phi)$为不饱和函数，因此生成网络参数θ的梯度不会消失，理论上解决了原始GAN训练不稳定的问题．并且W-GAN中生成网络的目标函数不再是两个分布的比率，在一定程度上缓解了模型坍塌问题，使得生成的样本具有多样性.

- 总体目标函数看做最小化最大化游戏(Minimax Game)

$$\min_{\theta}\max_{\phi}\left(\mathbb{E}_{x\sim p_r}\left[f(\boldsymbol{x};\phi)\right]-\mathbb{E}_{z\sim p(z)}\left[f\left(G(\boldsymbol{z};\theta);\phi\right)\right]\right)$$

➢ W-GAN训练过程

- 对于生成网络，在实际训练时一般使用下面公式，因为其梯度性质更好.

$$\max_{\theta} \mathbb{E}_{z\sim p(z)}\left[f\left(G(z;\theta);\phi\right)\right]$$

- 每次迭代时，判别网络更新K次而生成网络更新一次，即首先要保证判别网络足够强才能开始训练生成网络．在实践中K是个超参数，其取值一般取决于具体任务.
- 和原始GAN相比，W-GAN的评价网络最后一层不使用Sigmoid函数，损失函数不取对数.
- 为了使得$f(\boldsymbol{x};\ \phi)$满足K-Lipschitz连续，一种近似的方法是限制参数的取值范围．因为神经网络为连续可导函数，满足K-Lipschitz连续可以近似为其关于$\boldsymbol{x}$的偏导数的模 $\left\|\frac{\partial f(x;\phi)}{\partial x}\right\|$ 小于某个上界．由于这个偏导数的大小一般和参数的取值范围相关，我们可以通过限制参数ϕ的取值范围来近似，令$\phi\in[-c,\ c]$，c为一个比较小的正数，比如0.01.

输入: 训练集 $\mathcal{D}$，对抗训练迭代次数 T，每次评价网络的训练迭代次数 K，小批量样本数量 M，参数限制大小 c;

随机初始化 θ,ϕ;

for $t\leftarrow 1$ **to** T **do**

  // 训练评价网络 $f(\boldsymbol{x};\phi)$

  for $k\leftarrow 1$ **to** K **do**

    // 采集小批量训练样本

    从训练集 $\mathcal{D}$ 中采集 M 个样本 $\{\boldsymbol{x}^{(m)}\}, 1\le m\le M$;

    从分布 $\mathcal{N}(0,\boldsymbol{I})$ 中采集 M 个样本 $\{\boldsymbol{z}^{(m)}\}, 1\le m\le M$;

    // 计算评价网络参数 ϕ 的梯度

    $g_\phi=\frac{\partial}{\partial\phi}\left[\frac{1}{M}\sum_{m=1}^{M}\left(f(\boldsymbol{x}^{(m)};\phi)-f\left(G(\boldsymbol{z}^{(m)};\theta);\phi\right)\right)\right]$;

    $\phi\leftarrow\phi+\alpha\cdot\mathrm{RMSProp}(\phi,g_\phi)$; // 使用RMSProp算法更新 ϕ

    $\phi\leftarrow\mathrm{clip}(\phi,-c,c)$; // 梯度截断

  end

  // 训练生成网络 $G(\boldsymbol{z};\theta)$

  从分布 $\mathcal{N}(0,\boldsymbol{I})$ 中采集 M 个样本 $\{\boldsymbol{z}^{(m)}\}, 1\le m\le M$;

  // 更新生成网络参数 θ

  $g_\theta=\frac{\partial}{\partial\theta}\left[\frac{1}{M}\sum_{m=1}^{M}f\left(G(\boldsymbol{z}^{(m)};\theta);\phi\right)\right]$;

  $\theta\leftarrow\theta+\alpha\cdot\mathrm{RMSProp}(\theta,g_\theta)$; // 使用RMSProp算法更新 θ

end

输出: 生成网络 $G(\boldsymbol{z};\theta)$

拓展：由生成对抗网络联想到假设检验中的两类错误

生成器：支持原假设，$p_{data}=p_{\theta}$，生成的图像越真实越好；

判别器：支持备择假设，$p_{data} \neq p_{\theta}$，越能判别出假图像越好。

假设检验会出现两类错误，本来图像是真实的，原假设是正确的，但是却拒绝原假设，认为图像是假的，这是第一类错误；本来图像是假的，原假设是错误的，却接受原假设，认为图像是真实的，这是第二类错误。统计学告诉我们这两类错误都无法避免，也无法同时使两者出现的概率都最小，一类错误的减少必然会使另一类错误增加。一种折中的方案是，只限制犯第一类错误的概率，这就是Fisher显著性检验。对于GAN来说，生成器生成了一些重复但是很安全的样本，缺乏多样性。

4. 参考文献

[1] 邱锡鹏，神经网络与深度学习，机械工业出版社，https://nndl.github.io/, 2020.

[2] Lilian Weng, From GAN to WGAN, ariXiv, https://arxiv.org/pdf/1904.08994.pdf, 2019.