#学习周报

2019年11月7日周四

徐皓

adversarial training is the coolest thing since sliced bread

1.GAN的先导

1-1 KL 散度(divergence)

KL 散度是一种衡量两个分布（比如两条线）之间的匹配程度的方法。

KL散度和熵与交叉熵之间的关系

$$D_{KL}(p||q) = \sum_{i=1}^{n} p(x_i) log(\frac{p(x_i)}{q(x_i)}) = \sum_{i=1}^{n} p(x_i) log(p(x_i)) - \sum_{i=1}^{n} p(x_i) log(q(x_i)) = -H(p(x)) + [-\sum_{i=1}^{n} p(x_i) log(q(x_i))]$$

等式的前一部分恰巧就是p的熵，等式的后一部分，就是交叉熵：

在机器学习中，我们需要评估label和predicts之间的差距，使用KL散度刚刚好，由于KL散度中的前一部分−H(y)不变，故在优化过程中，只需要关注交叉熵就可以了。所以一般在机器学习中直接用用交叉熵做loss，评估模型。

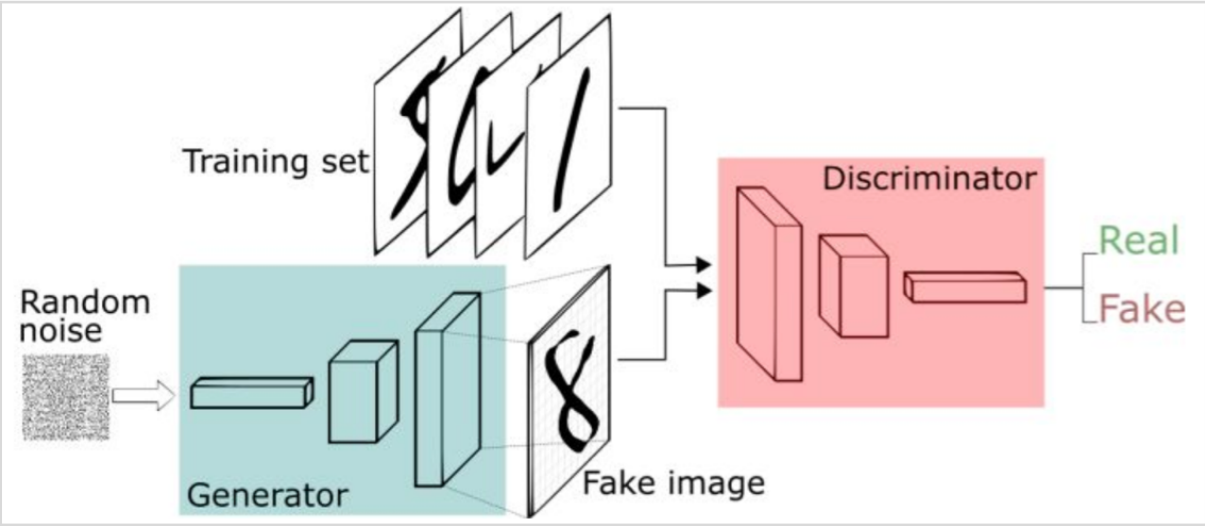
初学机器学习：直观解读KL散度的数学概念 - 知乎

1-2 JS 散度(divergence)

交叉熵、相对熵（KL散度）、JS散度和Wasserstein距离（推土机距离） - 知乎

2.GAN

基本结构

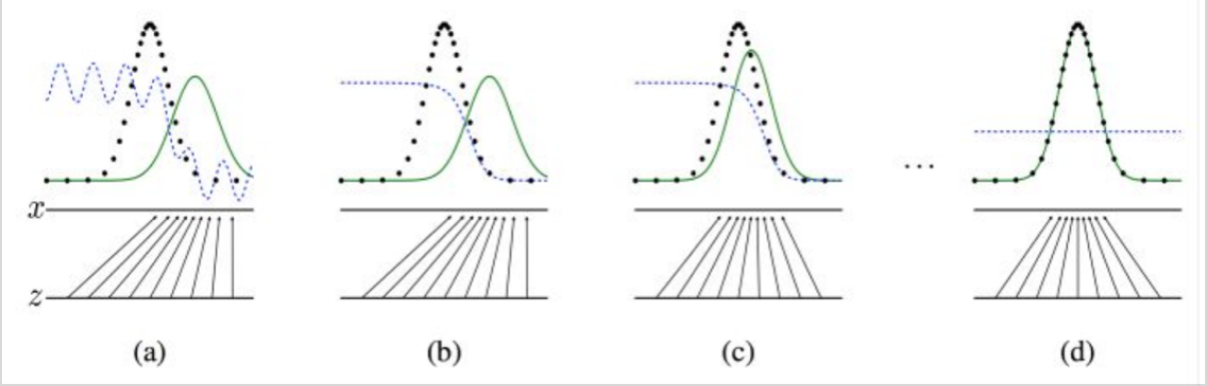


生成器（Generator）

判别器（Discrimnator）

- 初始化判别器D的参数 θ_d 和生成器G的参数 θ_g 。
- 从真实样本中采样 m 个样本{ $x^1, x^2, \ldots x^m$ }，从先验分布噪声中采样 m 个噪声样本{ $z^1, z^2, \ldots, z^m$ }并通过生成器获取 m 个生成样本{ $\tilde{x}^1, \tilde{x}^2, \ldots, \tilde{x}^m$ }。固定生成器G，训练判别器D尽可能好地准确判别真实样本和生成样本，尽可能大地区分正确样本和生成的样本。
- **循环k次更新判别器之后，使用较小的学习率来更新一次生成器的参数**，训练生成器使其尽可能能够减小生成样本与真实样本之间的差距，也相当于尽量使得判别器判别错误。
- 多次更新迭代之后，最终理想情况是使得判别器判别不出样本来自于生成器的输出还是真实的输出。亦即最终样本判别概率均为0.5。

即训练到最后即达到纳什均衡。



即让生成器逼近真实的样本分布，判别器分辨不出样本是生成的还是真实的（判别概率均为0.5）。

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

Where will D go (fixed G)

Proposition 1. *For G fixed, the optimal discriminator D is*

$$D^*_G(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})} \quad (2)$$

Proof. The training criterion for the discriminator D, given any generator G, is to maximize the quantity $V(G,D)$

$$V(G,D) = \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x}))dx + \int_{\boldsymbol{z}} p_{\boldsymbol{z}}(\boldsymbol{z}) \log(1 - D(g(\boldsymbol{z})))dz$$
$$= \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x})) + p_g(\boldsymbol{x}) \log(1 - D(\boldsymbol{x}))dx \quad (3)$$

$$f(\tilde{x}) = A log\tilde{x} + B log(1-\tilde{x})$$
$$\frac{df(\tilde{x})}{d\tilde{x}} = A\frac{1}{ln10}\frac{1}{\tilde{x}} - B\frac{1}{ln10}\frac{1}{1-\tilde{x}}$$
$$= \frac{1}{ln10}(\frac{A}{\tilde{x}} - \frac{B}{1-\tilde{x}})$$
$$= \frac{1}{ln10}\frac{A-(A+B)\tilde{x}}{\tilde{x}(1-\tilde{x})}$$

Thus, set $\frac{df(\tilde{x})}{d\tilde{x}} = 0$, we get the best value of the discriminator:

$$D^*(x) = \tilde{x}^* = \frac{A}{A+B} = \frac{p_r(x)}{p_r(x)+p_g(x)} \in [0,1]$$

Where will G go (after D*)

$$
\begin{aligned}
D_{JS}(p_r\|p_g) &= \frac{1}{2}D_{KL}(p_r||\frac{p_r+p_g}{2}) + \frac{1}{2}D_{KL}(p_g||\frac{p_r+p_g}{2}) \\
&= \frac{1}{2}\left(\log 2 + \int_x p_r(x)\log\frac{p_r(x)}{p_r+p_g(x)}dx\right) + \\
&\quad \frac{1}{2}\left(\log 2 + \int_x p_g(x)\log\frac{p_g(x)}{p_r+p_g(x)}dx\right) \\
&= \frac{1}{2}\left(\log 4 + L(G,D^*)\right)
\end{aligned}
$$

$$D_{JS}(p_r \parallel p_g) \geq 0$$

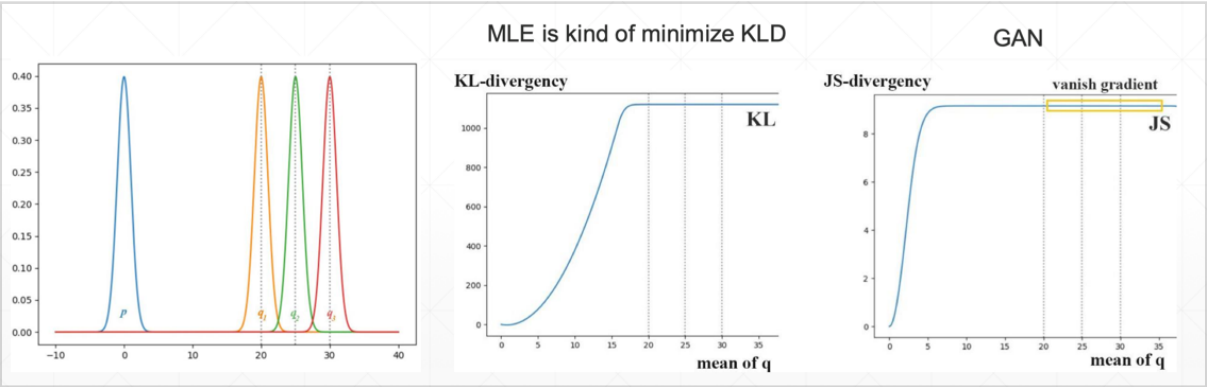
$$p_r = p_g$$

$$L(G,D^*) = 2D_{JS}(p_r\|p_g) - 2\log 2$$

这里因为JS散度一定是大于等于0的，而我们要最小化L(G,D*) 所以最后一定是Pr=Pg 即两个分布相等，训练到最后即达到纳什均衡。

但是对于GAN来说这KL，JS散度两种度量方式作用都不是很大，因为如果两个分配P,Q离得很远，完全没有重叠的时候，那么KL散度值是没有意义的，而JS散度值是一个常数。对于随机生成的噪声点来说，不可能使得P,Q相等。

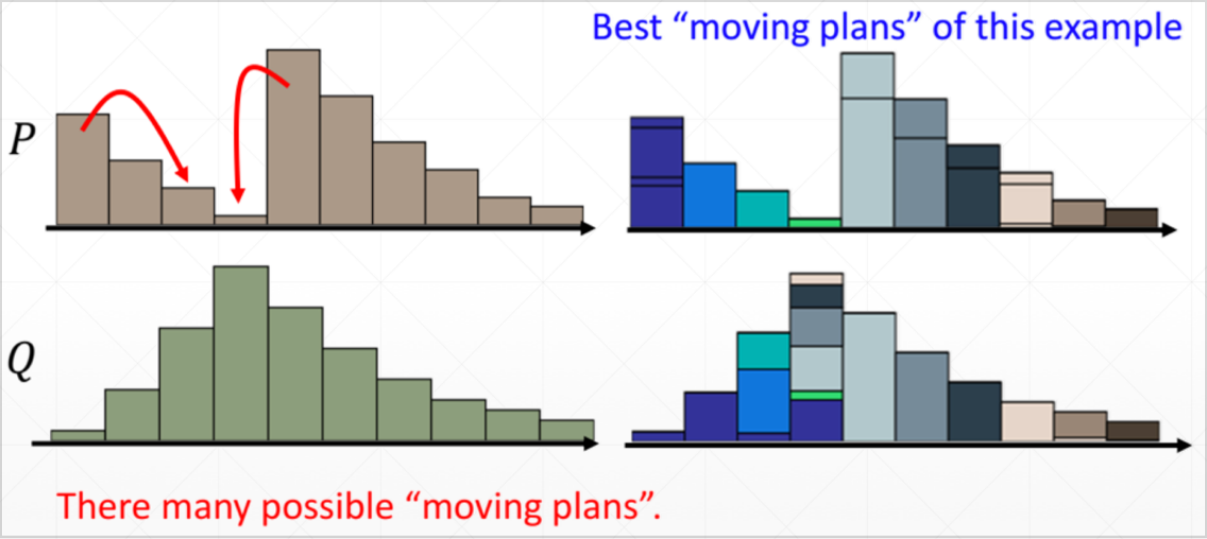
而且这会导致其出现梯度消失问题（gradient vanishing problem）



通俗理解生成对抗网络GAN - 知乎

3.WGAN

我们这里换一种评价方式 我们有我们模拟的分布P和真实分布Q



我们要做的就是如何最快速的“搬砖”让我们P更接近于Q的真实分布。

How to compute Wasserstein Distance

两种模型的区别主要在

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}\left[\, \|x - y\| \,\right],$$

$$\left[f\left(x^{(i)}\right) - f\left(G\left(z^{(i)}\right)\right)\right]$$

	Discriminator/Critic	Generator
GAN	$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(x^{(i)}\right) + \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right)\right]$	$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} -\log\left(D\left(G\left(z^{(i)}\right)\right)\right)$
WGAN	$\nabla_w \frac{1}{m} \sum_{i=1}^{m} \left[f\left(x^{(i)}\right) - f\left(G\left(z^{(i)}\right)\right)\right]$	$\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} -f\left(G\left(z^{(i)}\right)\right)$

$$|f(x_1) - f(x_2)| \leq |x_1 - x_2|.$$ 1-Lipschitz function

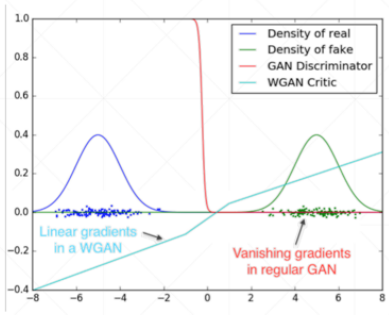
3-1 1-lipschitz functions

即必须满足这个条件才行|f(x1) – f(x2)| ≤ |x1 – x2|

转化之后就相当于梯度必须小于1

WGAN-Gradient Penalty

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



$$L = \underbrace{\mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}\left[D(\tilde{\boldsymbol{x}})\right] - \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}\left[D(\boldsymbol{x})\right]}_{\text{Original critic loss}} + \underbrace{\lambda \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}}\sim\mathbb{P}_{\hat{\boldsymbol{x}}}}\left[\left(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1\right)^2\right]}_{\text{Our gradient penalty}}.$$

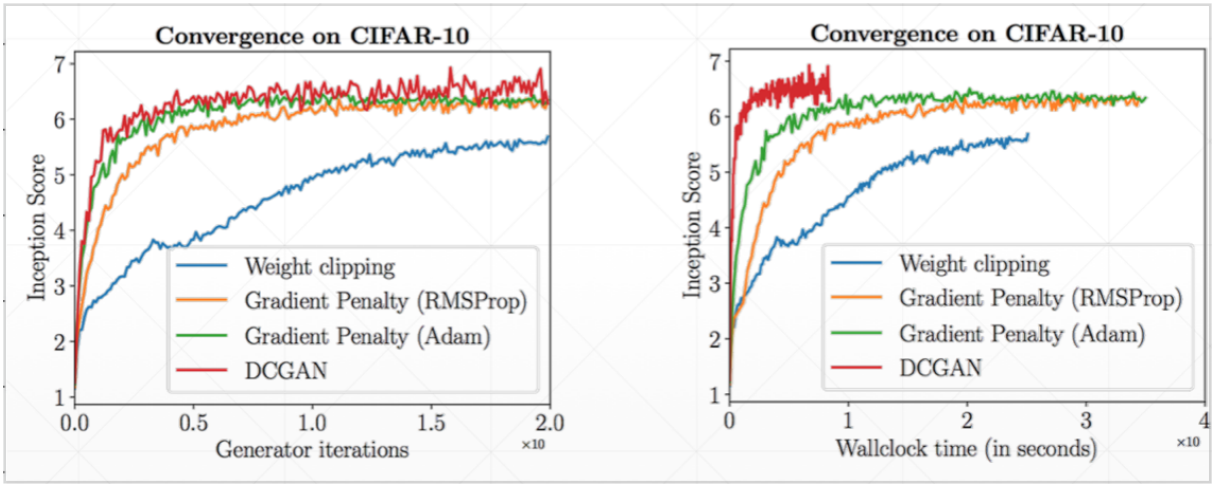
where $\hat{\boldsymbol{x}}$ *sampled from* $\tilde{\boldsymbol{x}}$ *and* $\boldsymbol{x}$ *with* t *uniformly sampled between 0 and 1*

$$\hat{\boldsymbol{x}} = t\tilde{\boldsymbol{x}} + (1-t)\boldsymbol{x} \text{ with } 0 \leq t \leq 1$$

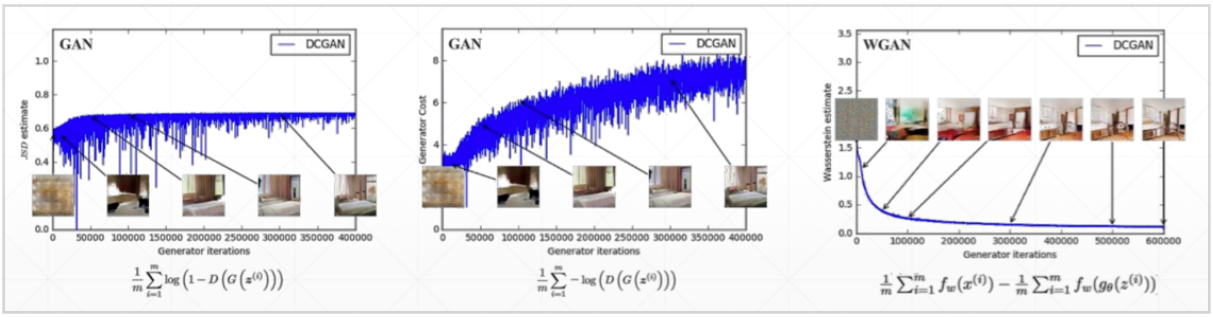
这里写成(D(x)-1)^2 只是工程上的一个写法

4.两种方法的优劣

虽然如果按下图中的数据，DCGAN的training时间比WGAN短且Score更高，但是这是在DCGAN最优的情况下。事实上，DCGAN更依赖于超参数的选择和网络结构的安排。而WGAN更加稳定简单，并且易于观察训练情况。



Training Progress Indicator



5.实例

Anime数据集

这个数据集计算量较大

CVPR18基于GANs的非配对学习用于图像增强 - 知乎

GAN最新进展：8大技巧提高稳定性 - 云+社区 - 腾讯云