对抗生成网络

ref.GAN ref.WGAN ref.郑华滨知乎博客 ref.WGAN-GP

GAN的目标函数是

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

分析目标函数，其中对判别器D，其目标函数是：$E_r[logD(x)] + E_g[log(1 - D(G(z))]$，对生
$E_g[log(1 - D(G(z))]$。

对判别器，一个样本的损失函数是

$$-P_r(x)\log D(x) - P_g(x)\log[1 - D(x)].....(2$$

首先最大化D(x)，对D(x)求导，得到

$$D^*(x) = \frac{P_r(x)}{P_r(x) + P_g(x)}.....(3)$$

其含义是，最优判别器$D^*(x)$可以准确给出输入样本中真实样本所占的比例。把(3)带入判别器的损

$$\mathbb{E}_{x\sim P_r}\log\frac{P_r(x)}{\frac{1}{2}[P_r(x) + P_g(x)]} + \mathbb{E}_{x\sim P_g}\log\frac{P_g(x)}{\frac{1}{2}[P_r(x) + P_g(x)]} -$$

即为2倍JS散度，

$$2JS(P_r\|P_g) - 2\log 2.....(4)(b)$$

原始GAN的损失函数是JS散度，对JS散度，当两个分布是不相交时，JS散度恒为为$log2$，如图1，
明)

The Total Variation (TV) distance

$$\delta(\mathbb{P}_r, \mathbb{P}_g) = \sup_{A\in\Sigma}|\mathbb{P}_r(A) - \mathbb{P}_g(A)|$$

The Kullback-Leibler (KL) divergence

$$KL(\mathbb{P}_r\|\mathbb{P}_g) = \int\log\left(\frac{P_r(x)}{P_g(x)}\right)P_r(x)d\mu(x)$$

The Jensen-Shannon (JS) divergence

$$JS(\mathbb{P}_r, \mathbb{P}_g) = KL(\mathbb{P}_r\|\mathbb{P}_m) + KL(\mathbb{P}_g\|\mathbb{P}_m)$$

其中$\mathbb{P}_m = (\mathbb{P}_r + \mathbb{P}_g)/2$

The Earth-Mover (EM) distance or Wasserstein-1

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

对图1的分布，三种不同距离分别为，后两种距离不连续

$\cdot W(\mathbb{P}_0, \mathbb{P}_\theta) = |\theta|$

$$\cdot JS\left(\mathbb{P}_0, \mathbb{P}_\theta\right) = \begin{cases} \log 2 & \text{if } \theta \neq 0 \\ 0 & \text{if } \theta = 0 \end{cases}$$

$$\cdot KL\left(\mathbb{P}_\theta \| \mathbb{P}_0\right) = KL\left(\mathbb{P}_0 \| \mathbb{P}_\theta\right) = \begin{cases} +\infty & \text{if } \theta \neq 0 \\ 0 & \text{if } \theta = 0 \end{cases}$$

由于推土机距离是计算下确界inf，很难计算，根据Kantorovich-Rubinstein duality(对偶)，具体计

$$W\left(\mathbb{P}_r, \mathbb{P}_\theta\right) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] - \mathbb{E}_{x\sim\mathbb{P}_\theta}[f(x)$$

距离计算变为求两个分布的期望差的上确界sup。个人理解，相较与下确界inf，sup的好处是，我来进行优化。这是因为目标是最小化其上确界，一般情况下的期望差必然更小

WGAN算法:

Algorithm 1 WGAN, our proposed algorithm. All experiments in th
the default values $\alpha = 0.00005$, $c = 0.01$, $m = 64$, $n_{\text{critic}} = 5$.

Require: : α, the learning rate. c, the clipping parameter. m, th
n_{critic}, the number of iterations of the critic per generator iteratio
Require: : w_0, initial critic parameters. θ_0, initial generator's param
while θ has not converged **do**
  for $t = 0, ..., n_{\text{critic}}$ **do**
    Sample $\{x^{(i)}\}_{i=1}^m \sim \mathbb{P}_r$ a batch from the real data.
    Sample $\{z^{(i)}\}_{i=1}^m \sim p(z)$ a batch of prior samples.
    $g_w \leftarrow \nabla_w \left[\frac{1}{m}\sum_{i=1}^m f_w(x^{(i)}) - \frac{1}{m}\sum_{i=1}^m f_w(g_\theta(z^{(i)}))\right]$
    $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$
    $w \leftarrow \text{clip}(w, -c, c)$
  end for
  Sample $\{z^{(i)}\}_{i=1}^m \sim p(z)$ a batch of prior samples.
  $g_\theta \leftarrow -\nabla_\theta \frac{1}{m}\sum_{i=1}^m f_w(g_\theta(z^{(i)}))$
  $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$
end while

WGAN算法有一个混合真实样本和生成样本的步骤，判别器损失和原始GAN相同，生成器损失变为

对WGAN-GP, WGAN对权值裁剪，WGAN-GP对梯度进行裁剪。原文中提到了一些梯度裁剪的好处如

```
Gradient norms of deep WGAN critics during training on
toy datasets either explode or vanish when using weight
clipping, but not when using a gradient penalty. (right)
Weight clipping (top) pushes weights towards two values
(the extremes of the clipping range), unlike gradient
penalty (bottom).
```

WGAN-GP的算法如下：

Algorithm 1 WGAN with gradient penalty. We use default values of $\lambda = 1($ 0.0001, $\beta_1 = 0$, $\beta_2 = 0.9$.

Require: The gradient penalty coefficient λ, the number of critic iterations per n_{critic}, the batch size m, Adam hyperparameters α, β_1, β_2.
Require: initial critic parameters w_0, initial generator parameters θ_0.

while θ has not converged **do**
  for $t = 1, ..., n_{\text{critic}}$ **do**
    for $i = 1, ..., m$ **do**
      Sample real data $\boldsymbol{x} \sim \mathbb{P}_r$, latent variable $\boldsymbol{z} \sim p(\boldsymbol{z})$, a random nu
      $\tilde{\boldsymbol{x}} \leftarrow G_\theta(\boldsymbol{z})$
      $\hat{\boldsymbol{x}} \leftarrow \epsilon \boldsymbol{x} + (1 - \epsilon)\tilde{\boldsymbol{x}}$
      $L^{(i)} \leftarrow D_w(\tilde{\boldsymbol{x}}) - D_w(\boldsymbol{x}) + \lambda(\|\nabla_{\hat{\boldsymbol{x}}} D_w(\hat{\boldsymbol{x}})\|_2 - 1)^2$
    end for
    $w \leftarrow \text{Adam}(\nabla_w \frac{1}{m} \sum_{i=1}^{m} L^{(i)}, w, \alpha, \beta_1, \beta_2)$
  end for
  Sample a batch of latent variables $\{\boldsymbol{z}^{(i)}\}_{i=1}^{m} \sim p(\boldsymbol{z})$.
  $\theta \leftarrow \text{Adam}(\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} -D_w(G_\theta(\boldsymbol{z})), \theta, \alpha, \beta_1, \beta_2)$
end while

```
import  numpy  as  np
import  matplotlib.pyplot  as  plt
from  scipy  import  stats
mu  =  0
sd  =  0.3
t1  =  np.linspace(-1,1,1000)
t2  =  np.linspace(2,4,1000)
y1  =  stats.norm(mu,  sd).pdf(t1)
y2  =  stats.norm(mu,  sd).pdf(t1)

plt.figure(1)
plt.plot(t1,y1)
plt.plot(t2,y2)
plt.title("Figure  1")

plt.figure(2)
plt.plot(t1,y1)
plt.plot(t1,y1+0.5)
plt.title("Figure  2")
```

```
Text(0.5, 1.0, 'Figure 2')
```

Figure 1

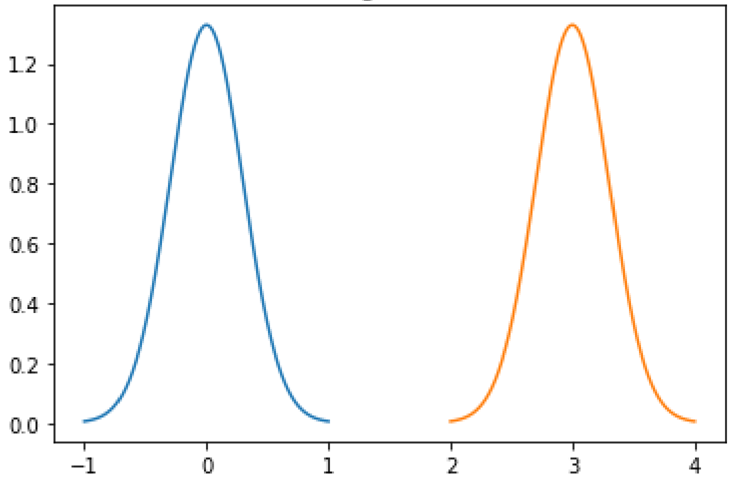

Figure 2