今天的主题是LSGAN——最小二乘GAN[1]，请注意，不是loss sensitive GAN。有两种LSGAN，least square GAN 和 loss sensitive GAN，两者有很大的差别。本期的主题是前者，后者我们作为下期的主题。

GAN回顾

什么是GAN？GAN就是警察抓小偷，是个博弈，警察总是想着如何分辨小偷和非小偷，而小偷总是想着尽可能地伪装成正常人不被发现。这个博弈达到纳什均衡时，小偷的表现就跟正常人一样，而警察也无法判断“小偷”是不是小偷......这个时候的“小偷”就不能算小偷了？

我们现在把警察看成D，分辨网络（discriminator），小偷看成G，生成网络（generator）。警察认为小偷会有一些特征，满足一个“小偷”分布 p_g，而正常人满足一个“正常人”分布 p_d。从这个角度来看，小偷G的目标就是让他的分布尽可能地接近“正常人”分布。对于一个小偷来说，他服从“小偷”分布p_g，他会有一个独特的特征z，服从一个小偷特有的特征分布 p_z，这是跟别人不一样的地方。写成公式就是

\min_G L(G)=\mathbb{E}_{z \sim p_z} \log(1 - D(G(z))) <---- 尽可能不让警察D发现

而警察的任务是分辨常人和小偷，常人输出1，小偷输出0。写成公式就是

\max_D L(D) = \mathbb{E}_{x \sim p_d} \log D(x) + \mathbb{E}_{z \sim p_z} \log (1-D(G(z))) <---- 尽可能把常人判为1，小偷判为0

Ian Goodfellow证明了，GAN存在纳什均衡解。实践中，我们很难去找一个博弈的纳什均衡，一般转而采用梯度算法优化目标函数L(G), L(D)。

LSGAN

最小二乘GAN，正如它的名字所指示的，目标函数将是一个平方误差，考虑到D网络的目标是分辨两类，如果给生成样本和真实样本分别编码为a,b，那么采用平方误差作为目标函数，D的目标就是

\max_D L(D) = \mathbb{E}_{x \sim p_x} (D(x)-b)^2 + \mathbb{E}_{z \sim p_z} (D(G(z))-a)^2

G的目标函数将编码a换成编码c，这个编码表示D将G生成的样本当成真实样本，

\min_G L(G) = \mathbb{E}_{z \sim p_z} (D(G(z))-c)^2

在下一节我们会证明，当b-c=1, b-a=2时，目标函数等价于皮尔森卡方散度（Pearson \chi^2 divergence）。一般来说，取a=-1, b=1, c=0或者a=-1, b=c=1。作者说，这两种设置在实验中效果没有显著差别（**实际上，我用DCGAN代码修改目标函数为平方误差，然后发现在MNIST上，前者效果还可以，但是后者就只能产生噪声了，两者的差别只有a,b,c的取值！或许这还跟网络架构有关，我用的不是LSGAN论文中的网络架构**）。

LSGAN收敛性

LSGAN收敛性可以套用原始GAN的证明框架：

固定G以后，我们能够求出最优的D，令D的目标函数的导数为0，不难求得

D^\ast(x)=\frac{bp_d(x)+ap_g(x)}{p_d(x)+p_g(x)}

将这个结果代入到L(G)中，对L(G)我们人为地添加一个与G无关的常数项 \mathbb{E}_{x \sim p_x} (D(x)-c)^2，化简以后就得到了

L(G)=\int_{\mathcal{X}} \frac{((b-c)(p_d(x)+p_g(x))-(b-a)p_g(x))^2}{p_d(x)+p_g(x)}dx

当b-c=1, b-a=2时，

L(G)=0.5\chi^2_{\text{Pearson}}(p_d+p_g||2p_g)

也就是说，此时优化LSGAN等价于优化皮尔森卡方散度。

类似地，我们是否能够构造出其他散度对应的目标函数呢？KL散度和皮尔森卡方散度都属于 f 散度， 常见的 f 散度有

KL divergence: f(t)=t\log t

\chi^2 divergence: f(t)=(t-1)^2, t^2-1

reversed KL divergence: f(t)=-\log t

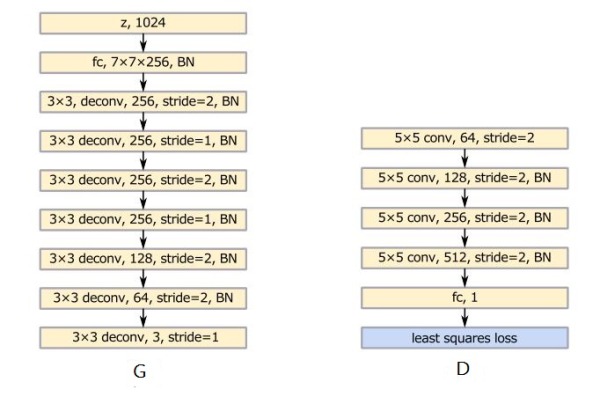
Hellinger distance: f(t)=(\sqrt t-1)^2, 2(1-\sqrt t)

Total variation distance: f(t)=0.5|t-1|

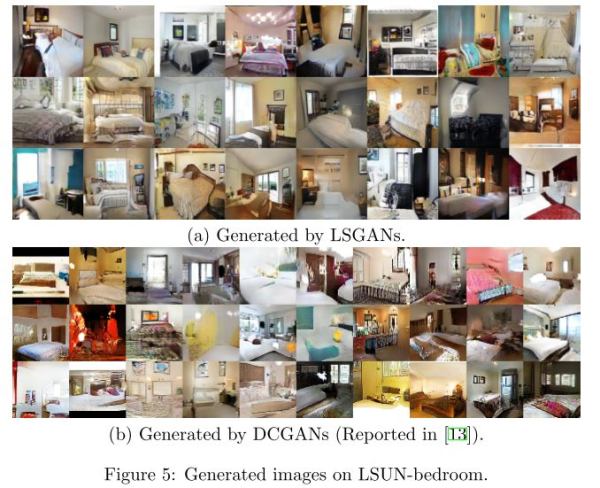
事实上，其他散度对应的目标函数不好构造。大家可以尝试一下。我没构造出来...

实验

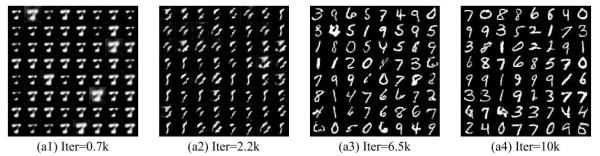
LSGAN的论文[1]做了图像生成的实验，在MNIST、LSUN和HWDB1.0（手写汉字）数据集上进行。作者提出了两类架构，第一种处理类别少的情况，例如MNIST、LSUN。网络设计如下：

跟DCGAN相比，多了一些stride=1的卷积。

在LSUN bedroom数据集上产生实验，产生的图像效果跟DCGAN没有什么差别（架构跟DCGAN相同，只是改了目标函数），如下图所示，也就是说，采用平方误差作为目标函数是有效的。

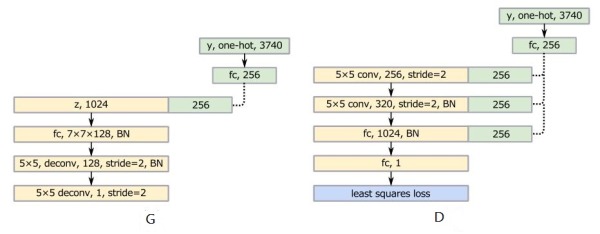


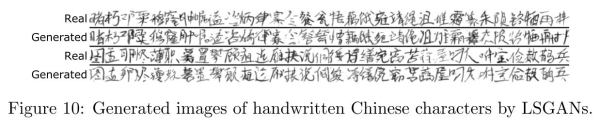
在MNIST数据集上实验，产生的图像质量也还不错。



**实际上，你可以直接把DCGAN的代码修改一下目标函数，就成了LSGAN**。但是训练的效果可能没有作者提出的架构好，作者应该对网络架构也做了一些探索。

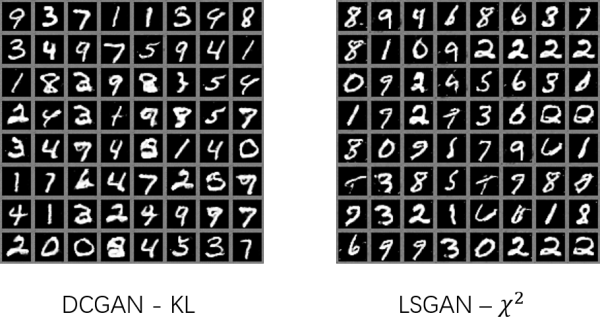
第二类处理类别特别多的情形，实际上是个条件版本的LSGAN。针对手写汉字数据集，有3740类，提出的网络结构如下：

类别多的情形效果也还不错，以下是产生的样本图像：



评价

Least Square GAN相较于GAN，主要是换了个目标函数，从论文的描述来看，效果比GAN要好，而我用MNIST数据集做的实验发现，相同架构，只是换了目标函数，产生的图像质量没有太大差别，如下图。



然而，从WGAN的证明来看，尽管LSGAN优化的目标不是KL散度了，而是皮尔森卡方散度，它们并没有本质上的变化，用divergence衡量两个分布的相似程度，避不开零测集的问题，训练仍然会震荡。

代码

1. tensorflow/pytorch: [wiseodd/generative-models](https://github.com/wiseodd/generative-models)

2. chainer: [musyoku/LSGAN](https://github.com/musyoku/LSGAN)

github上相关的代码非常多，这里就不一一列举了。拿DCGAN改一下目标函数也行。

参考文献

1. Mao, X., Li, Q., Xie, H., Lau, R. Y. K., & Wang, Z. (2016). Least Squares Generative Adversarial Networks, 1–15. [[1611.04076] Least Squares Generative Adversarial Networks](http://arxiv.org/abs/1611.04076)