


[登录](#) [忘记密码](#) [免费注册](#)

全部

输入关键词

快捷导航


[门户](#) [数据·中国](#) [专才计划](#) [特训营](#) [培训](#) [课程](#) [专业](#) [企业服务](#) [论坛](#) [奖学金](#) [大数据](#) [商业智能](#) [量化投资](#) [创业](#) [专家团](#) [关于我们](#)
[门户](#) [商业智能](#) [深度学习](#) [查看内容](#)

令人拍案叫绝的Wasserstein GAN

2017-2-26 22:10 | 发布者: 炼数成金_小数 | 查看: 15118 | 评论: 0 | 原作者: 郑华滨 | 来自: 知乎

摘要:要知道自从2014年Ian Goodfellow提出以来，GAN就存在着训练困难、生成器和判别器的loss无法指示训练进程、生成样本缺乏多样性等问题。从那时起，很多论文都在尝试解决，但是效果不尽人意，比如最有名的一个改进DCGAN ...

网络 工具 算法 架构 数学

在GAN的相关研究如火如荼甚至可以说是泛滥的今天，一篇新鲜出炉的arXiv论文《Wasserstein GAN》却在Reddit的Machine Learning频道火了，连Goodfellow都在帖子里和大家热烈讨论，这篇论文究竟有什么了不得的地方呢？

- [ROS机器人操作系统实战（第...](#)
- [Qt编程快速入门（第四期）](#)
- [Python机器学习Kaggle案例实...](#)
- [Mycat从入门到精通（第11期...](#)
- [深入浅出Spring（第四期）](#)
- [黄美灵的Spark ML机器学习实...](#)
- [R语言魔鬼训练营（第八期）](#)
- [Python突击—从入门到精通到...](#)

Goodfellow提出以来，GAN就存在着训练困难、生成器和判别器、生成样本缺乏多样性等问题。从那时起，很多论文都在尝试，比如最有名的一个改进DCGAN依靠的是对判别器和生成器最终找到一组比较好的网络架构设置，但是实际上是治标不治本。而今天的主角Wasserstein GAN（下面简称WGAN）成功地解决了这些问题，不再需要小心平衡生成器和判别器的训练程度和参数的问题，确保了生成样本的多样性。交叉熵、准确率这样的数值来指示训练的进程，这个数值越大代表生成器产生的图像质量越高（如题图所示）。

以上一切好处不需要精心设计的网络架构，最简单的多层全连接网络就可以做到。那以上好处来自哪里？这就是令人拍案叫绝的部分了——实际上作者整整花了两篇论文，在第一篇《Towards Principled Methods for Training Generative Adversarial Networks》里面推了一堆公式定理，从理论上分析了原始GAN的问题所在，从而针对性地给出了改进要点；在这第二篇《Wasserstein GAN》里面，又再从这个改进点出发推了一堆公式定理，最终给出了改进的算法实现流程，而改进后相比原始GAN的算法实现流程却只改了四点：

判别器最后一层去掉sigmoid

生成器和判别器的loss不取log

每次更新判别器的参数之后把它们的值截断到不超过一个固定常数c

不要用基于动量的优化算法（包括momentum和Adam），推荐RMSProp，SGD也行
算法截图如下：

热门频道

[大数据](#) [商业智能](#)

[量化投资](#) [科学探索](#) [创业](#)



炼数成金个性化教育产品“专才计划”上线，导师制私人订制尊贵服务

即将开课

- 黄美灵的Spark ML机器学习实...
- 【免费公开课】Qt编程快速入...
- Python机器学习（第11期）
- ROS机器人操作系统实战（第...)
- 深入浅出Spring（第四期）
- R语言魔鬼训练营（第八期）
- zabbix企业级监控（第11期）
- MySQL性能优化（第11期）
- OpenCV计算机视觉（第11期）
- 【免费公开课】Python自动化运维工具（第1期）
- 【免费公开课】赢在大数据-人工智能的应用
- 数据库系统实现技术内幕（第11期）
- Oracle 12c特性解读-容器数据库和灾备（第11期）
- 机器学习（第20期）
- 大数据必知的java基础（第七期）
- Python网络爬虫（第八期）
- 软件架构必备基础（第一期）
- Python机器学习（第四期）
- 企业级Hadoop大数据平台实践（第12期）
- 深度学习框架Tensorflow学习与应用（第四期）
- 深入JVM内核—原理、诊断与优化（第13期）
- Excel数据分析师突击—从入门到精通到项目
- Spark大数据平台应用实战（第二期）
- 金融市场基础（第四期）
- NoSQL与NewSQL数据库引航（第17期）
- 数据分析与SAS（第17期）
- 深入浅出Oracle（第六期）
- 【免费公开课】Julia快速数据分析（第一期）
- 区块链技术从入门到精通（第三期）
- 基于案例学习时间序列分析（第三期）
- JAVA极客特训（第三期）
- Python数据分析（第十期）
- 人工智能前沿系列之生成式对抗网络（第四期）

Algorithm 1 WGAN, our proposed algorithm. All experiments in the paper used 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 , the batch size. n_{critic} , the number of iterations of the critic per generator iteration.

Require: : w_0 , initial critic parameters. θ_0 , initial generator's parameters.

```

1: while  $\theta$  has not converged do
2:   for  $t = 0, \dots, n_{\text{critic}}$  do
3:     Sample  $\{x^{(i)}\}_{i=1}^m \sim \mathbb{P}_r$  a batch from the real data.
4:     Sample  $\{z^{(i)}\}_{i=1}^m \sim p(z)$  a batch of prior samples.
5:      $g_w \leftarrow \nabla_w [\frac{1}{m} \sum_{i=1}^m f_w(x^{(i)}) - \frac{1}{m} \sum_{i=1}^m f_w(g_\theta(z^{(i)}))]$ 
6:      $w \leftarrow w + \alpha \cdot \text{RMSPProp}(w, g_w)$ 
7:      $w \leftarrow \text{clip}(w, -c, c)$ 
8:   end for
9:   Sample  $\{z^{(i)}\}_{i=1}^m \sim p(z)$  a batch of prior samples.
10:   $g_\theta \leftarrow -\nabla_\theta \frac{1}{m} \sum_{i=1}^m f_w(g_\theta(z^{(i)}))$ 
11:   $\theta \leftarrow \theta - \alpha \cdot \text{RMSPProp}(\theta, g_\theta)$ 
12: end while

```

改动是如此简单，效果却惊人地好，以至于Reddit上不少人在感叹：就这样？没有别的了？太简单了吧！这些反应让我想起了一个颇有年头的鸡汤段子，说是一个工程师在电机外壳上用粉笔划了一条线排除了故障，要价一万美元——画一条线，1美元；知道在哪画线，9999美元。上面这四点改进就是作者Martin Arjovsky划的简简单单四条线，对于工程实现便已足够，但是知道在哪划线，背后却是精巧的数学分析，而这也是本文想要整理的内容。

本文内容分为五个部分：

ROS机器人操作系统实战（第三期）
Qt编程快速入门（第四期）
Python机器学习Kaggle案例实... Mycat从入门到精通（第11期... 深入浅出Spring（第四期） 黄美灵的Spark ML机器学习实... R语言魔鬼训练营（第八期） Python突击—从入门到精通到...

问题？（此部分较长）

方案

质

N

需要对测度论、拓扑学等数学知识有所掌握，本文会从直观
进行解读，有时通过一些低维的例子帮助读者理解数学背景。
谨，如有引喻不当之处，欢迎在评论中指出。

IN》为“WGAN本作”，简称《Towards Principled

Methods for Training Generative Adversarial Networks》为“WGAN前作”。

WGAN源码实现：[martinarjovsky/WassersteinGAN](https://github.com/martinarjovsky/WassersteinGAN)

第一部分：原始GAN究竟出了什么问题？

回顾一下，原始GAN中判别器要最小化如下损失函数，尽可能把真实样本分为正例，生成样本分为负例：

$$-\mathbb{E}_{x \sim P_r} [\log D(x)] - \mathbb{E}_{x \sim P_g} [\log(1 - D(x))] \quad (\text{公式1})$$

其中 P_r 是真实样本分布， P_g 是由生成器产生的样本分布。对于生成器，Goodfellow一开始提出来一个损失函数，后来又提出了一个改进的损失函数，分别是

$$\mathbb{E}_{x \sim P_g} [\log(1 - D(x))] \quad (\text{公式2})$$

$$\mathbb{E}_{x \sim P_g} [-\log D(x)] \quad (\text{公式3})$$

后者在WGAN两篇论文中称为“the - log D alternative”或“the - log D trick”。WGAN前作分别分析了这两种形式的原始GAN各自的问题所在，下面分别说明。

第一种原始GAN形式的问题

一句话概括：判别器越好，生成器梯度消失越严重。WGAN前作从两个角度进行了论证，第一个角度是从生成器的等价损失函数切入的。

首先从公式1可以得到，在生成器G固定参数时最优的判别器D应该是什么。对于一个具

- 深度学习框架Keras学习与应用（第一期）
- 【免费公开课】R七种武器之网络爬虫
- 基于R的Kaggle实战案例详解（第三期）
- JavaScript从入门到精通（第六期）
- 机器读心术之神经网络与深度学习（第八期）
- 深入浅出Git（第三期）
- Hive数据仓库实践（第五期）
- 金融的人工智能革命（第一期）
- 大数据算法导论（第13期）
- 开源计算机视觉库OpenCV从入门到应用
- R语言数据分析、展现与实例（第29期）
- Java魔鬼训练营（第四期）
- Hadoop集群原理与运维实践（第二期）
- 从零构建HA实时计算系统及在推荐/搜索应用
- 【免费公开课】数据库设计（第24期）
- 机器读心术之文本挖掘与自然语言处理（第8期）
- Redis技术实战（第八期）
- 实战Java高并发程序设计（第13期）
- python魔鬼训练营（第九期）
- 大数据的统计学基础（第20期）
- Hadoop应用开发实战案例（第18期）
- 大数据的矩阵计算基础（第13期）
- MySQL DBA从小白到大神实战（第六期）
- Hadoop数据分析平台（第45期）
- 面试突击-数据结构与算法速成（第四期）



热门文章

- 陈天奇团队发布TensorFlow深度学习框架，江湖自此无TensorFlow
- 深度学习不是AI的未来
- 最全知识图谱综述#2: 构建技术与典型应用
- 最全知识图谱综述#1: 概念以及构建技术
- 揭秘支付宝中的深度学习引擎: xNN
- MATLAB更新R2017b: 转换CUDA代码极大
- 功成身退: Yoshua Bengio宣布即将终止
- 文森特系统用深度学习将涂鸦变成艺术创作
- 被Geoffrey Hinton抛弃，反向传播为何饱受
- 如何优雅地用TensorFlow预测时间序



体的样本，它可能来自真实分布也可能来自生成分布，它对公式1损失函数的贡献是

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

令其关于 $D(x)$ 的导数为0，得

$$-\frac{P_r(x)}{D(x)} + \frac{P_g(x)}{1 - D(x)} = 0$$

化简得最优判别器为：

$$D^*(x) = \frac{P_r(x)}{P_r(x) + P_g(x)} \quad (\text{公式4})$$

这个结果从直观上很容易理解，就是看一个样本 x 来自真实分布和生成分布的可能性的相对比例。如果 $P_r(x) = 0$ 且 $P_g(x) \neq 0$ ，最优判别器就应该非常自信地给出概率0；如果 $P_r(x) = P_g(x)$ ，说明该样本是真是假的可能性刚好一半一半，此时最优判别器也应该给出概率0.5。

然而GAN训练有一个trick，就是别把判别器训练得太好，否则在实验中生成器会完全学不动（loss降不下去），为了探究背后的原因，我们就可以看看在极端情况——判别器最优时，生成器的损失函数变成什么。给公式2加上一个不依赖于生成器的项，使之变成

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

注意，最小化这个损失函数等价于最小化公式2，而且它刚好是判别器损失函数的反。代入最优判别器即公式4，再进行简单的变换可以得到

$$-\left[\mathbb{E}_{x \sim P_r}[\log \frac{P_r(x)}{\mathbb{E}_{x \sim P_g}[\frac{P_g(x)}{\frac{1}{2}[P_r(x) + P_g(x)}]]}] + 2\log 2 \right] \quad (\text{公式5})$$

入Kullback–Leibler divergence（简称KL散度）和Jensen-Shannon这两个重要的相似度衡量指标，后面的主角之一Wasserstein距离。所以接下来介绍这两个重要的配角——KL散度和JS散度：

$$\frac{P_1}{P_2} \quad (\text{公式6})$$

$$\frac{P_1 + P_2}{2} + \frac{1}{2} KL(P_2 || \frac{P_1 + P_2}{2}) \quad (\text{公式7})$$

\hat{g}

$$\mathbb{E}_{x \sim P_g}[\hat{g}(x)] \quad (\text{公式8})$$

到这里读者可以先喘一口气，看看目前得到了什么结论：根据原始GAN定义的判别器loss，我们可以得到最优判别器的形式；而在最优判别器的下，我们可以把原始GAN定义的生成器loss等价变换为最小化真实分布 P_r 与生成分布 P_g 之间的JS散度。我们越训练判别器，它就越接近最优，最小化生成器的loss也就会越近似于最小化 P_r 和 P_g 之间的JS散度。

问题就出在这个JS散度上。我们会希望如果两个分布之间越接近它们的JS散度越小，我们通过优化JS散度就能将 P_g “拉向” P_r ，最终以假乱真。这个希望在两个分布有所重叠的时候是成立的，但是如果两个分布完全没有重叠的部分，或者它们重叠的部分可忽略（下面解释什么叫可忽略），它们的JS散度是多少呢？

» 最新活动

» 联系客服

答案是 $\log 2$ ，因为对于任意一个 x 只有四种可能：

$$P_1(x) = 0 \text{ 且 } P_2(x) = 0$$

$$P_1(x) \neq 0 \text{ 且 } P_2(x) \neq 0$$

$$P_1(x) = 0 \text{ 且 } P_2(x) \neq 0$$

$$P_1(x) \neq 0 \text{ 且 } P_2(x) = 0$$

第一种对计算JS散度无贡献，第二种情况由于重叠部分可忽略所以贡献也为0，第三种情况对公式7右边第一个项的贡献是 $\log \frac{P_2}{\frac{1}{2}(P_2 + 0)} = \log 2$ ，第四种情况与之类似，所以最终 $JS(P_1||P_2) = \log 2$ 。

换句话说，无论 P_r 跟 P_g 是远在天边，还是近在眼前，只要它们俩没有一点重叠或者重叠部分可忽略，JS散度就固定是常数 $\log 2$ ，而这对于梯度下降方法意味着——梯度为0！此时对于最优判别器来说，生成器肯定是得不到一丁点梯度信息的；即使对于接近最优的判别器来说，生成器也有很大机会面临梯度消失的问题。

但是 P_r 与 P_g 不重叠或重叠部分可忽略的可能性有多大？不严谨的答案是：非常大。比较严谨的答案是：当 P_r 与 P_g 的支撑集（support）是高维空间中的低维流形（manifold）时， P_r 与 P_g 重叠部分测度（measure）为0的概率为1。

不用被奇怪的术语吓得关掉页面，虽然论文给出的是严格的数学表述，但是直观上其实很容易理解。首先简单介绍一下这几个概念：

- 支撑集（support）其实就是函数的非零部分子集，比如ReLU函数的支撑集就是 $(0, +\infty)$ ，就是所有概率密度非零部分的集合。

空间中曲线、曲面概念的拓广，我们可以在低维上直观理解。空间中的一个曲面是一个二维流形，因为它的本质维度只有2，一个点在这个二维流形上移动只有两个方向的自由。二维空间中的一条曲线都是一个一维流形。

空间中长度、面积、体积概念的拓广，可以理解为“超体”。 P_r 与 P_g 的支撑集是高维空间中的低维流形时，基本上是成立的。是从某个低维（比如100维）的随机分布中采样出一个编码向量，一个高维样本（比如64x64的图片就有4096维）。当生成器的参数

固定时，生成样本的概率分布虽然是定义在4096维的空间上，但它本身所有可能产生的变化已经被那个100维的随机分布限定了，其本质维度就是100，再考虑到神经网络带来的映射降维，最终可能比100还小，所以生成样本分布的支撑集就在4096维空间中构成一个最多100维的低维流形，“撑不满”整个高维空间。

“撑不满”就会导致真实分布与生成分布难以“碰到面”，这很容易在二维空间中理解：一方面，二维平面中随机取两条曲线，它们之间刚好存在重叠线段的概率为0；另一方面，虽然它们很可能可能存在交叉点，但是相比于两条曲线而言，交叉点比曲线低一个维度，长度（测度）为0，可忽略。三维空间中也是类似的，随机取两个曲面，它们之间最多就是比较有可能存在交叉线，但是交叉线比曲面低一个维度，面积（测度）是0，可忽略。从低维空间拓展到高维空间，就有了如下逻辑：因为一开始生成器随机初始化，所以 P_g 几乎不可能与 P_r 有什么关联，所以它们的支撑集之间的重叠部分要么不存在，要么就比 P_r 和 P_g 的最小维度还要低至少一个维度，故而测度为0。所谓“重叠部分测度为0”，就是上文所言“不重叠或者重叠部分可忽略”的意思。

我们就得到了WGAN前作中关于生成器梯度消失的第一个论证：在（近似）最优判别器下，最小化生成器的loss等价于最小化 P_r 与 P_g 之间的JS散度，而由于 P_r 与 P_g 几乎不可能有不可忽略的重叠，所以无论它们相距多远JS散度都是常数 $\log 2$ ，最终导致生成器的梯度（近似）为0，梯度消失。

接着作者写了很多公式定理从第二个角度进行论证，但是背后的思想也可以直观地解释：

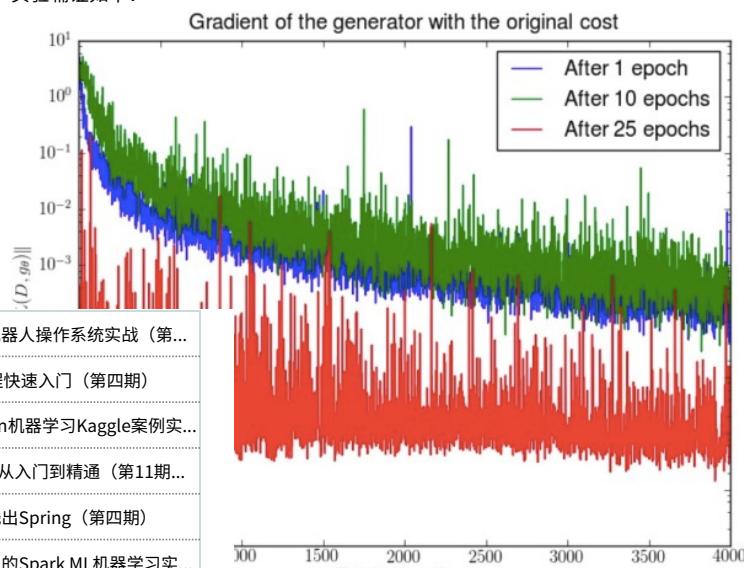
» 最新活动

» 联系客服

- 首先， P_r 与 P_g 之间几乎不可能有不可忽略的重叠，所以无论它们之间的“缝隙”多狭小，都肯定存在一个最优分割曲面把它们隔开，最多就是在那些可忽略的重叠处隔不开而已。
- 由于判别器作为一个神经网络可以无限拟合这个分隔曲面，所以存在一个最优判别器，对几乎所有真实样本给出概率1，对几乎所有生成样本给出概率0，而那些隔不开的部分就是难以被最优判别器分类的样本，但是它们的测度为0，可忽略。
- 最优判别器在真实分布和生成分布的支撑集上给出的概率都是常数（1和0），导致生成器的loss梯度为0，梯度消失。

有了这些理论分析，原始GAN不稳定的原因就彻底清楚了：判别器训练得太好，生成器梯度消失，生成器loss降不下去；判别器训练得不好，生成器梯度不准，四处乱跑。只有判别器训练得不好不坏才行，但是这个火候又很难把握，甚至在同一轮训练的前后不同阶段这个火候都可能不一样，所以GAN才那么难训练。

实验辅证如下：



特别将DCGAN训练1, 20, 25个epoch，然后固定生成器不说从头开始训练，对于第一种形式的生成器loss产生的梯度可以进行对比。可以看到随着判别器的训练，生成器的梯度均迅速衰减。注意y轴是对数坐标轴。

第二种原始GAN形式的问题

一句话概括：最小化第二种生成器loss函数，会等价于最小化一个不合理的距离衡量，导致两个问题，一是梯度不稳定，二是collapse mode即多样性不足。WGAN前作又是从两个角度进行了论证，下面只说第一个角度，因为对于第二个角度我难以找到一个直观的解释方式，感兴趣的读者还是去看论文吧（逃）。

» 最新活动

» 联系客服

如前文所说，Ian Goodfellow提出的“- log D trick”是把生成器loss改成

$$\mathbb{E}_{x \sim P_g} [-\log D(x)] \quad (\text{公式3})$$

上文推导已经得到在最优判别器 D^* 下

$$\mathbb{E}_{x \sim P_r} [\log D^*(x)] + \mathbb{E}_{x \sim P_g} [\log(1 - D^*(x))] = 2JS(P_r || P_g) - 2\log 2 \quad (\text{公式9})$$

我们可以把KL散度（注意下面是先g后r）变换为含 D^* 的形式：

$$\begin{aligned} KL(P_g || P_r) &= \mathbb{E}_{x \sim P_g} [\log \frac{P_g(x)}{P_r(x)}] \\ &= \mathbb{E}_{x \sim P_g} [\log \frac{P_g(x)/(P_r(x) + P_g(x))}{P_r(x)/(P_r(x) + P_g(x))}] \quad (\text{公式10}) \\ &= \mathbb{E}_{x \sim P_g} [\log \frac{1 - D^*(x)}{D^*(x)}] \\ &= \mathbb{E}_{x \sim P_g} \log[1 - D^*(x)] - \mathbb{E}_{x \sim P_g} \log D^*(x) \end{aligned}$$

由公式3, 9, 10可得最小化目标的等价变形

$$\begin{aligned} \mathbb{E}_{x \sim P_g} [-\log D^*(x)] &= KL(P_g || P_r) - \mathbb{E}_{x \sim P_g} \log[1 - D^*(x)] \\ &= KL(P_g || P_r) - 2JS(P_r || P_g) + 2\log 2 + \mathbb{E}_{x \sim P_r} [\log D^*(x)] \end{aligned}$$

注意上式最后两项不依赖于生成器G，最终得到最小化公式3等价于最小化

$$KL(P_g || P_r) - 2JS(P_r || P_g) \quad (\text{公式11})$$

深入浅出机器学习与深度学习	
ROS机器人操作系统实战（第三期）	这个问题有两个严重的问题。第一是它同时要最小化生成分布与真实分布的KL散度，一个要拉近，一个却要推远！这在直观上非常荒谬，在数值上是后面那个JS散度项的毛病。
Qt编程快速入门（第四期）	的KL散度项也有毛病。因为KL散度不是一个对称的衡量，是有差别的。以前者为例
Python机器学习Kaggle案例实... Mycat从入门到精通（第11期... 深入浅出Spring（第四期） 黄美灵的Spark ML机器学习实... R语言魔鬼训练营（第八期） Python突击—从入门到精通到...	$\rightarrow 1$ 时， $P_g(x) \log \frac{P_g(x)}{P_r(x)} \rightarrow 0$ ，对 $KL(P_g P_r)$ 贡献趋近0 $\rightarrow 0$ 时， $P_g(x) \log \frac{P_g(x)}{P_r(x)} \rightarrow +\infty$ ，对 $KL(P_g P_r)$ 贡献趋近正无穷 上面两种错误的惩罚是不一样的，第一种错误对应的是“生成器没能力；第二种错误对应的是“生成器生成了不真实的样本”，惩罚巨大。

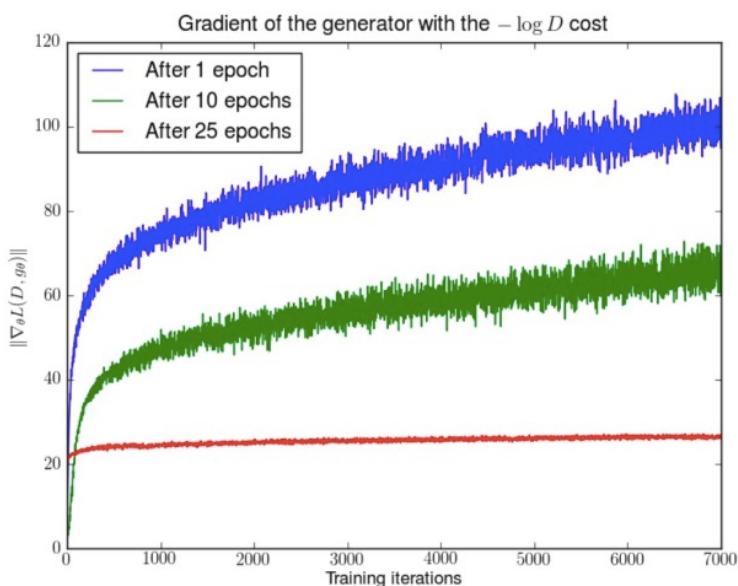
第一种错误对应的是缺乏多样性，第二种错误对应的是缺乏准确性。这一放一打之下，生成器宁可多生成一些重复但是很“安全”的样本，也不愿意去生成多样性的样本，因为那样一不小心就会产生第二种错误，得不偿失。这种现象就是大家常说的collapse mode。

第一部分小结：在原始GAN的（近似）最优判别器下，第一种生成器loss面临梯度消失问题，第二种生成器loss面临优化目标荒谬、梯度不稳定、对多样性与准确性惩罚不平衡导致mode collapse这几个问题。

实验辅证如下：

» 最新活动

» 联系客服



WGAN前作Figure 3。先分别将DCGAN训练1, 20, 25个epoch, 然后固定生成器不动, 判别器重新随机初始化从头开始训练, 对于第二种形式的生成器loss产生的梯度可以打印出其尺度的变化曲线, 可以看到随着判别器的训练, 蓝色和绿色曲线中生成器的梯度迅速增长, 说明梯度不稳定, 红线对应的是DCGAN相对收敛的状态, 梯度才比较稳定。

ROS机器人操作系统实战（第...
Qt编程快速入门（第四期）
Python机器学习Kaggle案例实...
Mycat从入门到精通（第11期...
深入浅出Spring（第四期）
黄美灵的Spark ML机器学习实...
R语言魔鬼训练营（第八期）
Python突击—从入门到精通到...

-一个过渡解决方案
总结为两点, 一是等价优化的距离衡量 (KL散度、JS散度)
初始化后的生成分布很难与真实分布有不可忽略的重叠。

第二点提出了一个解决方案, 就是对生成样本和真实样本加噪
向两个低维流形“弥散”到整个高维空间, 强行让它们产生
存在重叠, JS散度就能真正发挥作用, 此时如果两个分布越
向部分重叠得越多, JS散度也会越小而不会一直是一个常
数。GAN形式下) 梯度消失的问题就解决了。在训练过程中, 我
们可以通过对两个分布的噪声进行退火 (annealing), 慢慢减小其方差, 到后面两个低维流形
“本体”都已经有重叠时, 就算把噪声完全拿掉, JS散度也能照样发挥作用, 继续产
生有意义的梯度把两个低维流形拉近, 直到它们接近完全重合。以上是对原文的直观
解释。

在这个解决方案下我们可以放心地把判别器训练到接近最优, 不必担心梯度消失的问
题。而当判别器最优时, 对公式9取反可得判别器的最小loss为

$$\begin{aligned}\min L_D(P_{r+\epsilon}, P_{g+\epsilon}) &= -\mathbb{E}_{x \sim P_{r+\epsilon}}[\log D^*(x)] - \mathbb{E}_{x \sim P_{g+\epsilon}}[\log(1 - D^*(x))] \\ &= 2 \log 2 - 2JS(P_{r+\epsilon} || P_{g+\epsilon})\end{aligned}$$

其中 $P_{r+\epsilon}$ 和 $P_{g+\epsilon}$ 分别是加噪后的真实分布与生成分布。反过来说, 从最优判别器的loss可以
反推出当前两个加噪分布的JS散度。两个加噪分布的JS散度可以在某种程度上代表两个原本
分布的距离, 也就是说可以通过最优判别器的loss反映训练进程!真的有这样好事吗?

并没有, 因为加噪JS散度的具体数值受到噪声的方差影响, 随着噪声的退火, 前后的数值就沒
法比较了, 所以它不能成为 P_r 和 P_g 距离的本质性衡量。

因为本文的重点是WGAN本身, 所以WGAN前作的加噪方案简单介绍到这里, 感兴趣的读者可
以阅读原文了解更多细节。加噪方案是针对原始GAN问题的第二点根源提出的, 解决了训练不
稳定的问题, 不需要小心平衡判别器训练的火候, 可以放心地把判别器训练到接近最优, 但是
仍然没能够提供一个衡量训练进程的数值指标。但是WGAN本作就从第一点根源出发, 用
Wasserstein距离代替JS散度, 同时完成了稳定训练和进程指标的问题!

作者未对此方案进行实验验证。

第三部分: Wasserstein距离的优越性质

» 最新活动
» 联系客服

Wasserstein距离又叫Earth-Mover (EM) 距离，定义如下：

$$W(P_r, P_g) = \inf_{\gamma \sim \Pi(P_r, P_g)} \mathbb{E}_{(x,y) \sim \gamma} [| | | x - y | | |] \quad (\text{公式12})$$

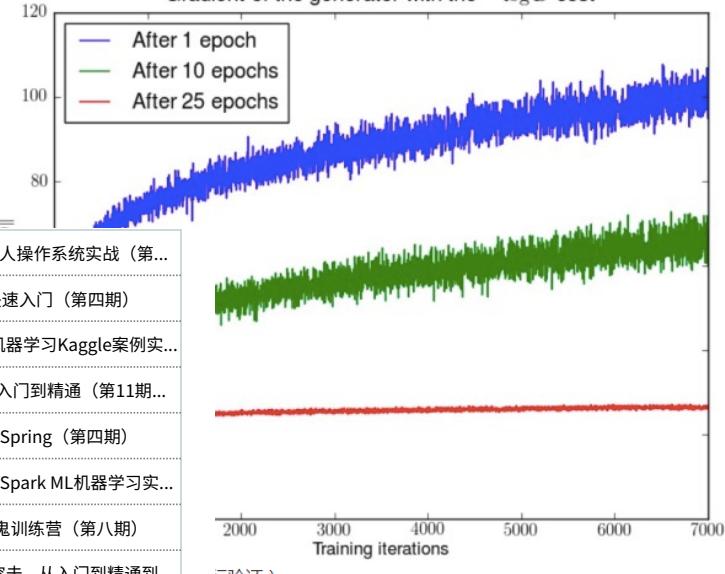
解释如下： $\Pi(P_r, P_g)$ 是 P_r 和 P_g 组合起来的所有可能的联合分布的集合，反过来说， $\Pi(P_r, P_g)$ 中每一个分布的边缘分布都是 P_r 和 P_g 。对于每一个可能的联合分布 γ 而言，可以从中采样 $(x, y) \sim \gamma$ 得到一个真实样本 x 和一个生成样本 y ，并算出这对样本的距离 $| | | x - y | | |$ ，所以可以计算该联合分布 γ 下样本对距离的期望值 $\mathbb{E}_{(x,y) \sim \gamma} [| | | x - y | | |]$ 。在所有可能的联合分布中能够对这个期望值取到的下界 $\inf_{\gamma \sim \Pi(P_r, P_g)} \mathbb{E}_{(x,y) \sim \gamma} [| | | x - y | | |]$ ，就定义为Wasserstein距离。

直观上可以把 $\mathbb{E}_{(x,y) \sim \gamma} [| | | x - y | | |]$ 理解为在 γ 这个“路径规划”下把 P_r 这堆“沙土”挪到 P_g “位置”所需的“消耗”，而 $W(P_r, P_g)$ 就是“最优路径规划”下的“最小消耗”，所以才叫Earth-Mover (推土机) 距离。

Wasserstein距离相比KL散度、JS散度的优越性在于，即使两个分布没有重叠，

Wasserstein距离仍然能够反映它们的远近。WGAN本作通过简单的例子展示了这一点。考虑如下二维空间中的两个分布 P_1 和 P_2 ， P_1 在线段AB上均匀分布， P_2 在线段CD上均匀分布，通过控制参数 θ 可以控制着两个分布的距离远近。

Gradient of the generator with the $-\log D$ cost



- ROS机器人操作系统实战（第...
- Qt编程快速入门（第四期）
- Python机器学习Kaggle案例实...
- Mycat从入门到精通（第11期...
- 深入浅出Spring（第四期）
- 黄美灵的Spark ML机器学习实...
- R语言魔鬼训练营（第八期）
- Python突击—从入门到精通到...

» 最新活动

» 联系客服

$$KL(P_1 || P_2) = KL(P_1 || P_2) = \begin{cases} +\infty & \text{if } \theta \neq 0 \\ 0 & \text{if } \theta = 0 \end{cases} \quad (\text{突变})$$

$$JS(P_1 || P_2) = \begin{cases} \log 2 & \text{if } \theta \neq 0 \\ 0 & \text{if } \theta = 0 \end{cases} \quad (\text{突变})$$

$$W(P_0, P_1) = |\theta| \quad (\text{平滑})$$

KL散度和JS散度是突变的，要么最大要么最小，Wasserstein距离却是平滑的，如果我们要用梯度下降法优化 θ 这个参数，前两者根本提供不了梯度，Wasserstein距离却可以。类似地，在高维空间中如果两个分布不重叠或者重叠部分可忽略，则KL和JS既反映不了远近，也提供不了梯度，但是Wasserstein却可以提供有意义的梯度。

第四部分：从Wasserstein距离到WGAN

既然Wasserstein距离有如此优越的性质，如果我们能够把它定义为生成器的loss，不就可以产生有意义的梯度来更新生成器，使得生成分布被拉向真实分布吗？

没那么简单，因为Wasserstein距离定义（公式12）中的 $\inf_{\gamma \sim \Pi(P_r, P_g)}$ 没法直接求解，不过没关系，作者用了一个已有的定理把它变换为如下形式

$$W(P_r, P_g) = \frac{1}{K} \sup_{\|f\|_L \leq K} \mathbb{E}_{x \sim P_r}[f(x)] - \mathbb{E}_{x \sim P_g}[f(x)] \quad (\text{公式13})$$

证明过程被作者丢到论文附录中了，我们也姑且不管，先看看上式究竟说了什么。

首先需要介绍一个概念——Lipschitz连续。它其实就是在连续函数 f 上面额外施加了一个限制，要求存在一个常数 $K \geq 0$ 使得定义域内的任意两个元素 x_1 和 x_2 都满足

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

此时称函数 f 的Lipschitz常数为 K 。

简单理解，比如说 f 的定义域是实数集合，那上面的要求就等价于 f 的导函数绝对值不超过 K 。再比如说 $\log(x)$ 就不是Lipschitz连续，因为它的导函数没有上界。Lipschitz连续条件限制了一个连续函数的最大局部变动幅度。

公式13的意思就是在要求函数 f 的Lipschitz常数 $\|f\|_L$ 不超过 K 的条件下，对所有可能满足条件的 f 取到 $\mathbb{E}_{x \sim P_r}[f(x)] - \mathbb{E}_{x \sim P_g}[f(x)]$ 的上界，然后再除以 K 。特别地，我们可以用一组参数 w 来定义一系列可能的函数 f_w ，此时求解公式13可以近似变成求解如下形式

$$K \cdot W(P_r, P_g) \approx \max_{w: \|f_w\|_L \leq K} \mathbb{E}_{x \sim P_r}[f_w(x)] - \mathbb{E}_{x \sim P_g}[f_w(x)] \quad (\text{公式14})$$

再用上我们搞深度学习的人最熟悉的那一套，不就可以把 f 用一个带参数 w 的神经网络来表示嘛！由于神经网络的拟合能力足够强大，我们有理由相信，这样定义出来的一系列 f_w 虽然无法囊括所有可能，但是也足以高度近似公式13要求的那个 $\sup_{\|f\|_L \leq K}$ 了。

ROS机器人操作系统实战（第...

Qt编程快速入门（第四期）

Python机器学习Kaggle案例实...

Mycat从入门到精通（第11期...

深入浅出Spring（第四期）

黄美灵的Spark ML机器学习实...

R语言魔鬼训练营（第八期）

Python突击—从入门到精通到...

式14中 $\|f_w\|_L \leq K$ 这个限制。我们其实不关心具体的 K 是多少，只要 \exists 只是会使得梯度变大 K 倍，并不影响梯度的方向。所以作者采，就是限制神经网络 f_θ 的所有参数 w_i 的不超过某个范围 $[-c, c]$ ，比关于输入样本 x 的导数 $\frac{\partial f_w}{\partial x}$ 也不会超过某个范围，所以一定存在某的局部变动幅度不会超过它，Lipschitz连续条件得以满足。具体在更新完 w 后把它clip回这个范围就可以了。

一个含参数 w 、最后一层不是非线性激活层的判别器网络 f_w ，在限生下，使得

$$[f_w(x)] \quad (\text{公式15})$$

尽可能取到最大，此时 L 就会近似真实分布与生成分布之间的Wasserstein距离（忽略常数倍数 K ）。注意原始GAN的判别器做的是真假二分类任务，所以最后一层是sigmoid，但是在WGAN中的判别器 f_w 做的是近似拟合Wasserstein距离，属于回归任务，所以要把最后一层的sigmoid拿掉。

接下来生成器要近似地最小化Wasserstein距离，可以最小化 L ，由于Wasserstein距离的优良性质，我们不需要担心生成器梯度消失的问题。再考虑到 L 的第一项与生成器无关，就得到了WGAN的两个loss。

$$-\mathbb{E}_{x \sim P_g}[f_w(x)] \quad (\text{公式16, WGAN生成器loss函数})$$

$$\mathbb{E}_{x \sim P_g}[f_w(x)] - \mathbb{E}_{x \sim P_r}[f_w(x)] \quad (\text{公式17, WGAN判别器loss函数})$$

公式15是公式17的反，可以指示训练进程，其数值越小，表示真实分布与生成分布的Wasserstein距离越小，GAN训练得越好。

WGAN完整的算法流程已经贴过了，为了方便读者此处再贴一遍：

» 最新活动

» 联系客服

Algorithm 1 WGAN, our proposed algorithm. All experiments in the paper used 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 , the batch size. n_{critic} , the number of iterations of the critic per generator iteration.

Require: : w_0 , initial critic parameters. θ_0 , initial generator's parameters.

- 1: **while** θ has not converged **do**
- 2: **for** $t = 0, \dots, n_{\text{critic}}$ **do**
- 3: Sample $\{x^{(i)}\}_{i=1}^m \sim \mathbb{P}_r$ a batch from the real data.
- 4: Sample $\{z^{(i)}\}_{i=1}^m \sim p(z)$ a batch of prior samples.
- 5: $g_w \leftarrow \nabla_w [\frac{1}{m} \sum_{i=1}^m f_w(x^{(i)}) - \frac{1}{m} \sum_{i=1}^m f_w(g_\theta(z^{(i)}))]$
- 6: $w \leftarrow w + \alpha \cdot \text{RMSPProp}(w, g_w)$
- 7: $w \leftarrow \text{clip}(w, -c, c)$
- 8: **end for**
- 9: Sample $\{z^{(i)}\}_{i=1}^m \sim p(z)$ a batch of prior samples.
- 10: $g_\theta \leftarrow -\nabla_\theta \frac{1}{m} \sum_{i=1}^m f_w(g_\theta(z^{(i)}))$
- 11: $\theta \leftarrow \theta - \alpha \cdot \text{RMSPProp}(\theta, g_\theta)$
- 12: **end while**

上文说过，WGAN与原始GAN第一种形式相比，只改了四点：

判别器最后一层去掉sigmoid

生成器和判别器的loss不取log

每次更新判别器的参数之后把它们的值截断到不超过一个固定常数c

不要用基于动量的优化算法（包括momentum和Adam），推荐RMSPProp，SGD也行

前两点都是从理论分析中得到的，已经介绍完毕；第四点却是作者从实验中发现的，

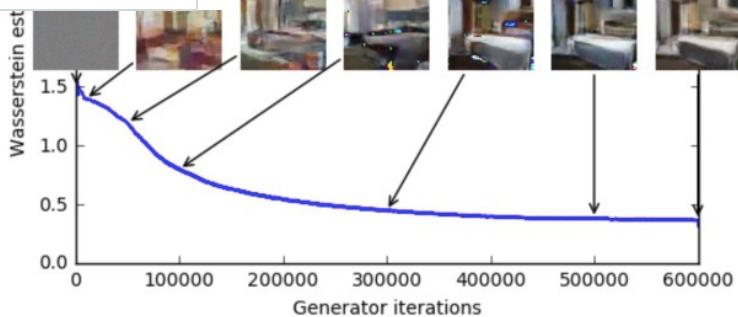
属于trick，相对比较“玄”。作者发现如果使用Adam，判别器的loss有时候会崩掉，

当它崩掉时，Adam给出的更新方向与梯度方向夹角的cos值就变成负数，更新方向与

ROS机器人操作系统实战（第三期）	意味着判别器的loss梯度是不稳定的，所以不适合用Adam这类
Qt编程快速入门（第四期）	首先改用RMSPProp之后，问题就解决了，因为RMSPProp适合梯
Python机器学习Kaggle案例实...	
Mycat从入门到精通（第11期...）	
深入浅出Spring（第四期）	验证，本文只提比较重要的三点。第一，判别器所近似的
黄美灵的Spark ML机器学习实...	生成图片质量高度相关，如下所示（此即题图）：
R语言魔鬼训练营（第八期）	
Python突击—从入门到精通到...	

» 最新活动

» 联系客服



第二， WGAN如果用类似DCGAN架构，生成图片的效果与DCGAN差不多：



但是厉害的地方在于WGAN不用DCGAN各种特殊的架构设计也能做到不错的效果，比如如果大家一起拿掉Batch Normalization的话，DCGAN就崩了：



如果WGAN和原始GAN都使用多层全连接网络（MLP），不用CNN，WGAN质量会变差些，但是原始GAN不仅质量变得更差，而且还出现了collapse mode，即多样性不足：



第三，在所有WGAN的实验中未观察到collapse mode，作者也只说应该是解决了，

最后补充一点论文没提到，但是我个人觉得比较微妙的问题。判别器所近似的Wasserstein距离能够用来指示单次训练中的训练进程，这个没错；接着作者又说它可以用于比较多次训练进程，指引调参，我倒是觉得需要小心些。比如说我下次训练时改了判别器的层数、节点数等超参，判别器的拟合能力就必然有所波动，再比如说我

ROS机器人操作系统实战（第...
Qt编程快速入门（第四期）
Python机器学习Kaggle案例实...
Mycat从入门到精通（第11期...
深入浅出Spring（第四期）
黄美灵的Spark ML机器学习实...
R语言魔鬼训练营（第八期）
Python突击—从入门到精通到...

欠迭代之间，判别器的迭代次数，这两种常见的变动都会使误差就与上次不一样。那么这个拟合误差的变动究竟有多

令时判别器的拟合能力或迭代次数相差实在太大，那它们之

旨标，我都是存疑的。

进一步指出，相比于判别器迭代次数的改变，对判别器架构超参的改

：chitz常数 K ，进而改变近似Wasserstein距离的倍数，前后两轮

了，这是需要在实际应用中注意的。对此我想到了一个工程化的解

样一对生成分布和真实分布，让前后两个不同架构的判别器各自拟

差多少倍，可以近似认为是后面的 K_2 相对前面 K_1 的变化倍数，于

：正前后两轮训练的指标。

第五部分：总结

WGAN前作分析了Ian Goodfellow提出的原始GAN两种形式各自的问题，第一种形式等价在最优判别器下等价于最小化生成分布与真实分布之间的JS散度，由于随机生成分布很难与真实分布有不可忽略的重叠以及JS散度的突变特性，使得生成器面临梯度消失的问题；第二种形式在最优判别器下等价于既要最小化生成分布与真实分布直接的KL散度，又要较大化其JS散度，相互矛盾，导致梯度不稳定，而且KL散度的不对称性使得生成器宁可丧失多样性也不愿丧失准确性，导致collapse mode现象。

WGAN前作针对分布重叠问题提出了一个过渡解决方案，通过对生成样本和真实样本加噪声使得两个分布产生重叠，理论上可以解决训练不稳定的问题，可以放心训练判别器到接近最优，但是未能提供一个指示训练进程的可靠指标，也未做实验验证。

WGAN本作引入了Wasserstein距离，由于它相对KL散度与JS散度具有优越的平滑特性，理论上可以解决梯度消失问题。接着通过数学变换将Wasserstein距离写成可求解的形式，利用一个参数数值范围受限的判别器神经网络来较大化这个形式，就可以近似Wasserstein距离。在此近似最优判别器下优化生成器使得Wasserstein距离缩小，就能有效拉近生成分布与真实分布。WGAN既解决了训练不稳定的问题，也提供了一个可靠的训练进程指标，而且该指标确实与生成样本的质量高度相关。作者对WGAN进行了实验验证。

[欢迎加入本站公开兴趣群](#)

» 最新活动

» 联系客服

商业智能与数据分析群

兴趣范围包括各种让数据产生价值的办法，实际应用案例分享与讨论，分析工具，ETL工具，数据仓库，数据挖掘工具，报表系统等全方位知识

QQ群：81035754



鲜花 握手 雷人 路过 鸡蛋

[邀请](#) [分享](#) [收藏](#) [分享到新浪微博](#)

上一篇：最近比较火的三个GAN应用及代码--Pix2pix

下一篇：深度学习的难点

最新评论



评论

[登录](#) | [注册](#)



顺便说点什么吧.....

还可以输入 **140** 字

还没有人评论过, 赶快抢沙发吧!

- [ROS机器人操作系统实战（第...](#)
- [Qt编程快速入门（第四期）](#)
- [Python机器学习Kaggle案例实...](#)
- [Mycat从入门到精通（第11期...](#)
- [深入浅出Spring（第四期）](#)
- [黄美灵的Spark ML机器学习实...](#)
- [R语言魔鬼训练营（第八期）](#)
- [Python突击—从入门到精通到...](#)

[到微博](#)

» 最新活动
» 联系客服



订阅号 小程序

[关于我们](#) [手机版](#) [友情链接](#) [站点统计](#) [文本模式](#) [小游戏](#)

版权所有 广州市皓岚信息技术有限公司 合作伙伴 中山大学海量数据与云计算研究中心 粤ICP备08028958号

CopyRight 2011-2015 dataguru.cn All Right Reserved.