

## 15 WGAN新方案：通过梯度归一化来实现L约束

Nov By 苏剑林 | 2021-11-15 | 16288位读者

当前，WGAN主流的实现方式包括参数裁剪（Weight Clipping）、谱归一化（Spectral Normalization）、梯度惩罚（Gradient Penalty），本来则来介绍一种新的实现方案：梯度归一化（Gradient Normalization），该方案出自两篇有意思的论文，分别是《[Gradient Normalization for Generative Adversarial Networks](#)》和《[Gr aN-GAN: Piecewise Gradient Normalization for Generative Adversarial Networks](#)》。

有意思在什么地方呢？从标题可以看到，这两篇论文应该是高度重合的，甚至应该是同一作者的。但事实上，这是两篇不同团队的、大致是同一时期的论文，一篇中了ICCV，一篇中了WACV，它们基于同样的假设推出了几乎一样的解决方案，内容重合度之高让我一直以为是同一篇论文。果然是巧合无处不在啊~

## 基础回顾 #

关于WGAN，我们已经介绍过多次，比如《[互怼的艺术：从零直达WGAN-GP](#)》和《[从Wasserstein距离、对偶理论到WGAN](#)》，这里就不详细重复了。简单来说，WGAN的迭代形式为：

$$\min_G \max_{\|D\|_L \leq 1} \mathbb{E}_{x \sim p(x)} [D(x)] - \mathbb{E}_{z \sim q(z)} [D(G(z))] \quad (1)$$

这里的关键是判别器 $D$ 是一个带约束优化问题，需要在优化过程中满足L约束 $\|D\|_L \leq 1$ ，所以WGAN的实现难度就是如何往 $D$ 里边引入该约束。

这里再普及一下，如果存在某个常数 $C$ ，使得定义域中的任意 $x, y$ 都满足 $|f(x) - f(y)| \leq C\|x - y\|$ ，那么我们称 $f(x)$ 满足Lipschitz约束（L约束），其中 $C$ 的最小值，我们称为Lipschitz常数（L常数），记为 $\|f\|_L$ 。所以，对于WGAN判别器来说，要做到两步：1、 $D$ 要满足L约束；2、L常数要不超过1。

事实上，当前我们主流的神经网络模型，都是“线性组合+非线性激活函数”的形式，而主流的激活函数是“近线性的”，比如ReLU、LeakyReLU、SoftPlus等，它们的导函数的绝对值都不超过1，所以当前主流模型其实都满足L约束，所以关键是如何让L常数不超过1，当然其实也不用非1不可，能保证它不超过某个固定常数就行。

## 方案简介 #

参数裁剪和谱归一化的思路是相似的，它们都是通过约束参数，保证模型每一层的L常数都有界，所以总的L常数也有界；而梯度惩罚则是留意到 $\|D\|_L \leq 1$ 的一个充分条件是 $\|\nabla_x D(x)\| \leq 1$ ，所以就通过惩罚项 $(\|\nabla_x D(x)\| - 1)^2$ 来施加“软约束”。

本文介绍的梯度归一化，也是基于同样的充分条件，它利用梯度将 $D(x)$ 变换为 $\hat{D}(x)$ ，使其自动满足 $\|\nabla_x \hat{D}(x)\| \leq 1$ 。具体来说，我们通常用ReLU或LeakyReLU作为激活函数，在这个激活函数之下， $D(x)$ 实际上是一个“分段线性函数”，这就意味着，除了边界之外， $D(x)$ 在局部的连续区域内都是一个线性函数，相应地， $\nabla_x D(x)$ 就是一个常向量。

于是梯度归一化就想着令 $\hat{D}(x) = D(x) / \|\nabla_x D(x)\|$ ，这样一来就有

$$\|\nabla_x \hat{D}(x)\| = \left\| \nabla_x \left( \frac{D(x)}{\|\nabla_x D(x)\|} \right) \right\| = \left\| \frac{\nabla_x D(x)}{\|\nabla_x D(x)\|} \right\| = 1 \quad (2)$$

当然，这样可能会有除0错误，所以两篇论文提出了不同的解决方案，第一篇（ICCV论文）直接将 $|D(x)|$ 也加到了分母中，连带保证了函数的有界性：

$$\hat{D}(x) = \frac{D(x)}{\|\nabla_x D(x)\| + |D(x)|} \in [-1, 1]$$

(3)

第二篇（WACV论文）则是比较朴素地加了个 $\epsilon$ ：

$$\hat{D}(x) = \frac{D(x) \cdot \|\nabla_x D(x)\|}{\|\nabla_x D(x)\|^2 + \epsilon}$$

(4)

同时第二篇也提到试验过 $\hat{D}(x) = D(x)/(\|\nabla_x D(x)\| + \epsilon)$ ，效果略差但差不多。

## 实验结果 #

现在我们先来看看实验结果。当然，能双双中顶会，实验结果肯定是正面的，部分结果如下图：

Table 3: Inception Score and FID with unconditional image generation on CIFAR-10 and STL-10. We report the average and standard deviation of the results trained with 5 different random seeds. Note that “-” denotes that result is not reported by the original paper. Moreover, † represents that the original paper does not provide an evaluation on STL-10, so we provide a implementation for reference. ‡ denotes that we provide a re-implementation result for reliable comparison.

Method	CIFAR-10			STL-10		
	Inception Score↑	FID(train)↓	FID(test)↓	Inception Score↑	FID(50k) ↓	FID(10k) ↓
Real data	11.24±.12	0	7.80	26.08±.26	0	0
<i>Standard CNN</i>						
SN-GAN [22]	7.58±.12	-	25.50	8.79±.14	-	43.20
SN-GAN†	7.86±.09	18.52	22.67	8.87±.09	32.90	35.10
SN-GAN-CR [35]	7.93	-	18.72	8.69±.08†	32.11†	34.14†
(our) GN-GAN	7.71±.14	19.31±.76	23.52±.80	9.00±.15	30.18±.82	32.41±.73
(our) GN-GAN-CR	8.04±.19	18.59±1.5	22.89±1.5	9.00±.14	27.61±.69	29.53±.62
<i>ResNet</i>						
WGAN-GP [10]	7.86±.08	-	-	-	-	-
SN-GAN	8.22±.05	-	21.70±.21	9.10±.04	-	40.10±.50
SN-GAN†	8.48±.11	12.35	16.59	9.18±.10	29.16	31.85
SN-GAN-CR	8.40	-	14.56	9.38±.07†	25.78†	28.4†
(our) GN-GAN	8.49±.11	11.13±.18	15.33±.16	9.60±.14	26.14±.7	28.12±0.61
(our) GN-GAN-CR	<b>8.72±.11</b>	<b>9.55±.47</b>	<b>13.71±.40</b>	<b>9.74±.15</b>	<b>23.62±.89</b>	<b>25.80±0.59</b>
<i>Neural Architecture Search</i>						
AutoGAN [8]	8.55±.10	12.42	-	9.16±.12	31.01	-
E²GAN [28]	8.51±.13	11.26	-	9.51±.09	25.53	-

ICCV论文的实验结果表格

Table 2: Inception scores (IS), FIDs, and KIDs with unsupervised image generation on CIFAR-10, CIFAR-100, and STL-10. The best and the second best models per evaluation metric and GAN family (i.e., with discriminators or critics) are indicated by **bold red** and **bold blue** fonts. † indicates modified baselines with an altered Lipschitz constant  $\mathcal{K}$ . The table is split comparing discriminators (top) and critics (bottom). We write “-” for cases where a model did not achieve a FID < 70.

Method	IS ↑			FID ↓			KID (×1000) ↓		
	CIFAR-10	CIFAR-100	STL-10	CIFAR-10	CIFAR-100	STL-10	CIFAR-10	CIFAR-100	STL-10
NSGAN	7.655	6.611	7.920	23.750	30.842	44.179	14.5	20.5	40.0
NSGAN-GP	8.016	-	8.568	<b>15.813</b>	-	38.848	<b>12.9</b>	-	38.9
NSGAN-SN	7.792	7.258	8.167	20.998	25.564	<b>38.669</b>	15.7	18.4	<b>35.7</b>
NSGAN-GP†	<b>8.019</b>	<b>7.892</b>	<b>8.623</b>	15.911	<b>20.894</b>	40.110	13.1	<b>17.0</b>	39.8
NSGAN-SN†	7.814	7.526	8.135	20.323	24.200	39.013	15.3	17.7	36.7
GraND-GAN (Ours)	<b>8.031</b>	<b>8.314</b>	<b>8.743</b>	<b>14.965</b>	<b>18.978</b>	<b>35.226</b>	<b>12.3</b>	<b>13.7</b>	<b>35.0</b>
WGAN-GP	7.442	7.520	8.492	22.927	27.231	42.170	21.1	23.5	43.0
SNGAN	<b>8.112</b>	7.778	8.385	17.107	20.739	38.218	<b>12.6</b>	<b>14.3</b>	<b>34.3</b>
WGAN-GP†	7.344	7.684	8.466	22.705	25.211	42.595	20.6	21.2	44.7
SNGAN†	<b>7.991</b>	<b>7.959</b>	<b>8.552</b>	<b>16.740</b>	<b>20.104</b>	<b>36.203</b>	<b>12.0</b>	<b>14.3</b>	<b>33.3</b>
GraNC-GAN (Ours)	7.966	<b>8.208</b>	<b>8.957</b>	<b>16.361</b>	<b>19.131</b>	<b>35.770</b>	13.7	<b>14.8</b>	35.4

WACV论文的实验结果表格



Figure 1: Generated samples on CelebA-HQ 256×256. FID=7.67.



Figure 2: Generated samples on LSUN Church Outdoor 256×256. FID=5.405.

ICCV论文的生成效果演示

## 尚有疑问 #

结果看上去很好，理论看上去也没问题，还同时被两个顶会认可，看上去是一个好工作无疑了。然而，笔者的困惑才刚刚开始。

该工作最重要的问题是，如果按照分段线性函数的假设，那么 $D(x)$ 的梯度虽然在局部是一个常数，但整体来看它是不连续的（如果梯度全局连续又是常数，那么就是一个线性函数而不是分段线性了），然而 $D(x)$ 本身是一个连续函数，那么 $\hat{D}(x) = D(x)/\|\nabla_x D(x)\|$ 就是连续函数除以不连续函数，结果就是一个不连续的函数！

所以问题就来了，不连续的函数居然可以作为判别器，这看起来相当不可思议。要知道这个不连续并非只在某些边界点不连续，而是在两个区域之间的不连续，所以这个不连续是不可忽略的存在。在Reddit上，也有读者有着同样的疑问，但目前作者也没有给出合理的解释（[链接](#)）。

另一个问题是，如果分段线性函数的假设真的有效，那么我用 $\hat{D}(x) = \left\langle \frac{\nabla_x D(x)}{\|\nabla_x D(x)\|}, x \right\rangle$ 作为判别器，理论上应该是等价的，但笔者的实验结果显示这样的 $\hat{D}(x)$ 效果极差。所以，有一种可能性就是，梯度归一化确实是有效的，但其作用的原因并不像上面两篇论文分析的那么简单，也许有更复杂的生效机制我们还没发现。此外，也可能是我们对GAN的理解还远远不够充分，也就是说，对判别器的连续性等要求，也许远远不是我们所想的那样。

最后，在笔者的实验结果中，梯度归一化的效果并不如梯度惩罚，并且梯度惩罚仅仅是训练判别器的时候用到了二阶梯度，而梯度归一化则是训练生成器和判别器都要用到二阶梯度，所以梯度归一化的速度明显下降，显存占用量也明显增加。所以从个人实际体验来看，梯度归一化不算一个特别友好的方案。

## 文章小结 #

本文介绍了一种实现WGAN的新方案——梯度归一化，该方案形式上比较简单，论文报告的效果也还不错，但个人认为其中还有不少值得疑问之处。

转载到请包括本文地址：<https://kexue.fm/archives/8757>

**更详细的转载事宜请参考：《科学空间FAQ》**

**如果您需要引用本文，请参考：**

苏剑林. (Nov. 15, 2021). 《WGAN新方案：通过梯度归一化来实现L约束》[Blog post]. Retrieved from <https://kexue.fm/archives/8757>