GAN中的正则化和归一化研究

A Large-Scale Study on Regularization and Normalization in GANs

2019 年 6 月 22 日

GAN训练

- 存在训练深度神经网络相关的优化问题
- GAN的训练还对下面几种选择非常敏感：
  - 损失函数
  - 网络结构
  - 正则化和归一化s

GAN

- 论文对这些方法进行了全面的实验分析，通过超参数优化，在几个流行的大规模数据集上进行实验。
- 超参数集还参考了一些文献中提出的“好”的超参数集，以及通过贝叶斯优化获得的参数集。

一些概念

- 消融研究（ Ablation study ）： 指通过移除某个模型或者算法的某些特征，来观察这些特征对模型效果的影响。

生成器和判别器的结构

- DC-GAN
- ResNet:
  - 生成器中有5个ResNet块，在判别器中有6个ResNet块。
- 都使用Adam优化器训练。

评估准则(IS)

- Inception Score（IS）

$$IS(G) = exp(\mathbb{E}_{x \backsim p_g} D_{KL}(p(y|x)||p(y)))$$

其中：

exp：仅仅是为了好看，没有具体含义。

$x \backsim P_g$ ：表示从生成器中生图片。

$p(y|x)$: 判别器输出x的概率的分布

p_y: N 个生成的图片,各自得到一个自己的概率分布向量，把这些向量求一个平均,得到生成器生成的图片全体在所有类别上的边缘分布

评估准则(FID)

- FID

$$FID = \|\mu_x - \mu_y\|_2^2 + Tr(\Sigma_x + \Sigma_y - 2(\Sigma_x\Sigma_y)^{\frac{1}{2}})$$

其中（μx，Σx）和（μy，Σy）分别是来自P和Q的嵌入样本的均值和协方差。作者认为，FID与人类的判断相一致，并且比IS更加稳健。

数据集

- CIFAR10
  - 60K个训练图像和10K个测试图像
- CELEBA-HQ-128
  - 30K张图像，3K个作为测试集
- LSUN-BEDROOM
  - 300万张图像， 其中30588张作为测试集

结论

- 证明梯度惩罚以及谱归一化适用于复杂的网络结构
- 通过分析损失函数的影响，非饱和损失在数据集和超参数之间是足够稳定的
- 实验表明类似的结论适用于最新模型
- 当计算资源有限时，应将非饱和GAN损失和谱归一化视为将GAN应用于新数据集时的默认选择。 给定额外的计算资源，建议添加Gulrajani等人的梯度惩罚。

正则化和归一化

- 主要比较了下面几种归一化和正则化方法：
  - 批量归一化(BN)、层归一化(LN)、谱归一化(SN)
  - 梯度惩罚(GP)、 Dragan惩罚(DR)、 L2正则化
- 实验时设置损失为非饱和损失。

正则化和归一化（网络结构的选择）

(b) ResNet19 generator

LAYER	KERNEL	RS	OUTPUT
z	-	-	128
Linear	-	-	$4 \times 4 \times 512$
ResBlock	$[3, 3, 1]$	U	$8 \times 8 \times 512$
ResBlock	$[3, 3, 1]$	U	$16 \times 16 \times 256$
ResBlock	$[3, 3, 1]$	U	$32 \times 32 \times 256$
ResBlock	$[3, 3, 1]$	U	$64 \times 64 \times 128$
ResBlock	$[3, 3, 1]$	U	$128 \times 128 \times 64$
BN, ReLU	-	-	$128 \times 128 \times 64$
Conv	$[3, 3, 1]$	-	$128 \times 128 \times 3$
Sigmoid	-	-	$128 \times 128 \times 3$

Figure: 生成器结构.

(a) ResNet19 discriminator

LAYER	KERNEL	RS	OUTPUT
ResBlock	$[3, 3, 1]$	D	$64 \times 64 \times 64$
ResBlock	$[3, 3, 1]$	D	$32 \times 32 \times 128$
ResBlock	$[3, 3, 1]$	D	$16 \times 16 \times 256$
ResBlock	$[3, 3, 1]$	D	$8 \times 8 \times 256$
ResBlock	$[3, 3, 1]$	D	$4 \times 4 \times 512$
ResBlock	$[3, 3, 1]$	D	$2 \times 2 \times 512$
ReLU, MP	-	-	512
Linear	-	-	1

Figure: 判别器结构.

正则化和归一化（超参数选择）

PARAMETER	RANGE	LOG
Learning rate α	$[10^{-5}, 10^{-2}]$	Yes
λ for L_2	$[10^{-4}, 10^{1}]$	Yes
λ for non-L_2	$[10^{-1}, 10^{2}]$	Yes
$\beta_1 \times \beta_2$	$[0, 1] \times [0, 1]$	No

Table 2. We use sequential Bayesian optimization (Srinivas et al., 2010) to explore the hyperparameter settings from the specified ranges. We explore 120 hyperparameter settings in 12 rounds of optimization.

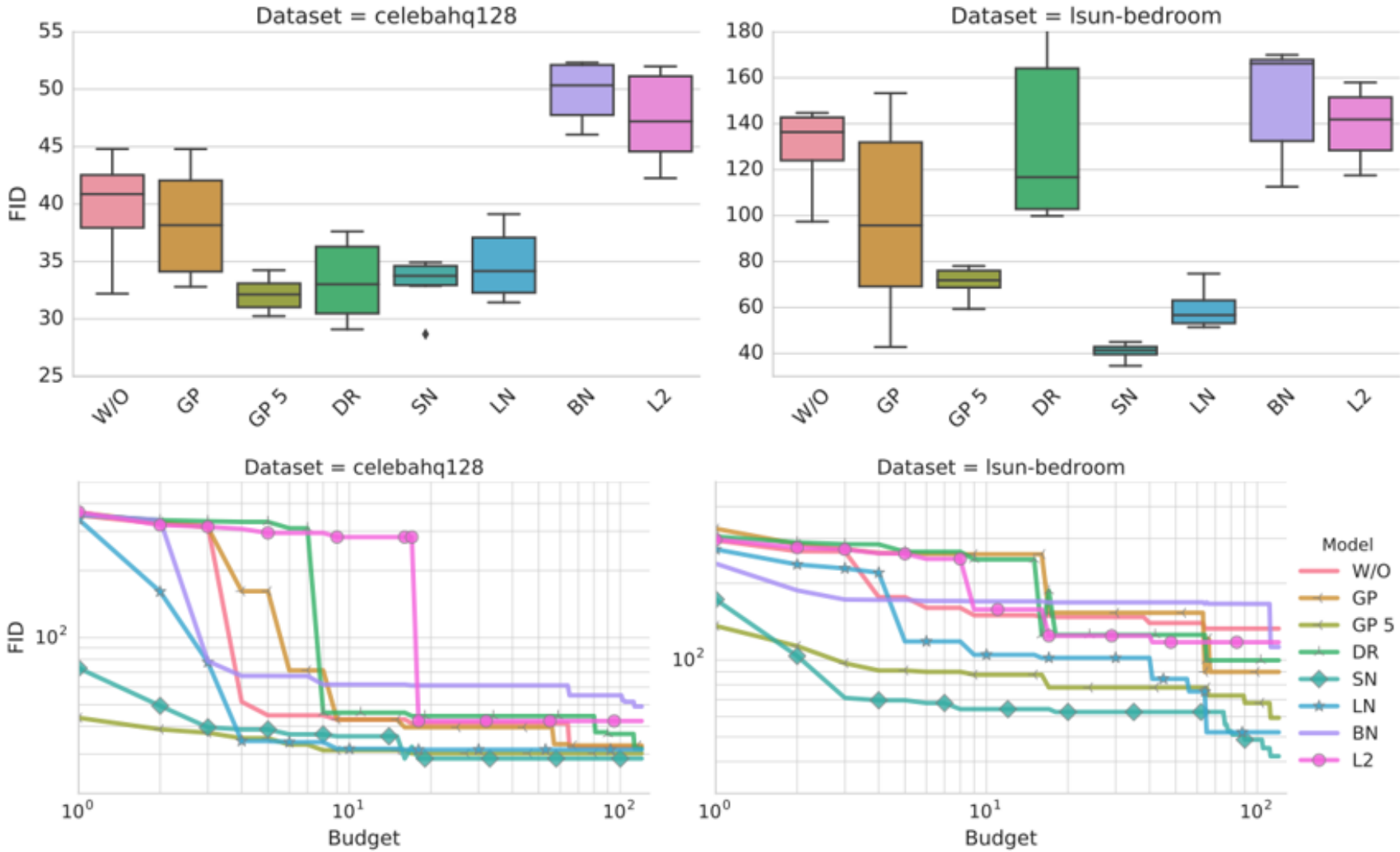
Figure: 贝叶斯优化设置.

PARAMETER	DISCRETE VALUE
Learning rate α	$\{0.0002, 0.0001, 0.001\}$
Reg. strength λ	$\{1, 10\}$
$(\beta_1, \beta_2, n_{dis})$	$\{(0.5, 0.900, 5), (0.5, 0.999, 1), (0.5, 0.999, 5), (0.9, 0.999, 5)\}$

Table 1. Hyperparameter ranges used in this study. The Cartesian product of the fixed values suffices to uncover most of the recent results from the literature.

Figure: 根据文献选择的.

正则化和归一化(结果)


Dataset = celebahq128
Dataset = lsun-bedroom
FID
55
50
45
40
35
30
25
180
160
140
120
100
80
60
40
W/O
GP
GP 5
DR
SN
LN
BN
L2
Dataset = celebahq128
Dataset = lsun-bedroom
FID
10^2
10^0
10^1
10^2
Budget
Model
W/O
GP
GP 5
DR
SN
LN
BN
L2

正则化和归一化(结论)

- 向判别器添加批量归一化会降低性能
- 梯度惩罚（GP）可以帮助降低FID，但它不能稳定训练
- 谱归一化有助于提高模型质量，比梯度惩罚具有更高的计算效率
- GP惩罚的模型可能受益于判别器与生成器更新比例为5:1
- 在一项单独的消融研究中，运行额外的10万步优化程序可能会提高GP惩罚模型的性能

损失函数

研究损失函数的变化是否也能得到上述的结论？

损失函数

GAN	Discriminator Loss	Generator Loss
MM GAN	$\mathcal{L}_{\mathrm{D}}^{\mathrm{GAN}} = -\mathbb{E}_{x\sim p_d}[\log(D(x))] - \mathbb{E}_{\hat{x}\sim p_g}[\log(1-D(\hat{x}))]$	$\mathcal{L}_{\mathrm{G}}^{\mathrm{GAN}} = \mathbb{E}_{\hat{x}\sim p_g}[\log(1-D(\hat{x}))]$
NS GAN	$\mathcal{L}_{\mathrm{D}}^{\mathrm{NSGAN}} = -\mathbb{E}_{x\sim p_d}[\log(D(x))] - \mathbb{E}_{\hat{x}\sim p_g}[\log(1-D(\hat{x}))]$	$\mathcal{L}_{\mathrm{G}}^{\mathrm{NSGAN}} = -\mathbb{E}_{\hat{x}\sim p_g}[\log(D(\hat{x}))]$
WGAN	$\mathcal{L}_{\mathrm{D}}^{\mathrm{WGAN}} = -\mathbb{E}_{x\sim p_d}[D(x)] + \mathbb{E}_{\hat{x}\sim p_g}[D(\hat{x})]$	$\mathcal{L}_{\mathrm{G}}^{\mathrm{WGAN}} = -\mathbb{E}_{\hat{x}\sim p_g}[D(\hat{x})]$
WGAN GP	$\mathcal{L}_{\mathrm{D}}^{\mathrm{WGANGP}} = \mathcal{L}_{\mathrm{D}}^{\mathrm{WGAN}} + \lambda\mathbb{E}_{\hat{x}\sim p_g}[(\|\nabla D(\alpha x + (1-\alpha\hat{x})\|_2 - 1)^2]$	$\mathcal{L}_{\mathrm{G}}^{\mathrm{WGANGP}} = -\mathbb{E}_{\hat{x}\sim p_g}[D(\hat{x})]$
LS GAN	$\mathcal{L}_{\mathrm{D}}^{\mathrm{LSGAN}} = -\mathbb{E}_{x\sim p_d}[(D(x)-1)^2] + \mathbb{E}_{\hat{x}\sim p_g}[D(\hat{x})^2]$	$\mathcal{L}_{\mathrm{G}}^{\mathrm{LSGAN}} = -\mathbb{E}_{\hat{x}\sim p_g}[(D(\hat{x}-1))^2]$
DRAGAN	$\mathcal{L}_{\mathrm{D}}^{\mathrm{DRAGAN}} = \mathcal{L}_{\mathrm{D}}^{\mathrm{GAN}} + \lambda\mathbb{E}_{\hat{x}\sim p_d+\mathcal{N}(0,c)}[(\|\nabla D(\hat{x})\|_2 - 1)^2]$	$\mathcal{L}_{\mathrm{G}}^{\mathrm{DRAGAN}} = \mathbb{E}_{\hat{x}\sim p_g}[\log(1-D(\hat{x}))]$
BEGAN	$\mathcal{L}_{\mathrm{D}}^{\mathrm{BEGAN}} = \mathbb{E}_{x\sim p_d}[\|x-\mathrm{AE}(x)\|_1] - k_t\mathbb{E}_{\hat{x}\sim p_g}[\|\hat{x}-\mathrm{AE}(\hat{x})\|_1]$	$\mathcal{L}_{\mathrm{G}}^{\mathrm{BEGAN}} = \mathbb{E}_{\hat{x}\sim p_g}[\|\hat{x}-\mathrm{AE}(\hat{x})\|_1]$

Figure: GAN的损失函数

损失函数

- 非饱和损失(NS)

$$\mathcal{L}_D^{NSGAN} = -\mathbb{E}_{x \backsim p_d}[log(D(x))] - \mathbb{E}_{\hat{x} \backsim p_g}[log(1 - D(\hat{x}))]$$

$$\mathcal{L}_G^{NSGAN} = \mathbb{E}_{\hat{x} \backsim p_g}[log(1 - D(\hat{x}))]$$

- 最小二乘损失(LS)

$$\mathcal{L}_D^{LSGAN} = -\mathbb{E}_{x \backsim p_d}[log(D(x) - 1)^2] + \mathbb{E}_{\hat{x} \backsim p_g}[log(D(\hat{x}))]$$

$$\mathcal{L}_G^{LSGAN} = -\mathbb{E}_{\hat{x} \backsim p_g}[log(D(\hat{x}) - 1)^2]$$

- Wasserstein loss

损失函数(结果)

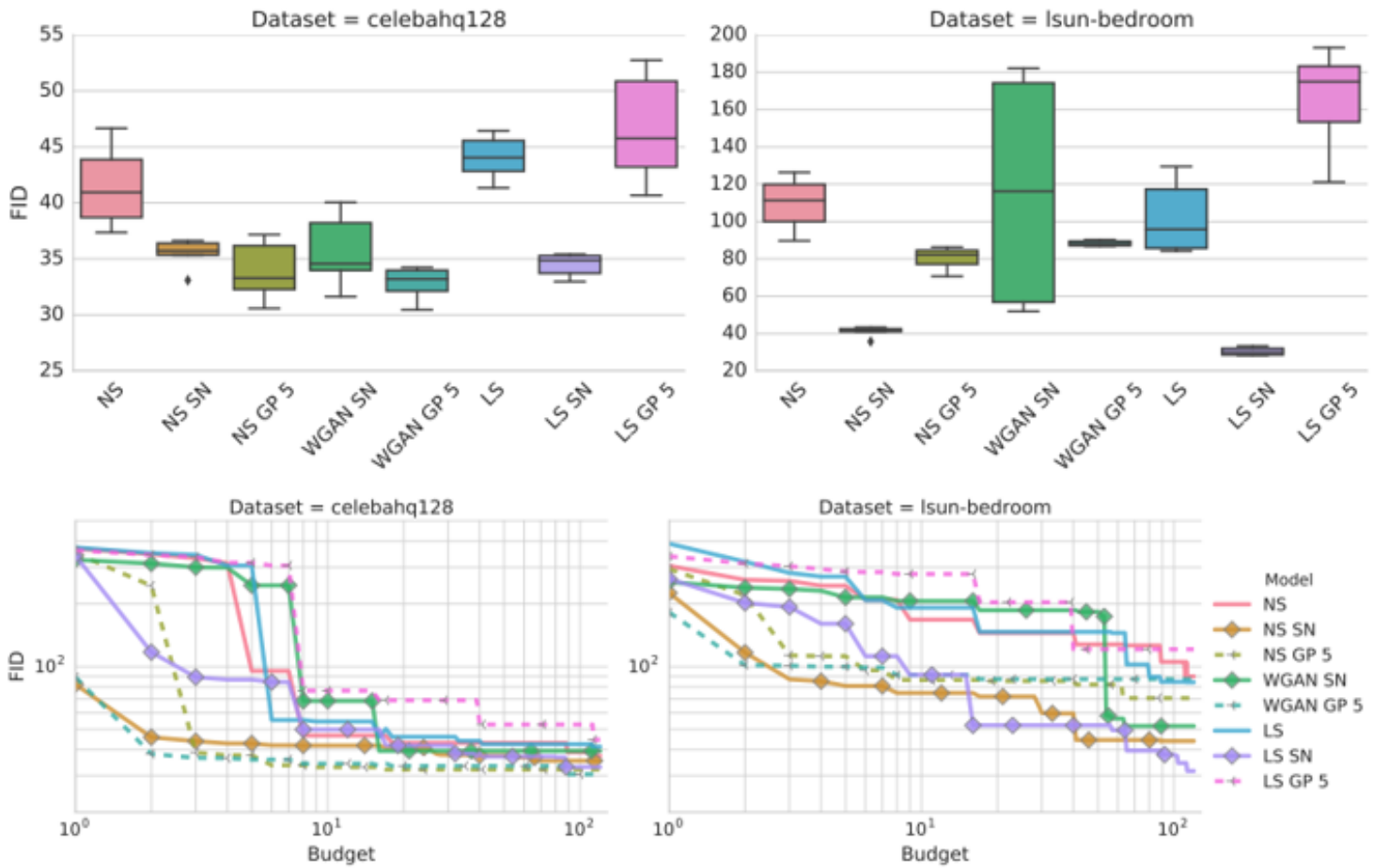


Figure: top 5% model损失函数的影响

损失函数(结论)

- 谱归一化提高了两个数据集的模型质量
- 梯度惩罚也会对模型的性能提升有所帮助
- 原始GAN的非饱和损失跟其它改进的损失结果差距不大

网络结构

考虑上述的结果是否也适用于不同的网络结构，为此还对SNDCGAN进行了研究。主要考虑下面几项：

- 非饱和GAN损失
- 梯度惩罚
- 谱归一化

网络结构的消融研究

作者注意到与目前github的实现相比， Resnet架构有六个细微差别。进行了消融研究以验证这些差异的影响。详细描述如下：

1. 默认情况：ResNet CIFAR架构，具有谱归一化和非饱和GAN损失
2. 残差连接：使用输入作为判别器ResBlock中残差连接的输出。 默认情况下，它是一个带有3x3内核的卷积层。
3. CIN: 使用c_i作为判别器ResBlock隐藏层输出通道数目。默认情况下设置为c_o, 但是MiyatoMiyatoet al. (2018)使用c_o作为第一层的输出通道数，其余的通道数设置为c_i

网络结构的消融研究

4 OPT: 判别器的第一个残差块设置一些优化，包括： （1）没有Relu激活 （2）一个包含残差连接的卷积层 （3）在残差块儿内部，用c_o替代c_i.

5 CIN OPT： 将CIN和OPT的设置结合

6 判别器输出层使用reduce sum操作, 默认使用reduce mean

结果如下图所示。

网络结构的消融研究

7 TAN: 使用Tanh作为生成器输出层激活，判别器输入值的范围为[-1, 1]。默认使用sigmoid激活，判别器输入范围为[0, 1]

8 EPS: 生成器中BN层使用较大的epsilon：2e-5。TensorFlow默认为1e-5

9 ALL：将上述的设置综合使用

网络结构的消融研究

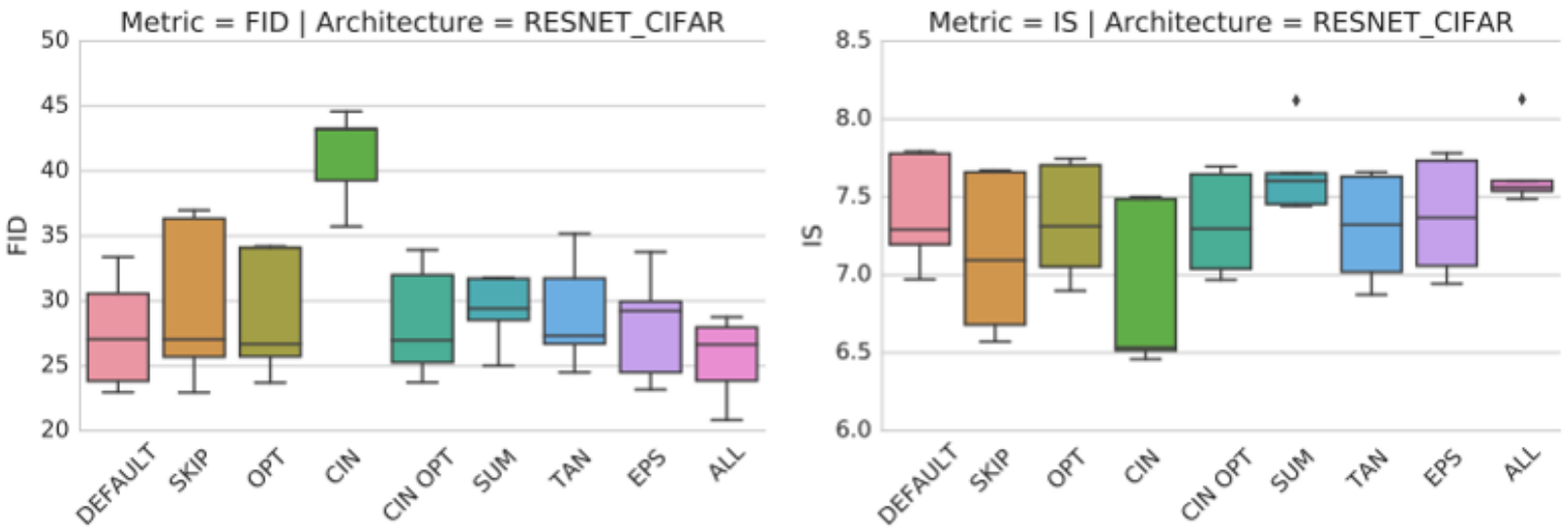


Figure 7. Ablation study of ResNet architecture differences. The experiment codes are described in Section D.

Figure: ResNet结构消融结果

CIN的设置得到的分数最差.当CIN和OPT结合效果有所提升(与其它结果相当), 根据对CIFAR10结果研究，这些差异的影响很小。

一些建议

实验中比较普遍的结果是，判别器的Lipschitzc常数对模型的性能至关重要,同时进行正则化和归一化可以提高模型的质量。为了量化这种效果，将损失固定为非饱和损失，使用Resnet19体系结构,结合几种标准化和正则化方案,使用上面表中所示的超参数设置和随机选择的24个参数。

结果

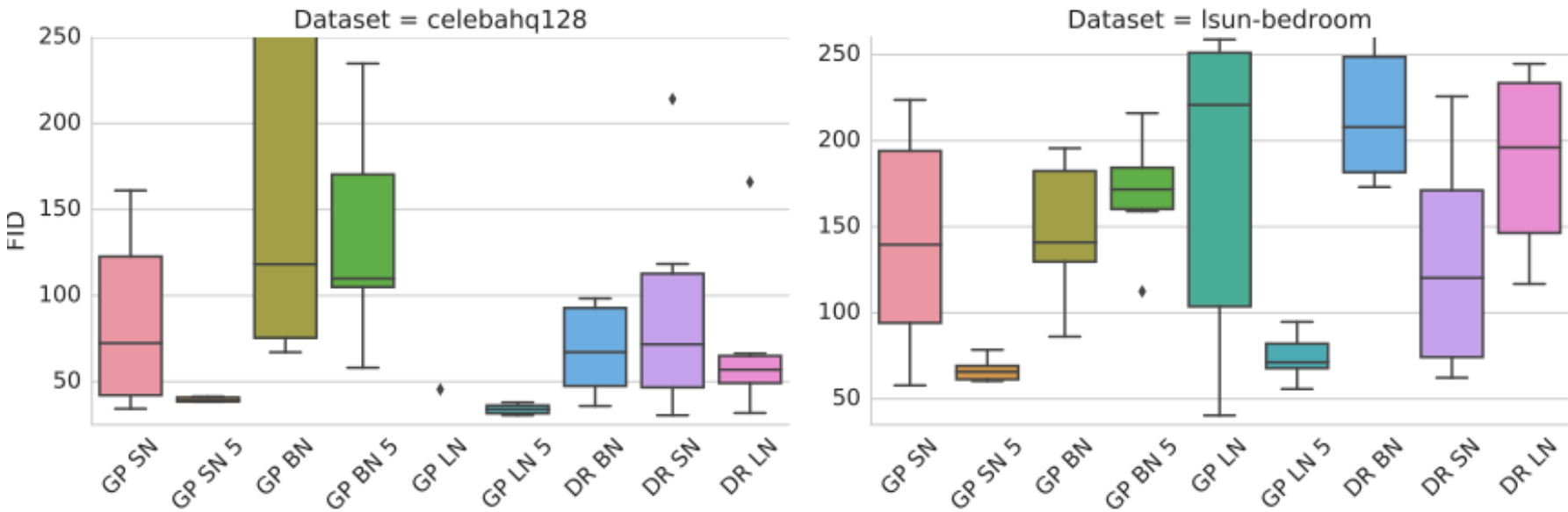


Figure: top 5% model 正则化和归一化结合使用

可以看出梯度惩罚和SN或LN结合能够很好的提升模型的性能.

问题1（不懂）

产生上述结果的原因： Gradient penalty coupled with spectral normaliza- tion (SN) or layer normalization (LN) strongly improves the performance over the baseline. This can be partially explained by the fact that SN doesn’t ensure that the discriminator is 1-Lipschitz due to the way convolutional layers are normalized.

实验的一些细节

- 应该根据测试数据集计算FID
- 网络结构的详细设计
- 数据集的预处理（如放大、裁剪的精确算法）

目前存在的一些问题（非确定性）

- 论文中提出的算法与开源的代码实现不匹配
- 良好的算法实现与不良的代码实现之间有着巨大差距
- 训练的随机性，即训练相同的模型两次能否获得相同的分数是十分重要的， 由于某些GPU操作存在随机性，如果禁用这些操作就会造成时间的损失

归一化公式

- 批量归一化

$$\hat{x^{(k)}} = \frac{x^{(k)} - E[x^{(k)}]}{\sqrt{Var[x^{(k)}]}}$$

$$y^{(k)} = \gamma^{(k)}\hat{x}^{(k)} + \beta^{(k)}$$

- 谱归一化: 可以使网络满足利普西茨约束
- 求某一层权值矩阵W的最大特征值$\sigma(W)$

$$\sigma(W) \simeq \hat{u}W^T\hat{v}$$

$$W \leftarrow \frac{W}{\sigma(W)}$$

其中：$\hat{u}$和$\hat{v}$对应矩阵W的特征分解矩阵

正则化公式

- 梯度裁剪
  - 设定裁剪阈值c，损失对参数W的梯度为g， 如果$\|g\|_2^2 > c$:

$$g = \frac{c}{\|g\|_2} \cdot g$$

否则g保持不变

- L2正则化

$$loss + \|\theta\|_2$$