



南开大学  
Nankai University

南 开 大 学

计 算 机 学 院

计算机图形学

---

Jittor:Conditional GAN

---

石家琪 2011739

年级：2020 级

专业：计算机科学与技术

2023 年 5 月 9 日

# 目录

一、 实验介绍	1
二、 网络结构	1
三、 实验环境	2
四、 实验结果	2
(一) 超参数设置 . . . . .	2
(二) 结果展示 . . . . .	4
(三) Jittor vs pytorch . . . . .	4
(四) 最终生成图片 . . . . .	4
五、 头歌平台评测结果	5
六、 评测编号	5
七、 Gitlink 开源地址	5
八、 Gitlink 主页	5

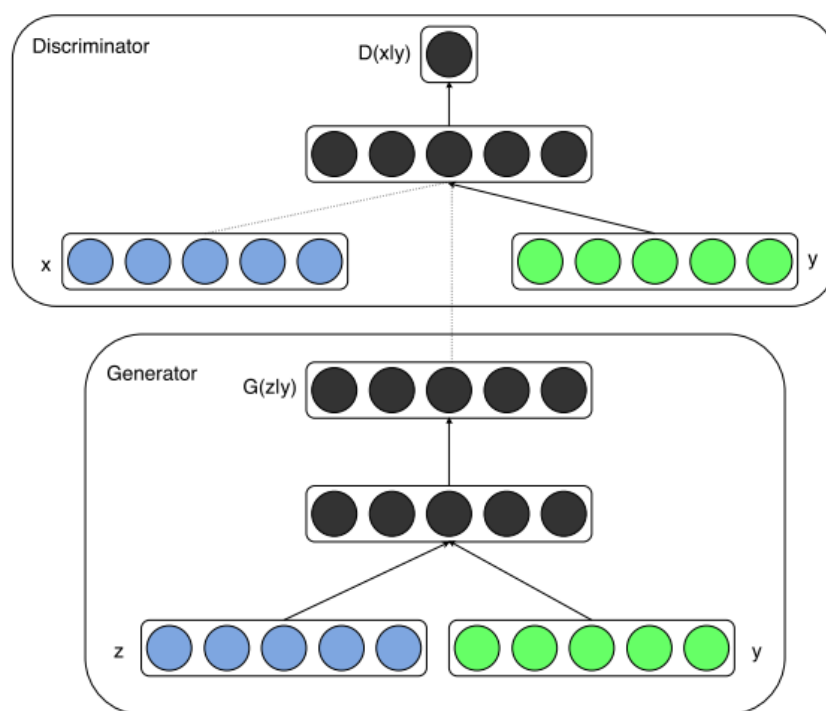


图 1: Conditional adversarial net

## 一、 实验介绍

本实验使用清华大学开源代码框架 Jittor, 利用主流的手写数字数据集 MNIST 实现了 Conditional GAN 网络, 目标是用计算机生成手写数字。并且横向对比 pytorch 版本下的 Conditional GAN 的运行速度, 测试 Jittor 框架的执行效率。

## 二、 网络结构

本实验依据论文 [2], 构建 G 和 D 两个网络, 并且在两个网络的输入端增加条件  $y$ 。由于该论文并未提供各网络内部的架构, 因此我参考PyTorch-GAN该 pytorch 框架下实现的 Conditional GAN 网络的架构, 复现以 jittor 为框架的 Conditional GAN 网络。

为了了解该网络的架构设计细节, 我通过查找相关资料Tips and tricks to make GANs work, 发现以下可以使 GAN 网络高效工作的架构设计技巧, 并且复现的 Conditional GAN 网络也采用了这些技巧。

- 正则化输入: 将图像正则化到  $[-1,1]$  之间, 用  $\text{Tanh}()$  作为 Generator 网络的最后一层。
- 修改损失函数: 在很多 GAN 的论文里, 优化 G 的损失函数是  $\min(\log(1-D))$ , 然而在训练中人们通常用  $\max(\log(D))$ , 因为第一个公式在早期存在梯度消失现象。
- 避免稀疏梯度: 如果有稀疏的梯度, GAN 的稳定性就会受到影响。可以使用 LeakyReLU 激活函数, 实践表明, 该激活函数在 G 和 D 网络中均表现很好。
- 使用 adam 优化器: 参考文献 [?]

- 离散变量: 使用 embedding 层, 作为附加通道添加到图像。
- dropout: 在训练和测试阶段的 G 中使用 dropout, 以 dropout (50%) 形式提供噪音 [1]

### 三、 实验环境

本机硬件设备配置:

- 设备名称:LAPTOP-MS4Q4BJC
- 处理器: Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11 GHz
- 机带: RAM 16.0 GB (15.8 GB 可用)

运行环境:

- Windows 10
- PyCharm Community Edition 2022.2.2
- Python 3.8

环境依赖:

- jittor 1.3.7.13
- numpy 1.24.2
- Pillow 9.5.0

### 四、 实验结果

#### (一) 超参数设置

为了获取较好的结果, 超参数的设置十分关键。因此本实验采用PyTorch-GAN的超参数设置。

```
1 args={'n_classes':100,
2       'latent_dim':100,
3       'img_size':32,
4       'channels':1,
5       'b1':0.5,
6       'b2':0.999,
7       'lr':0.0002,
8       'batch_size':64,
9       'n_epochs':50,
10      'n_cpu':8,
11      'sample_interval':1000,
12      'dropout':0.4,
13      'relu':0.2}
```

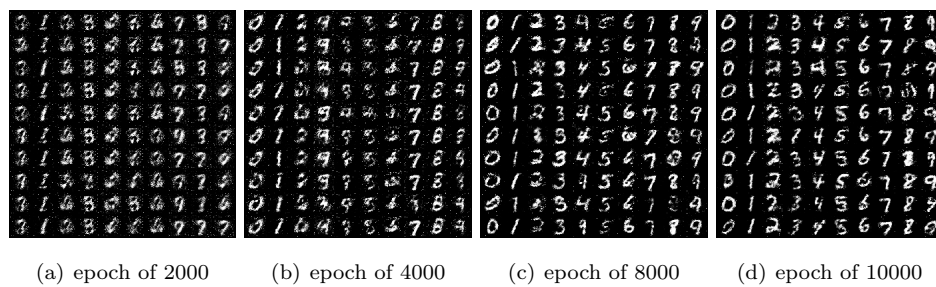


图 2: 生成手写数字

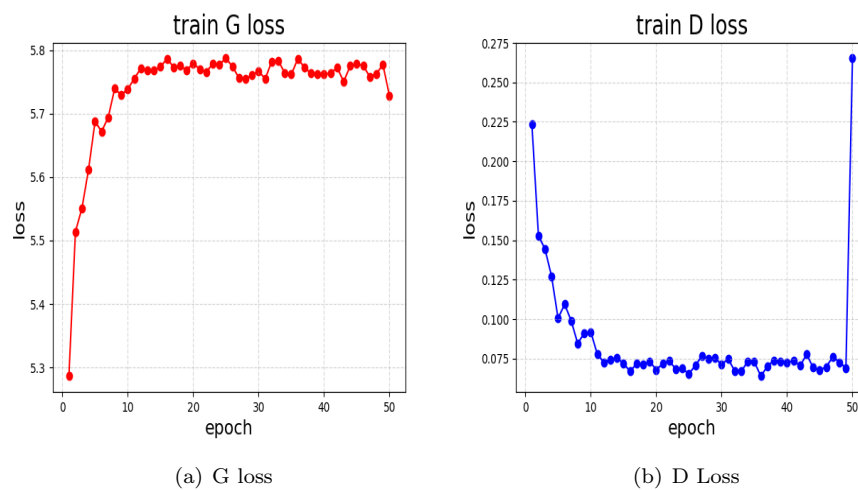


图 3: G、D loss of train

## (二) 结果展示

从图2可以观察到 G 网络生成手写数字效果的迭代过程，由模糊变为清晰真实。

如图3所示，将训练阶段的 G 网络和 D 网络的损失进行输出，可以观察到 D 和 G 网络均能够在较短训练周期内收敛到较好的状态，之后在一定浮动内保持稳定。但能够观察到 D 网络在最后一个周期时出现异常浮动，推测其原因可能与 model 设计中采用了 dropout 有关。

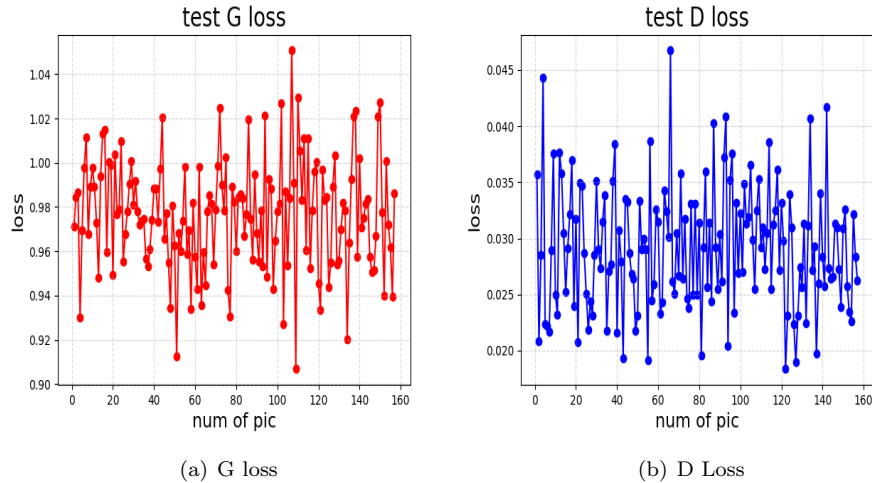


图 4: G、D loss of test

如图4所示，观察到 D 网络的损失值在 0.02 到 0.45 之间浮动，大部分集中在 0.025 到 0.035 的区间中。和训练时的 D 网络损失 0.075 相比较，甚至获得了更好的性能（不太理解）。而 G 网络的损失值则大部分集中在 1 附近，与测试时的后期稳定数据相比，要远远小于。可以看到，针对测试数据，训练出来的 D 网络具有更加精确的判断能力，而 G 网络则由于 D 网路判断能力的提高，而相对地减弱了伪造手写字体的能力。这也能够体现出生成对抗式网络的特征，相信在更加充足的训练周期下，G 网络会获得更高的伪造手写字体的能力。

## (三) Jittor vs pytorch

在相同训练集、超参数、网络架构、训练批次及次数下，Jittor 的训练时间在 2 小时左右，而 pytorch 的训练时间在 3.5-4 小时左右。

## (四) 最终生成图片



图 5: result

## 五、 头歌平台评测结果

2023050919221947345537	result.zip...	石家琪	2023-05-09 19:22:19	完成	0.871	是否通过: 1	下载
------------------------	---------------	-----	---------------------	----	-------	---------	----

图 6: 评测结果

## 六、 评测编号

2023050919221947345537

## 七、 Gitlink 开源地址

[https://www.gitlink.org.cn/civilizwa/CGAN\\_Jittor/tree/master](https://www.gitlink.org.cn/civilizwa/CGAN_Jittor/tree/master)

## 八、 Gitlink 主页

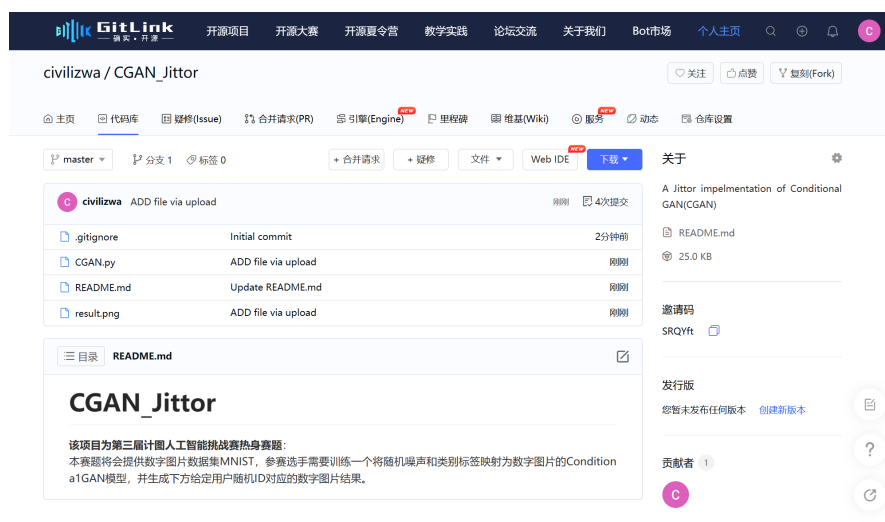


图 7: 项目主页

## 参考文献

- [1] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.
- [2] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.