

图形学实验 PA4: Conditional GAN

指导教师：胡事民 助教：方晓楠

2021 年 5 月 18 日

1 实验综述

本次作业中，你将使用 Jittor 机器学习框架，在数字图片数据集 MNIST 上训练一个将随机噪声和类别标签映射为数字图片的 Conditional GAN 模型，生成自己的学号对应的图片。

2 Conditionanl GAN 网络架构

Generative Adversarial Nets (GAN) 提出了一种新的方法来训练生成模型：输入为一个随机向量 z ，生成器 G 输出一幅图像 $G(z)$ ，而判别器 D 需要将真实图像 x 与合成图像 $G(z)$ 区分开来。然而，GAN 对于要生成的图片缺少控制。Conditional GAN (CGAN) 通过添加显式的条件或标签，来控制生成的图像。在生成器 generator 和判别器 discriminator 中添加相同的额外信息 y ，GAN 就可以扩展为一个 conditional 模型。 y 可以是任何形式的辅助信息，例如类别标签或者其他形式的辅助信息。我们可以通过将 y 作为额外输入层，添加到生成器和判别器来完成条件控制。

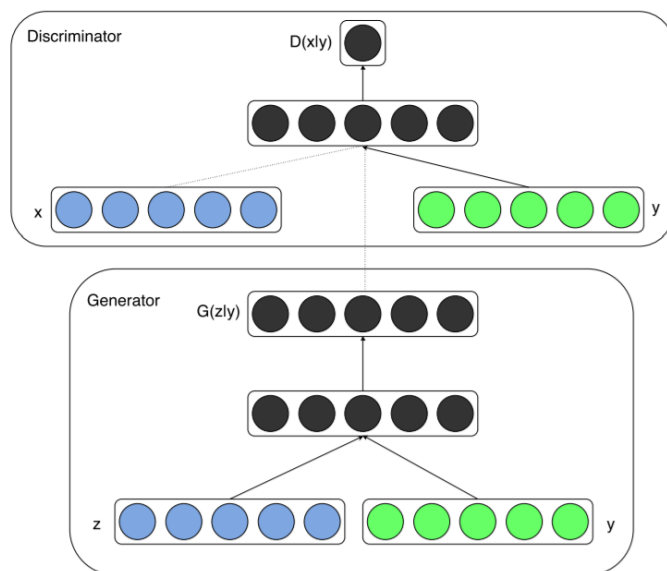


图 1: CGAN 模型示意

GAN 模型的损失函数设计为：

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (1)$$

对于判别器 D ，我们要训练最大化这个 loss。如果 D 的输入是来自真实样本的数据 x ，则 D 的输出 $D(x)$ 要尽可能地大， $\log(D(x))$ 也会尽可能大。如果 D 的输入是来自 G 生成的假图片 $G(z)$ ，则 D 的输出 $D(G(z))$ 应尽可能地小，从而 $\log(1-D(G(z)))$ 会尽可能地大。这样可以达到 $\max D$ 的目的。

对于生成器 G ，我们要训练最小化这个 loss。对于 G 生成的假图片 $G(z)$ ，我们希望能尽可能地骗过 D ，让它觉得我们生成的图片就是真的图片，这样就达到了 G “以假乱真”的目的。那么 D 的输出 $D(G(z))$ 应尽可能地大，从而 $\log(1-D(G(z)))$ 会尽可能地小。这样可以达到 $\min G$ 的目的。

D 和 G 以这样的方式联合训练，最终达到 G 的生成能力越来越强， D 的判别能力越来越强的目的。

在 CGAN 中，我们增加了限定条件 y ，即数字 0-9 的类别标签，因此生成器和判别器的输入都需要增加类别标签的维度，若真实图片为 x ，对应标签为 y_1 ，随机向量为 z ，随机标签为 y_2 ，则生成器的输出为 $G(z, y_2)$ ，判别器的输出为 $D(G(z, y_2), y_2)$ 及 $D(x, y_1)$ 。在本次作业中，我们采用平方误差函数替代对数函数来计算损失。记合成图片为第 0 类，真实图片为第 1 类，则分类器的损失函数为：

$$L_D = \frac{1}{2} ((D(G(z, y_2), y_2))^2 + (1 - D(x, y_1))^2) \quad (2)$$

生成器的目标则是希望合成图片能欺骗判别器，使其被分为第 1 类，因此生成器的损失函数为：

$$L_G = (1 - D(G(z, y_2), y_2))^2 \quad (3)$$

3 代码说明

3.1 Jittor 安装

Jittor 框架目前支持 Linux 或 Windows Subsystem of Linux (WSL)，需要使用 Python 及 C++ 编译器 (g++ 或 clang)。Jittor 提供了三种安装方法：docker，pip 和手动安装，具体安装教程请参考：

<https://cg.cs.tsinghua.edu.cn/jittor/download/>

3.2 代码框架

本次代码仅包含一个文件 PA4_CGAN.py，我们实现了大部分的模型定义、训练和推理的代码，在少量需要补充代码的位置加上了 TODO 标记。生成器 Generator 和判别器 Discriminator 中的 init 函数用于定义模型架构，execute 函数给定网络输入返回网络输出。模型中主要使用的模块有

- nn.Embedding(num, dim): 用于将 num 类整数标签转换为 dim 维向量
- nn.Linear(in_features, out_features): 全连接层，输入向量维度 in_features，输出向量维度 out_features
- nn.Dropout(p): 将比例为 p 的特征置为 0
- nn.LeakyReLU(scale): ReLU 函数的变种，输入为负值时输出乘以 scale

因为图像的尺度较小，我们直接使用了全连接层而不是通常的卷积层。

代码中已经为你定义好了优化器（optimizer），并会自动下载 MNIST 数据集。每轮迭代中，我们枚举数据集中的图片（imgs）和类别标签（labels）对，并随机生成一组输入向量，按照公式(3)和公式(2)分别为计算生成器和判别器损失函数，回传梯度并更新网络参数。每迭代若干轮会随机采样生成一批数字图片，下图是一个示例。

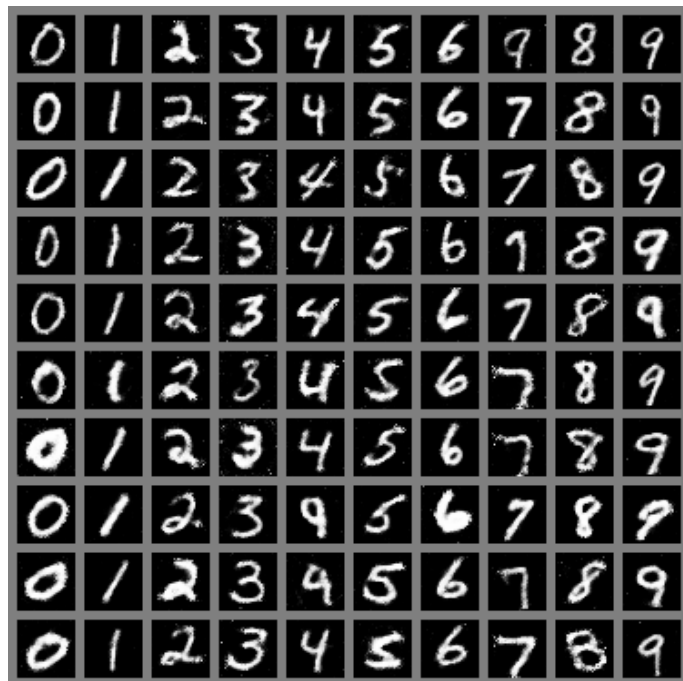


图 2: 中间结果示例

模型训练完毕后，我们给定一组学号作为输入的数字标签，将模型生成的图片保存至 result.png，结果应如下所示，其中的数字需要修改为你的学号。



图 3: 最终结果示例

4 提交要求

请将你的代码（PA4_CGAN.py），模型文件（generator_last.pkl）以及最终生成的学号图片（result.png）放入 code 文件夹，打包成 zip 文件提交至网络学堂。

本次作业的 Deadline 以网络学堂为准。迟交的同学将得到一定的惩罚：晚交 3 天内分数将降低为 80%，3 天以上 1 周以内分数降为 50%，迟交一周以上的同学分数为 0。