《人工智能导论》大作业

任务名称：Mnist条件生成器

完成组号：

小组人员：徐恒敬、闫家硕、胡馨月、王宇凡、王浩宇

完成时间：2023年6月17日

1．任务目标

基于 Mnist 数据集，构建一个条件生成模型，输入的条件为 0-9 的数字，输出对应条件的生成图像。

2．具体内容

（1）实施方案

生成模型：cDCGAN（条件深度卷积生成对抗网络）

由于原始 GAN 具有生成的图像随机、不可预测，无法控制网络输出特定的图片，生成目标不明确，可控性不强等缺点，本小组在讨论后决定使用 cDCGAN 模型。

原理： cDCGAN 又称为条件 GAN，可以根据给定的条件生成特定类别图像，原理为：在中间层 concat 上 label，相当于加入条件。原本的 GAN 可以看成噪声到目标的映射关系，比如 G(noise) = fake_img, D(img) = p_of_real， 加入条件后变为 G(noise | y=label) = (fake_img | y=label), D(img | label) = p_of_real | y = label。

主要步骤：

1、模型及优化器初始化：判别器、生成器模型 D、G 及其优化器 D_opt、G_opt，损失函数 criterion。

```
D = Discriminator().to(DEVICE)
G = Generator().to(DEVICE)

criterion = nn.BCELoss()
```

```
D_opt = torch.optim.Adam(D.parameters(), lr=0.0005, betas=(0.5, 0.999))
G_opt = torch.optim.Adam(G.parameters(), lr=0.0005, betas=(0.5, 0.999))
```

2、训练循环

3、保存训练结果：每 1000 步保存生成器生成的样本图像

```
if step % 1000 == 0:
G.eval()
img = get_sample_image(G, n_noise)
imsave('../samples/{}_step{}.jpg'.format(MODEL_NAME,
str(step).zfill(3)), img, cmap='gray')
G.train()
```

（2）核心代码分析

cDCGAN 是一种基于 DCGAN（深度卷积生成对抗网络），在生成过程中引入条件信息的变种，“c”即 Conditional 条件。

```
class Discriminator(nn.Module):
    """
        Convolutional Discriminator for MNIST
    """
    def __init__(self, in_channel=1, input_size=784, condition_size=10,
num_classes=1):
        super(Discriminator, self).__init__()
        self.transform = nn.Sequential(
            nn.Linear(input_size+condition_size, 784),
            nn.LeakyReLU(0.2),
        )
        self.conv = nn.Sequential(
            # 28 -> 14
            nn.Conv2d(in_channel, 512, 3, stride=2, padding=1,
bias=False),
            nn.BatchNorm2d(512),
            nn.LeakyReLU(0.2),
            # 14 -> 7
            nn.Conv2d(512, 256, 3, stride=2, padding=1, bias=False),
            nn.BatchNorm2d(256),
            nn.LeakyReLU(0.2),
            # 7 -> 4
            nn.Conv2d(256, 128, 3, stride=2, padding=1, bias=False),
            nn.BatchNorm2d(128),
            nn.LeakyReLU(0.2),
            nn.AvgPool2d(4),
```

```
        )
        self.fc = nn.Sequential(
            # reshape input, 128 -> 1
            nn.Linear(128, 1),
            nn.Sigmoid(),
        )

    def forward(self, x, c=None):
        # x: (N, 1, 28, 28), c: (N, 10)
        x, c = x.view(x.size(0), -1), c.float() # may not need
        v = torch.cat((x, c), 1) # v: (N, 794)
        y_ = self.transform(v) # (N, 784)
        y_ = y_.view(y_.shape[0], 1, 28, 28) # (N, 1, 28, 28)
        y_ = self.conv(y_)
        y_ = y_.view(y_.size(0), -1)
        y_ = self.fc(y_)
        return y_
```

该算法判别器网络接受真实图像和生成图像及分别对应的条件标签作为输入，通过（三层）卷积层将图像转换为特征表示，并输出表示输入图像是真实图像的概率，用以辨别真实图像和生成图像，辅助生成图像接近真实图像，与生成器形成对抗进步。在该任务中，判别器起到了筛选生成图像和提供反馈信号的作用。通过生成器和判别器之间的对抗学习，cDCGAN 能够逐步生成更加逼真的手写数字图像。

```
class Generator(nn.Module):
    """
        Convolutional Generator for MNIST
    """
    def __init__(self, input_size=100, condition_size=10):
        super(Generator, self).__init__()
        self.fc = nn.Sequential(
            nn.Linear(input_size+condition_size, 4*4*512),
            nn.ReLU(),
        )
        self.conv = nn.Sequential(
```

```
            # input: 4 by 4, output: 7 by 7
            nn.ConvTranspose2d(512, 256, 3, stride=2, padding=1,
bias=False),
            nn.BatchNorm2d(256),
            nn.ReLU(),
            # input: 7 by 7, output: 14 by 14
            nn.ConvTranspose2d(256, 128, 4, stride=2, padding=1,
bias=False),
            nn.BatchNorm2d(128),
            nn.ReLU(),
            # input: 14 by 14, output: 28 by 28
            nn.ConvTranspose2d(128, 1, 4, stride=2, padding=1,
bias=False),
            nn.Tanh(),
        )

    def forward(self, x, c):
        # x: (N, 100), c: (N, 10)
        x, c = x.view(x.size(0), -1), c.float() # may not need
        v = torch.cat((x, c), 1) # v: (N, 110)
        y_ = self.fc(v)
        y_ = y_.view(y_.size(0), 512, 4, 4)
        y_ = self.conv(y_) # (N, 28, 28)
        return y_
```

该算法生成器网络接受一个随机噪声向量和条件标签作为输入，通过训练生成接近真实图像的生成图像，并不断调整。且该生成器支持条件生成，根据输入的条件信息生成特定类别的生成图像。训练过程中生成器与判别器形成对抗，目标为生成更真实的生成图像以欺骗判别器。在该任务中，生成器负责生成接近真实手写数字图像的生成数字图像，并与判别器进行对抗学习，以不断提升生成质量。生成器的作用是实现从随机噪声和条件输入到逼真手写数字的转换，并通过学习生成真实数据分布的样本。

```
criterion = nn.BCELoss()
```

```
#判别器训练部分
x_outputs = D(x, y)
D_x_loss = criterion(x_outputs, D_labels)

z_outputs = D(G(z, y), y)
D_z_loss = criterion(z_outputs, D_fakes)
D_loss = D_x_loss + D_z_loss

#生成器训练部分
z_outputs = D(G(z, y), y)
G_loss = criterion(z_outputs, D_labels)
```

训练循环中根据判别器的输出生成损失（loss），具体计算步骤为：在训练循环的判别器训练部分，通过将真实图像和对应的标签传递给判别器，得到判别器对真实图像的输出结果 x_outputs。然后，将真实图像的输出结果与判别器标签 D_labels（全为 1）之间计算二进制交叉熵损失（即使用损失函数 criterion），得到 D_x_loss。

类似地，在训练循环的生成器训练部分，通过将生成的图像和对应的标签传递给判别器，得到判别器对生成图像的输出结果 z_outputs。然后，将生成图像的输出结果与判别器标签 D_labels（全为 1）之间计算二进制交叉熵损失，得到 G_loss。

使用损失函数将判别器的输出结果与目标标签进行比较，计算判别器的损失 D_loss 和生成器的损失 G_loss。

以下为判别器与生成器的损失对比图：

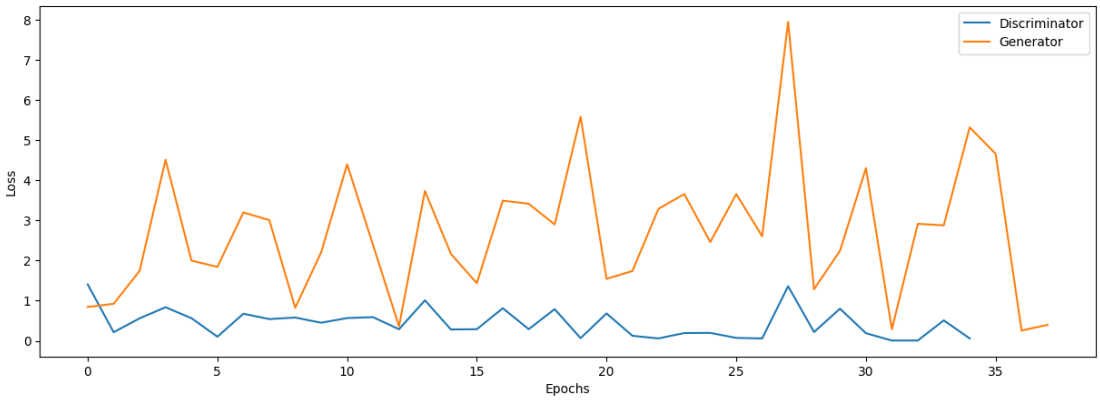

图 1 Discriminator 和 Generator 的 loss 对比图

以下为部分生成图：

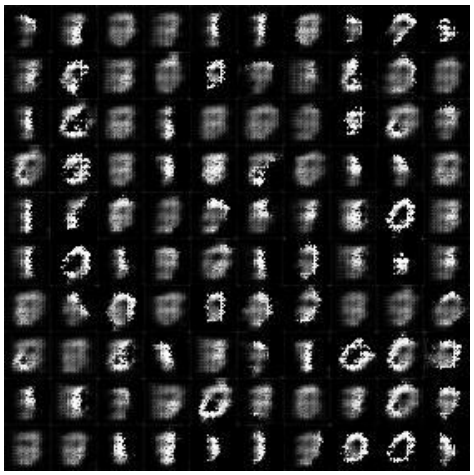

图 2　1000 步生成图

图 3　10000 步生成图

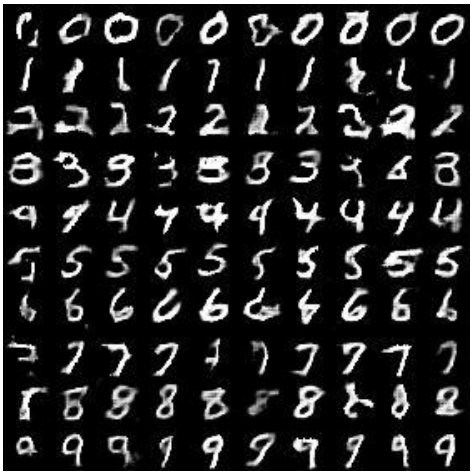

图 4　28000 步生成图

3．工作总结

（1）收获、心得

对于 cGAN、DCGAN、cDCGAN 等 GAN 中的一些变种模型有了基本的认识与了解，对人工智能项目的实现有了基本的了解。项目完成过程中，小组成员密切合作和交流，相互学习，取得了进步。

（2）遇到问题及解决思路

在模型选择过程中，我们小组先后考虑了 infoGAN、DCGAN 等模型，最终选择了 cDCGAN 模型。经过讨论，cDCGAN 模型相对 infoGAN 模型更简单直接，相对 DCGAN 模型可控性更强，故采用。

4．课程建议

人工智能导论课程的课堂教学以对着 PPT 讲解为主，辅以少量的课后作业或小问答，并最终以大作业作为期末考核。这样的平时教学缺少对代码的实际操作，学生对人工智能代码层面实现的认识较浅，对大作业的完成帮助也较小。希望课程增多平时教学中接触代码的机

会，如从易到难的平时代码作业等（以了解为主），这也有助于最终大作业的完成。

5. 组内分工

模型训练：闫家硕、徐恒敬

接口：胡馨月

报告撰写：王宇凡、王浩宇