

《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： 陈洋、丁华威、符景乐、谷韞名、苏世桢

完成时间：

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

主要分为3步

1. `test=AiGcMn("generator_params.pth","discriminator_params.pth")`

传入两个str类型的参数（是预先用infogan训练好并保存的模型参数）

2. `test.misc()`

加载模型参数和其他中间过程的处理

3. `output_tensor=test.generate(target=input)`

用generate函数接收一个整数型n维tensor（n是batch的大小，每个整数在0~9范围内，代表需要生成的数字），输出是 `n*1*28*28` 的tensor（n是batch的大小，每个 `1*28*28` 的tensor表示随机生成的数字图像）

（2）核心代码分析

infogan的大致思路如下：

在生成器网络中引入一些噪声，然后通过信息理论的方法来最大化潜在变量的互信息（mutual information）。互信息可以用来衡量两个随机变量之间的依赖关系，即它们能够提供多少关于彼此的信息。在InfoGAN中，生成器网络的输入被分成两部分：噪声和潜在变量。噪声是随机的，而潜在变量则是可以被控制的，并且可以描述数据的某些特征。生成器网络的任务是将这些输入转换为逼真的样本。

通过引入一个额外的信息理论损失函数，即最大化潜在变量和生成器网络输出之间的互信息，InfoGAN试图学习到一些有意义的潜在变量。这些潜在变量可以描述数据的一些特征，例如数字的旋转角度或手写数字的粗细等。通过最大化互信息，生成器网络不仅可以生成逼真的样本，还可以控制样本的某些属性，从而使生成的样本更加多样化和有趣。

InfoGAN的训练过程包括两个阶段：首先，使用标准的GAN方法训练生成器和鉴别器网络来生成逼真的样本。然后，在这个基础上，再引入信息理论损失函数，并使用反向传播算法来最大化互信息。这个过程需要花费一定的时间来优化生成器和鉴别器网络，并找到最佳的潜在变量，以产生高质量的、有意义的样本。

总之，InfoGAN的大致思路是通过在生成器网络中引入潜在变量，并使用信息理论方法来最大化潜在变量和生成器网络输出之间的互信息，从而使生成器网络不仅能够生成逼真的样本，还能够学习到数据的一些特征和属性，从而实现更加多样化和有趣的样本生成。

aigcmn的具体实现：

讲解其中的generate函数

```
def generate(self, target):

    if isinstance(target, torch.Tensor):
        target_tensor=target
    else:
        target_tensor=torch.tensor(target)

    # 定义映射关系字典
    mapping = {
        0: 0,
        1: 1,
        2: 2,
        3: 7,
        4: 8,
        5: 4,
        6: 5,
        7: 6,
        8: 3,
        9: 9
    }

    # 创建索引张量
    input_tensor= torch.tensor([mapping[i.item()] for i in target_tensor])
    #input_tensor=target_tensor

    result = torch.empty(0)
    for i in input_tensor:
        result = torch.cat([result, self.sample2[i+30].unsqueeze(0)], dim=0)

    #print("输出tensor: ",result.size())

    # 假设生成的图像为 gen_img, 类型为 torch.Tensor
    resized_result = torch.nn.functional.interpolate(result, size=(28, 28),
mode='bilinear', align_corners=True)
    #print("输出修改后的tensor: ", resized_result.size())
    save_image(resized_result, "AiGcmn_test_0.png",
nrow=resized_result.size(0), normalize=True)
    return resized_result
```

首先，判断 target 是否已经是 torch.Tensor 类型，如果是，则直接使用 target，否则将其转换为 torch.Tensor 类型，赋值给 target_tensor 变量。

接下来，定义了一个映射关系的字典 mapping，将目标张量中的每个数字映射到一个新的数字，从而得到一个新的索引张量 input_tensor。具体地，字典中的键表示目标张量中的数字，值表示映射后的数字。例如，键为 0，值为 0，表示将目标张量中的数字 0 映射为新的数字 0。

然后，使用 `input_tensor` 中的每个索引值作为下标，从模型的 `sample2` 属性中获取对应的随机噪声张量，并将这些张量拼接成一个新的张量 `result`。

接下来，使用 `torch.nn.functional.interpolate` 函数将 `result` 张量中的每个噪声张量插值到指定的大小（这里是 `28x28`），从而得到一个大小与 MNIST 数据集相同的张量 `resized_result`。插值的方式使用双线性插值（`bilinear`）。

最后，将 `resized_result` 保存为一个 PNG 图像文件，并返回 `resized_result` 张量作为生成的图像。保存图像使用了 PyTorch 中的 `save_image` 函数，它可以将一个张量保存为图像文件。`nrow` 参数指定每行显示的图像数量，`normalize` 参数指定是否对张量进行归一化处理。在这段代码中，`normalize` 参数被设置为 `True`，表示将张量的值归一化到 `[0, 1]` 区间内。

3. 工作总结

(1) 收获、心得

对于gan中一些基本的模型（如cgan、infogan等）有了基本的认识与了解，在实验过程中对于结果的分析让我对这部分的知识有了更深一步的掌握，同时我对于分组合作有了崭新的见解。我们小组每个人充分发挥每个人的优势和专长，互相学习和支持，共同完成项目的目标。

(2) 遇到问题及解决思路

问题：模型的参数有batch行，但我们只要求输出一行，我们小组关于输出哪一行展开了讨论

解决思路：我们经过大量统计后选择了平均输出质量最高的一行

4. 课程建议

人工智能导论课程应该注重实践和应用能力的培养。作为一个实践性很强的领域，人工智能需要我们具备一定的编程和实验能力。因此，建议在课程中加强编程实践，引导学生独立设计和实现 AI 模型，并提供一些案例和指导，以便更好地锻炼学生的实践和应用能力。