大家好， 我是Leo，今天我们将使用PyTorch对Fashion MNIST数据集进行条件生成对抗网络Conditional GAN训练。

条件生成对抗网络（Conditional GAN）是一种基于生成对抗网络（GAN）的模型，它在生成图像的过程中不仅考虑了噪声向量的影响，还引入了额外的条件信息，比如图像的标签或者一些属性值。这种额外的条件信息可以帮助模型生成具有特定属性的图像，比如给定“T恤”这个标签，模型可以生成一张具有“T恤”属性的图像。

与传统的GAN相比，条件GAN多了一个输入条件，通常是一个向量或一个标签。这个条件输入在生成器和判别器中都被使用，以帮助模型更好地生成符合条件的图像。在训练过程中，条件GAN的生成器和判别器通过博弈的方式相互训练，以使生成器逐渐生成更加逼真的图像，并让判别器更加难以区分真实和虚假的图像。

条件GAN可以被广泛地应用于各种图像生成任务，比如图像修复、图像翻译、图像生成等。

接下来, 我们搭建一个简单的条件生成对抗网络来了解更多。

首先，我们导入所需的libraries。

我们定义Conditional GAN的超参数：学习率、批大小、迭代轮数、隐向量的大小、图像大小。

接下来,我们转换Fashion MNIST数据集。转换将图像转换为PyTorch张量，并将像素值归一化为范围[-1, 1]。

然后，我们加载Fashion MNIST数据集并为训练集创建一个PyTorch数据加载器。

让我们定义生成器网络。在条件GAN中，生成器不仅接收一个随机噪声向量 **noise** 作为输入，还接收一个表示图像类别的标签 **label**。在生成图像时，生成器会将噪声向量 **noise** 和标签 **label** 结合在一起，通过一系列的线性层操作，生成一张与标签 **label** 相关的虚假图像。

在上面的代码中，我们定义了一个名为 **Generator** 的类，它继承自 **nn.Module**。在类的初始化函数中，我们定义了生成器的结构，包括线性层、和一个 Tanh 激活函数。在 **forward** 函数中，我们将输入的噪声向量 **noise** 和标签 **label** 拼接在一起，并通过一系列的线性层操作将其转化为一张大小为 28x28 的虚假图像。最后，我们将图像的像素值压缩到 [-1, 1] 的范围内，并返回生成的虚假图像。

而对于判别器网络。在条件GAN中，判别器不仅接收一个图像作为输入，还接收一个表示图像类别的标签 **label**。在判别真实和虚假图像时，判别器会将图像和标签结合在一起，通过一系列的线性操作，输出一个标量值，表示输入图像的真实程度。

在上面的代码中，我们定义了一个名为 **Discriminator** 的类，它同样继承自 **nn.Module**。在类的初始化函数中，我们定义了判别器的结构，包括一个Embedding层、和三个线性层。在 **forward** 函数中，我们将输入的图像和标签拼接在一起，并通过一系列的线性操作将其转化为一个标量值。最后，我们将标量值通过 Sigmoid 函数压缩到 [0, 1] 的范围内，并返回表示输入图像真实程度的概率值。

然后这边我们初始化生成器和判别器网络对象。

我们使用BCE损失函数作为我们的损失函数。因为这是一个二元分类问题，其中输入是一张图片，输出是一个二元标签（真实或假的）。BCE损失函数可以度量预测输出（由鉴别器生成）和真实标签（真实或假的）之间的差异。

然后我们用Adam优化器作为我们生成器和判别器的优化器。

在训练过程中，我们首先从数据集中随机抽取一个批次的真实图像和对应的标签。然后，我们随机生成一个噪声向量 **noise**，并将它和标签label输入到生成器中，以生成一批虚假的图像。接下来，我们用真实图像和对应的标签训练判别器，同时用虚假图像和对应的标签训练生成器。在训练判别器时，我们首先计算真实图像和标签的预测值和损失，然后计算虚假图像和标签的预测值和损失，最后将这两个损失加起来作为判别器的总损失，用反向传播算法更新判别器的参数。在训练生成器时，我们将虚假图像和对应的标签输入到判别器中计算预测值和损失，然后将这个损失反向传播到生成器中，并用优化器更新生成器的参数。每100步，我们打印当前的迭代轮数、步数、判别器损失（d\_loss）和生成器损失（g\_loss）。

最后,我们可视化一些生成器生成的图像，看看训练的进展情况。

总之，上面的代码是一个条件生成对抗网络（Conditional GAN），用于训练时尚MNIST数据集。生成器模型通过一个随机噪声向量和图像类别标签来生成一张虚假的图像，而判别器模型则将图像和标签作为输入，试图区分真实图像和虚假图像。这种生成器和判别器的博弈过程可以使生成器逐渐生成更加逼真的图像，同时让判别器更加难以区分真实和虚假的图像。