计算机科学与技术学院神经网络与深度学习课程实验报告

实验题目： Conditional GAN		**学号：** 201900130015
日期： 2021.11.25	**班级：** 智能班	**姓名：** 李德锋
Email： ldf2878945468@163.com		

实验目的：

开发和评估一个有条件的生成对抗网络来生成服装项目的照片

实验软件和硬件环境：

Intel(R) Core(TM) i7-8550U CPU

华为云

实验原理和方法：

根据原理和公式补全代码，并测试

实验步骤：（不要求罗列完整源代码）

在 GAN 模型中使用类标签信息，完善 GAN，实现目标图像生成

生成器和鉴别器模型都以类标签为条件的方式来训练 GAN

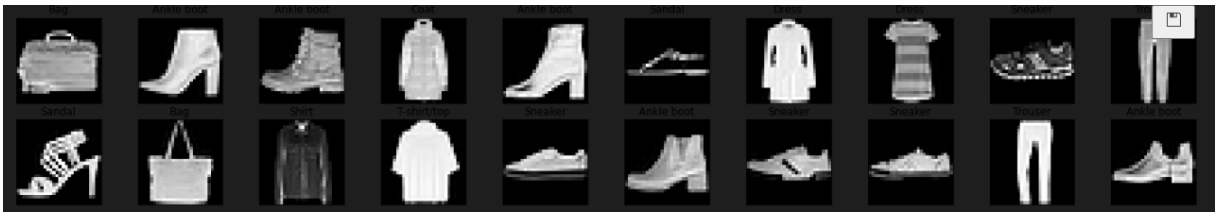
1.下载数据集

Fashion-MNIST 是一个数据集，由 60，000 个 28×28 像素的小正方形灰度图像组成

```
train_data = FashionMNIST(root='./data', train=True,
                          download=True, transform=data_transform)
```

2.显示图像

使用 imshow()函数绘制训练数据集，并通过“cmap”参数正确显示像素值



3. 构建生成器网络模型

```
architecture is (100+50)--->128--->256--->512--->1024--->(1,28,28)
```

nn.Sequential()

作为一个有顺序的容器，将特定神经网络模块按照在传入构造器的顺序依次被添加执行

```
self.model = nn.Sequential(
    nn.Linear(100+50, 128),
    nn.BatchNorm1d(128, 0.8),
    nn.LeakyReLU(0.2, inplace=True),

    nn.Linear(128, 256),
    nn.BatchNorm1d(256, 0.8),
    nn.LeakyReLU(0.2, inplace=True),

    nn.Linear(256, 512),
    nn.BatchNorm1d(512, 0.8),
    nn.LeakyReLU(0.2, inplace=True),

    nn.Linear(512, 1024),
    nn.BatchNorm1d(1024, 0.8),
    nn.LeakyReLU(0.2, inplace=True),

    nn.Linear(1024, opt.channels*opt.img_size*opt.img_size),
    nn.Tanh()
)
```

torch.cat （ (A, B), dim=1)是按列拼接，所以两个 tensor 的行数要相同

```
z = noise.view(noise.size(0), opt.latent_dim)
x = torch.cat([self.label_embedding(labels), z], 1)
img = self.model(x)
img = img.view(img.size(0), *img_shape)
```

4. 判别器

```
architecture is (100+784)--->512--->512--->512--->1
```

```
self.model = nn.Sequential(
    nn.Linear(50 + 784, 512),
    nn.LeakyReLU(0.2, inplace=True),

    nn.Linear(512, 512),
    nn.Dropout(0.4),
    nn.LeakyReLU(0.2, inplace=True),

    nn.Linear(512, 512),
    nn.Dropout(0.4),
    nn.LeakyReLU(0.2, inplace=True),

    nn.Linear(512, 1),
)
```

```
x = torch.cat([img.view(img.size(0), -1), self.label_embedding(labels)], 1)
validity = self.model(x)
```

5. 训练模型

```
optimizer_G.zero_grad( )

z = FloatTensor(np.random.normal(0, 1, (batch_size, opt.latent_dim)))
gen_labels = LongTensor(np.random.randint(0, opt.n_classes, batch_size))
gen_imgs = generator(z, gen_labels)


validity = discriminator(gen_imgs, gen_labels)
g_loss = adversarial_loss(validity, valid)

g_loss.backward()
optimizer_G.step()
```

```
optimizer_D.zero_grad( )

validity_real = discriminator(real_imgs, labels)
d_real_loss = adversarial_loss(validity_real, valid)

validity_fake = discriminator(gen_imgs.detach(), gen_labels)
d_fake_loss = adversarial_loss(validity_fake, fake)

d_loss = (d_real_loss + d_fake_loss) / 2


d_loss.backward()
optimizer_D.step()
```

```
[Epoch 2/200] [Batch 0/938] [D loss: 0.225447] [G loss: 0.303254]
[Epoch 2/200] [Batch 100/938] [D loss: 0.188554] [G loss: 0.494868]
[Epoch 2/200] [Batch 200/938] [D loss: 0.219584] [G loss: 0.388114]
[Epoch 2/200] [Batch 300/938] [D loss: 0.204119] [G loss: 0.400925]
[Epoch 2/200] [Batch 400/938] [D loss: 0.175461] [G loss: 0.449275]

show more (open the raw output data in a text editor) ...

[Epoch 199/200] [Batch 500/938] [D loss: 0.243099] [G loss: 0.268467]
[Epoch 199/200] [Batch 600/938] [D loss: 0.248634] [G loss: 0.251946]
[Epoch 199/200] [Batch 700/938] [D loss: 0.247393] [G loss: 0.248996]
[Epoch 199/200] [Batch 800/938] [D loss: 0.244154] [G loss: 0.259587]
[Epoch 199/200] [Batch 900/938] [D loss: 0.252481] [G loss: 0.254413]
```

运行示例加载保存的条件 GAN 模型，并使用它生成 100 件衣服

```
z = FloatTensor(np.random.normal(0, 1, (n_samples, latent_dim)))
labels = LongTensor(np.random.randint(0, n_classes, n_samples))
```

结论分析与体会：

学会了实现 Conditional GAN 并生成服装项目的照片

生成性对抗网络，简称 GAN，是一种用于训练基于深度学习的生成模型的架构。

该架构由一个生成器和一个鉴别器模型组成

类别标签，可以用于改进 GAN。以更稳定的训练、更快的训练和/或生成质量更好的图像的形式出现，类标签也可以用于有意或有针对性地生成给定类型的图像

就实验过程中遇到和出现的问题，你是如何解决和处理的，自拟 1－3 道问答题：

本地运行时间过长，报错

解决：使用华为云运行，调整参数