项目实践(2)：
利用CGAN生成fashion-mnist图像

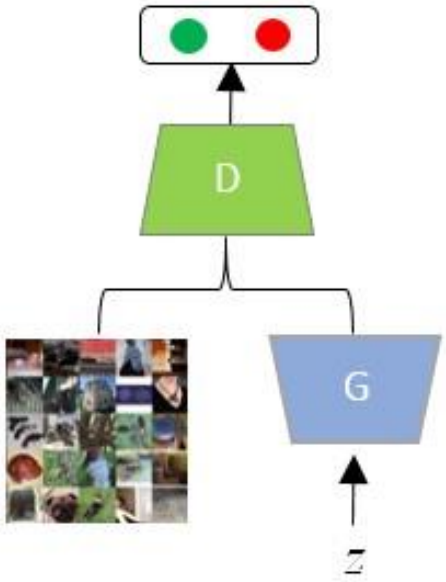
条件生成对抗网络CGAN

Name	Epoch 1	Epoch 20	Epoch 40
GAN			

条件生成对抗网络CGAN

Name	Epoch 1	Epoch 20	Epoch 40
GAN			
CGAN			

GAN

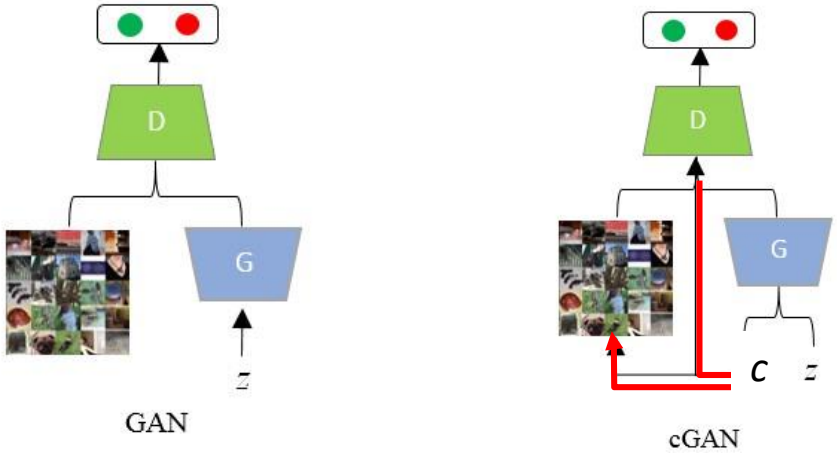


GAN

先来回忆GAN：

- 生成器G：输入一个噪声z，输出一个图像
- 判别器D：输入一个图像，输出该图像为真实图像的概率

GAN V.S. CGAN



CGAN(Conditional Generative Adversarial Networks):

- 为生成器和判别器都额外加入了一个条件变量c, 这个条件可以是希望生成的标签
- 生成器G：必须生成和条件c匹配的样本
- 判别器D：不仅要判别图像是否真实，还要判别图像和条件c是否匹配

GAN V.S. CGAN

在原始的GAN中，优化目标为：

$$L_D^{GAN} = E\left[\log\left(D(x)\right)\right] + E\left[\log\left(1 - D(G(z))\right)\right]$$

$$L_G^{GAN} = E\left[\log\left(D(G(z))\right)\right]$$

在CGAN中，只需做简单修改，加入条件c即可：

$$L_D^{CGAN} = E\left[\log\left(D(x,c)\right)\right] + E\left[\log\left(1 - D(G(z),c)\right)\right]$$

$$L_G^{CGAN} = E\left[\log\left(D(G(z),c)\right)\right]$$

文件夹结构


```
├── main.py # gateway
├── data
├── GAN.py # vanilla GAN
├── ops.py # some operations on layer
├── utils.py # utils
├── logs # log files for tensorboard to be saved here
└── checkpoint # model files to be saved here
```

CGAN.py

判别器*D*的网络结构


```
""" 搭建判别器网络结构 """
def discriminator(self, x, y, is_training=True, reuse=False):
    with tf.variable_scope("discriminator", reuse=reuse):

        # CGAN的判别器输入为图像x和标签y
        y = tf.reshape(y, [self.batch_size, 1, 1, self.y_dim])
        x = conv_cond_concat(x, y)
        net = lrelu(conv2d(x, 64, 4, 4, 2, 2, name='d_conv1'))
        net = lrelu(bn(conv2d(net, 128, 4, 4, 2, 2, name='d_conv2'), is_training=is_training, scope='d_bn2'))
        net = tf.reshape(net, [self.batch_size, -1])
        net = lrelu(bn(linear(net, 1024, scope='d_fc3'), is_training=is_training, scope='d_bn3'))
        out_logit = linear(net, 1, scope='d_fc4')
        out = tf.nn.sigmoid(out_logit)

        return out, out_logit, net
```

生成器G的网络结构


```
""" 搭建生成器网络结构 """
def generator(self, z, y, is_training=True, reuse=False):
    with tf.variable_scope("generator", reuse=reuse):

        # CGAN的生成器输入为噪声矢量z和标签y
        z = concat([z, y], 1)
        net = tf.nn.relu(bn(linear(z, 1024, scope='g_fc1'), is_training=is_training, scope='g_bn1'))
        net = tf.nn.relu(bn(linear(net, 128 * 7 * 7, scope='g_fc2'), is_training=is_training, scope='g_bn2'))
        net = tf.reshape(net, [self.batch_size, 7, 7, 128])
        net = tf.nn.relu(
            bn(deconv2d(net, [self.batch_size, 14, 14, 64], 4, 4, 2, 2, name='g_dc3'), is_training=is_training,
               scope='g_bn3'))

        out = tf.nn.sigmoid(deconv2d(net, [self.batch_size, 28, 28, 1], 4, 4, 2, 2, name='g_dc4'))

        return out
```

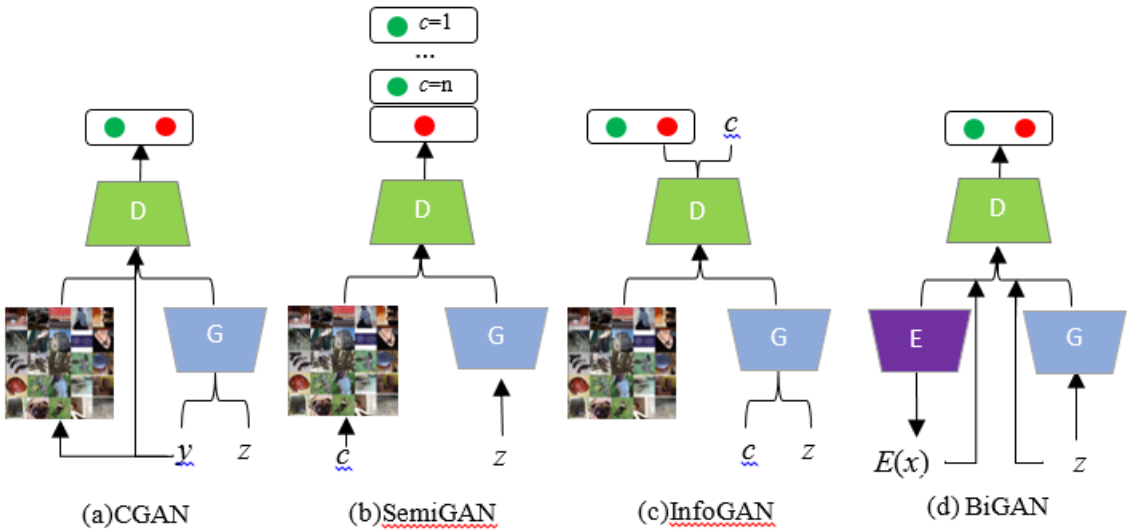
运行

python main.py --dataset fashion-mnist --gan_type CGAN --epoch 40 --batch_size 64

Name	Epoch 1	Epoch 20	Epoch 40
CGAN			

GAN的衍生模型

衍生模型的创新点包括：模型结构改进、理论扩展及应用等



王万良, 李卓蓉. 生成式对抗网络研究进展[J]. 通信学报, 2018, 39(2): 135-149.

延伸阅读/实践

- GAN的更详细介绍：http://blog.aylien.com/introduction-generative-adversarial-networks-code-tensorflow/
- 训练GAN的技巧清单：https://github.com/soumith/ganhacks
- pix2pix模型在线Demo: https://affinelayer.com/pixsrv/index.html
- 生成式模型合集：https://github.com/wiseodd/generative-models