PA4 GAN

陶天骅 2017010255 计81

任务内容

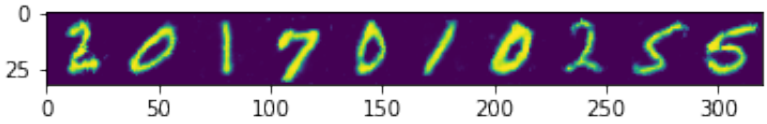
根据教程使用 jittor 生成学号图片。

我选择了在 Windows 上安装 Docker Desktop，使用`docker run -it -p 8888:8888 jittor/jittor`开启jupyter界面。

复制进`https://cg.cs.tsinghua.edu.cn/jittor/tutorial/2020-5-13-22-47-cgan/`的代码，运行生成学号图片。

运行代码可见附件`gan.html`，或最后的附图。

结果



In [1]:
```
!wget https://cg.cs.tsinghua.edu.cn/jittor/assets/build/generator_last.pkl
!wget https://cg.cs.tsinghua.edu.cn/jittor/assets/build/discriminator_last.pkl
```

```
--2020-05-23 12:28:59--  https://cg.cs.tsinghua.edu.cn/jittor/assets/build/generator_last.pkl
Resolving cg.cs.tsinghua.edu.cn (cg.cs.tsinghua.edu.cn)... 101.6.6.219
Connecting to cg.cs.tsinghua.edu.cn (cg.cs.tsinghua.edu.cn)|101.6.6.219|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 7045349 (6.7M)
Saving to: ‘generator_last.pkl’

generator_last.pkl  100%[===================>]   6.72M  4.74MB/s    in 1.4s

2020-05-23 12:29:12 (4.74 MB/s) - ‘generator_last.pkl’ saved [7045349/7045349]

--2020-05-23 12:29:12--  https://cg.cs.tsinghua.edu.cn/jittor/assets/build/discriminator_last.pkl
Resolving cg.cs.tsinghua.edu.cn (cg.cs.tsinghua.edu.cn)... 101.6.6.219
Connecting to cg.cs.tsinghua.edu.cn (cg.cs.tsinghua.edu.cn)|101.6.6.219|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4223997 (4.0M)
Saving to: ‘discriminator_last.pkl’

discriminator_last. 100%[===================>]   4.03M  4.70MB/s    in 0.9s

2020-05-23 12:29:13 (4.70 MB/s) - ‘discriminator_last.pkl’ saved [4223997/4223997]
```

In [2]:
```
!pwd
```

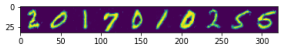
```
/root/.cache/jittor/notebook
```

In [3]:
```
import jittor as jt
from jittor import nn
import numpy as np
import pylab as pl

%matplotlib inline

# 隐空间向量长度
latent_dim = 100
# 类别数量
n_classes = 10
# 图片大小
img_size = 32
# 图片通道数量
channels = 1
# 图片张量的形状
img_shape = (channels, img_size, img_size)

class Generator(nn.Module):
    def __init__(self):
        super(Generator, self).__init__()
        self.label_emb = nn.Embedding(n_classes, n_classes)

        def block(in_feat, out_feat, normalize=True):
            layers = [nn.Linear(in_feat, out_feat)]
            if normalize:
                layers.append(nn.BatchNorm1d(out_feat, 0.8))
            layers.append(nn.LeakyReLU(0.2))
            return layers
        self.model = nn.Sequential(
            *block((latent_dim + n_classes), 128, normalize=False),
            *block(128, 256),
            *block(256, 512),
            *block(512, 1024),
            nn.Linear(1024, int(np.prod(img_shape))),
            nn.Tanh())

    def execute(self, noise, labels):
        gen_input = jt.contrib.concat((self.label_emb(labels), noise), dim=1)
        img = self.model(gen_input)
        img = img.view((img.shape[0], *img_shape))
        return img
class Discriminator(nn.Module):
    def __init__(self):
        super(Discriminator, self).__init__()
        self.label_embedding = nn.Embedding(n_classes, n_classes)
        self.model = nn.Sequential(
            nn.Linear((n_classes + int(np.prod(img_shape))), 512),
            nn.LeakyReLU(0.2),
            nn.Linear(512, 512),
            nn.Dropout(0.4),
            nn.LeakyReLU(0.2),
            nn.Linear(512, 512),
            nn.Dropout(0.4),
            nn.LeakyReLU(0.2),
            nn.Linear(512, 1))

    def execute(self, img, labels):
        d_in = jt.contrib.concat((img.view((img.shape[0], (- 1))), self.label_embedding(labels)), dim=1)
        validity = self.model(d_in)
        return validity

# 定义模型
generator = Generator()
discriminator = Discriminator()
generator.eval()
discriminator.eval()

# 加载参数
generator.load('./generator_last.pkl')
discriminator.load('./discriminator_last.pkl')
```

```
[SYNC][i 0523 12:29:36.843574 44 __init__.py:211] Found addr2line(2.30) at /usr/bin/addr2line.
[SYNC][i 0523 12:29:36.872892 44 compiler.py:849] pybind_include: -I/usr/include/python3.7m -I/usr/local/lib/python3.7/dist-packages/pybind11/include
[SYNC][i 0523 12:29:36.886487 44 compiler.py:851] extension_suffix: .cpython-37m-x86_64-linux-gnu.so

[i 0523 12:29:36.822735 40 __init__.py:219] Found g++(7.5.0) at /usr/bin/g++

[SYNC][i 0523 12:29:37.146679 44 __init__.py:140] Total mem: 15.64GB, using 5 procs for compiling.
[SYNC][i 0523 12:29:37.315349 44 jit_compiler.cc:20] Load cc_path: /usr/bin/g++
[SYNC][i 0523 12:29:37.384735 44 __init__.py:211] Found mpicc(2.1.1) at /usr/bin/mpicc.
```

In [4]:
```
number = "2017010255"
n_row = len(number)
z = jt.array(np.random.normal(0, 1, (n_row, latent_dim))).float32().stop_grad()
labels = jt.array(np.array([int(number[num]) for num in range(n_row)])).float32().stop_grad()
gen_imgs = generator(z,labels)

pl.imshow(gen_imgs.data.transpose((1,2,0,3))[0].reshape((gen_imgs.shape[2], -1)))
```

Out[4]:
```
<matplotlib.image.AxesImage at 0x7f278445bf90>
```



In [ ]: