**torchvision.datasets.MNIST(root,train,transform,download)**

加载MNIST数据集

**参数解释**

root：数据集的根目录

train：指定在数据集下载完成后需要载入的那部分数据，若为True则载入训练集；若为False则从载入测试集

transform：对图片进行变换

download：若为True，则查找根目录下的文件，若找到，则结束；若找不到则下载数据集并存放至根目录

train\_dataset = dsets.MNIST(root='./data',

train=True,

transform=transforms.ToTensor(),

download=True)

test\_dataset = dsets.MNIST(root='./data',

train=False,

transform=transforms.ToTensor())

print(train\_dataset)

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print(test\_dataset)

>>>

Dataset MNIST

Number of datapoints: 60000

Root location: ./data

Split: Train

StandardTransform

Transform: ToTensor()

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dataset MNIST

Number of datapoints: 10000

Root location: ./data

Split: Test

StandardTransform

Transform: ToTensor()

id = 100

img = train\_dataset[id][0].numpy()#img的大小为1×28×28

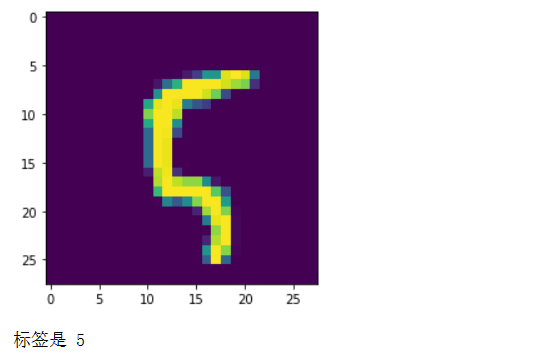
plt.figure()

plt.imshow(img[0,...])#转换为28×28后绘出

plt.show()

print('标签是',train\_dataset[id][1])

>>>



执行程序后，从电脑进入文件所在位置，可以看到其中包含训练集，测试集图像，标签及其对应的压缩文件  




t10k-images-idx3-ubyte：测试集图片

t10k-labels-idx3-ubyte：测试集标签

train-images-idx3-ubyte：训练集图片  
train-labels-idx3-ubyte：训练集标签

训练集和测试集必须都存在，若删除其中一者对应的4个文件，则另一个也无法载入