Q1 part 2

In [1]:

```
import numpy as np
import idx2numpy
import random
from matplotlib import pyplot as plt
import cv2
```

In [2]:

```
train_images = idx2numpy.convert_from_file('train-images.idx3-ubyte')
train_labels = idx2numpy.convert_from_file('train-labels.idx1-ubyte')
test_images = idx2numpy.convert_from_file('t10k-images.idx3-ubyte')
test_labels = idx2numpy.convert_from_file('t10k-labels.idx1-ubyte')
```

In [3]:

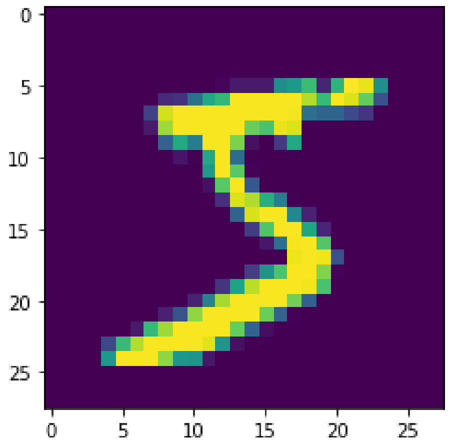
```
im1=train_images[0]
im2=train_images[56]
```

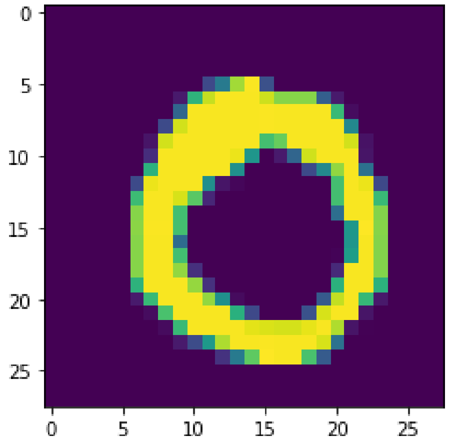
In [4]:

```
cv2.imwrite('color_img.jpg', im1)
plt.imshow(im1)
plt.figure()
cv2.imwrite('color_img.jpg', im2)
plt.imshow(im2)
```

Out[4]:

```
<matplotlib.image.AxesImage at 0x2473c0eaf88>
```





In [5]:

```
new_train=[]
for i in range(0,10):
    c=0
    for j in range(0,len(train_labels)):
        if(train_labels[j]==i and c<=1000):
            new_train.append(train_images[j])
            c+=1
```

In [6]:

```
train_new=[]
#flatten images to a 1d array
for i in new_train:
    i=np.array(i)
    train_new.append(i.flatten())
train_images=train_new
```

Using TSNE for dimensionality reduction

In [7]:

```
from sklearn.manifold import TSNE
embedded=X_embedded = TSNE(n_components=2, init='random').fit_transform(train_images)
```

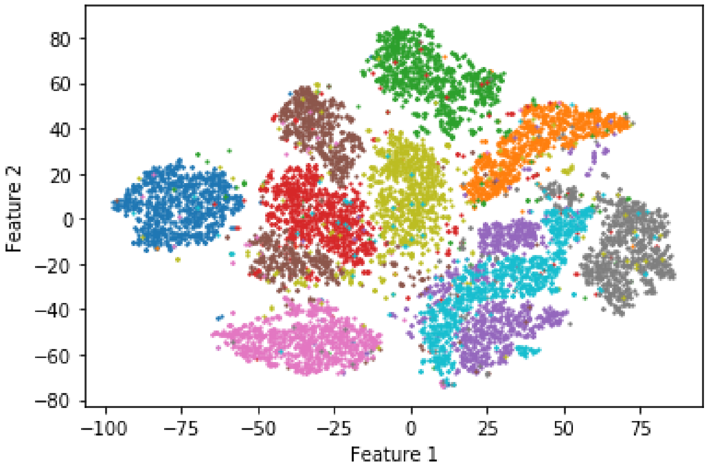
In [8]:

```
x=[]
y=[]
for i in embedded:
    x.append(i[0])
    y.append(i[1])
```

In [11]:

```
for i in range(0,10):
    x1=[]
    y1=[]
    for j in range(i*1000,(i+1)*1000):
        x1.append(x[j])
        y1.append(y[j])
    plt.scatter(x1,y1,s=1.5)

plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.show()
```



The Data is separable barring four classes that have seemed to mix up with each other (blue and purple, red and brown)