Gurmukhi Classification

```
#import necessary packages
import tensorflow as tf
from tensorflow import keras
import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
```

```
#creating a data set
(X_train, y_train) , (X_test, y_test) = keras.datasets.mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 [==============================] - 0s 0us/step
```

```
len(X_train)
```

```
60000
```

```
len(X_test)
```

```
10000
```

```
X_train[0].shape
```

```
(28, 28)
```

```
X_train[0]
```

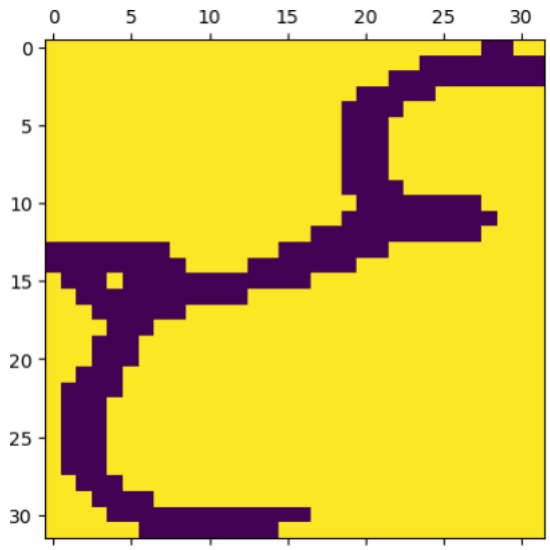
```
array([[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   3,
         18,  18,  18, 126, 136, 175,  26, 166, 255, 247, 127,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,  30,  36,  94, 154, 170,
        253, 253, 253, 253, 253, 225, 172, 253, 242, 195,  64,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,  49, 238, 253, 253, 253, 253,
        253, 253, 253, 253, 251,  93,  82,  82,  56,  39,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,  18, 219, 253, 253, 253, 253,
        253, 198, 182, 247, 241,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,  80, 156, 107, 253, 253,
        205,  11,   0,  43, 154,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,  14,   1, 154, 253,
         90,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0, 139, 253,
        190,   2,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  11, 190,
        253,  70,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  35,
        241, 225, 160, 108,   1,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         81, 240, 253, 253, 119,  25,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,  45, 186, 253, 253, 150,  27,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,  16,  93, 252, 253, 187,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0, 249, 253, 249,  64,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,  46, 130, 183, 253, 253, 207,   2,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  39,
        148, 229, 253, 253, 253, 250, 182,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  24, 114, 221,
        253, 253, 253, 253, 201,  78,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,  23,  66, 213, 253, 253,
        253, 253, 198,  81,   2,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,  18, 171, 219, 253, 253, 253, 253,
```

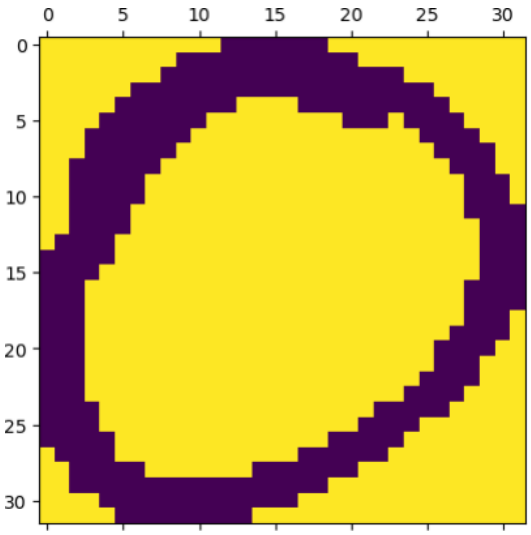
```
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  24, 114, 221,
 253, 253, 253, 253, 201,  78,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,  23,  66, 213, 253, 253,
 253, 253, 198,  81,   2,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,  18, 171, 219, 253, 253, 253, 253,
 195,  80,   9,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,  55, 172, 226, 253, 253, 253, 253, 244, 133,
  11,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0, 136, 253, 253, 253, 212, 135, 132,  16,   0,
   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0]], dtype=uint8)
```

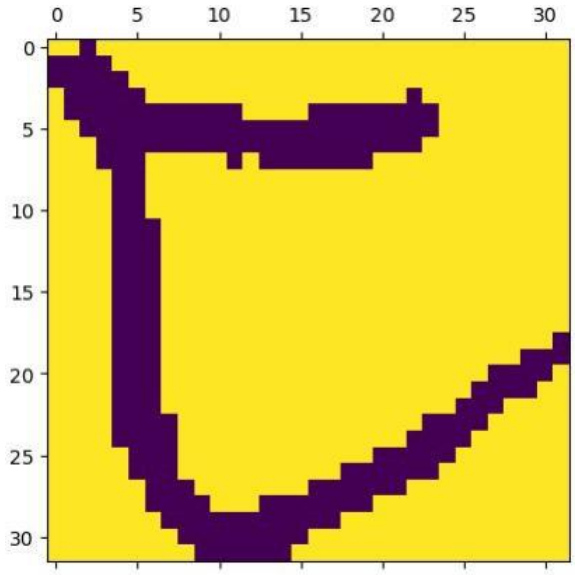
```
# Load the dataset
x_train = np.load('x_train.npy')
y_train = np.load('y_train.npy')
x_test = np.load('x_test.npy')
y_test = np.load('y_test.npy')
```

```
# test the images are loaded correctly

print(len(x_train))
print(len(x_test))
x_train[0].shape
x_train[0]
plt.matshow(x_train[0])
plt.matshow(x_train[999])
print(x_train.shape)
print(x_test.shape)
y_train
y_test
plt.matshow(x_test[150])
```

```
1000
178
(1000, 32, 32)
(178, 32, 32)
<matplotlib.image.AxesImage at 0x20ba468abb0>
```





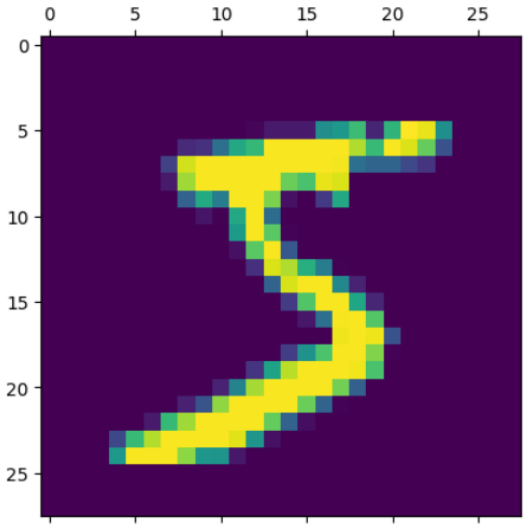



```
# # flatten the dataset i.e, change 2D to 1D (skipped this , and flattened in the model)

# x_train_flat = x_train.reshape(len(x_train),32*32)
# x_test_flat = x_test.reshape(len(x_test),32*32)

# print(x_train_flat.shape)
# print(x_test_flat.shape)

# x_train_flat[0]
```

```
# creating a simple nn
# create a dense layer where every input is connected to every other output, the number of inputs are 1000, outputs are 10
# activation function is sigmoid

model = keras.Sequential([
    keras.layers.Flatten(),
```

```
#plotting the graph
plt.matshow(X_train[0])
```

<matplotlib.image.AxesImage at 0x7fcb537a5c90>




```
[26] y_train[0]
```

5

```
[27] X_train = X_train / 255
     X_test = X_test / 255
```

```
X_train[0]
```

```
array([[0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.
, 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. ], [0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. ], [0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. ], [0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. ], [0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. ], [0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0.01176471, 0.07058824, 0.07058824, 0.07058824, 0.49411765, 0.53333333,
0.68627451, 0.10196078, 0.65098039, 1. , 0.96862745, 0.49803922, 0. ,
0. , 0. , 0. ], [0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.11764706,
0.14117647, 0.36862745, 0.60392157, 0.66666667, 0.99215686, 0.99215686,
0.99215686, 0.99215686, 0.99215686, 0.88235294, 0.6745098 , 0.99215686,
0.94901961, 0.76470588, 0.25098039, 0. , 0. , 0. , 0. ], [0. , 0. , 0.
, 0. , 0. , 0. , 0. , 0.19215686, 0.93333333, 0.99215686, 0.99215686,
0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686,
0.98431373, 0.36470588, 0.32156863, 0.32156863, 0.21960784, 0.15294118,
0. , 0. , 0. , 0. , 0. ], [0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0.07058824, 0.85882353, 0.99215686, 0.99215686, 0.99215686, 0.99215686,
0.99215686, 0.77647059, 0.71372549, 0.96862745, 0.94509804, 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ], [0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0.31372549, 0.61176471, 0.41960784, 0.99215686, 0.99215686,
0.80392157, 0.04313725, 0. , 0.16862745, 0.60392157, 0. , 0. , 0. , 0.
```

, 0. , 0. , 0. , 0. , 0. , 0.], [0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.
, 0. , 0.05490196, 0.00392157, 0.60392157, 0.99215686, 0.35294118, 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.], [0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.54509804,
0.99215686, 0.74509804, 0.00784314, 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0.], [0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0.04313725, 0.74509804, 0.99215686, 0.2745098 , 0. , 0.
, 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.], [0. , 0. , 0.
, 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.1372549 , 0.94509804,
0.88235294, 0.62745098, 0.42352941, 0.00392157, 0. , 0. , 0. , 0. , 0.
, 0. , 0. , 0. , 0. , 0. , 0.], [0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.
, 0. , 0. , 0. , 0. , 0. , 0.31764706, 0.94117647, 0.99215686, 0.99215686,
0.46666667, 0.09803922, 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.],
[0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0.17647059, 0.72941176, 0.99215686, 0.99215686, 0.58823529, 0.10588235,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.], [0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.0627451 , 0.36470588,
0.98823529, 0.99215686, 0.73333333, 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0.], [0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.97647059, 0.99215686, 0.97647059, 0.25098039, 0.
, 0. , 0. , 0. , 0. , 0. , 0.], [0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.
, 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.18039216, 0.50980392, 0.71764706,
0.99215686, 0.99215686, 0.81176471, 0.00784314, 0. , 0. , 0. , 0. , 0.
, 0. , 0.], [0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.
, 0.15294118, 0.58039216, 0.89803922, 0.99215686, 0.99215686,
0.99215686, 0.98039216, 0.71372549, 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0.], [0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.09411765,
0.44705882, 0.86666667, 0.99215686, 0.99215686, 0.99215686, 0.99215686,
0.78823529, 0.30588235, 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.],
[0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.09019608, 0.25882353,
0.83529412, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.77647059,
0.31764706, 0.00784314, 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.
], [0. , 0. , 0. , 0. , 0. , 0. , 0.07058824, 0.67058824, 0.85882353,
0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.76470588, 0.31372549,
0.03529412, 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.
], [0. , 0. , 0. , 0. , 0.21568627, 0.6745098 , 0.88627451, 0.99215686,
0.99215686, 0.99215686, 0.99215686, 0.95686275, 0.52156863, 0.04313725,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.],
[0. , 0. , 0. , 0. , 0.53333333, 0.99215686, 0.99215686, 0.99215686,
0.83137255, 0.52941176, 0.51764706, 0.0627451 , 0. , 0. , 0. , 0. , 0.
, 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.], [0. , 0. , 0.
, 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.], [0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.], [0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.]])

```
[29] X_train_flattened = X_train.reshape(len(X_train), 28*28)
     X_test_flattened = X_test.reshape(len(X_test), 28*28)
```

```
[30] X_train_flattened.shape
```

```
(60000, 784)
```

```
X_train_flattened[0]
```

```
array([0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0.01176471, 0.07058824, 0.07058824, 0.07058824, 0.49411765, 0.53333333,
0.68627451, 0.10196078, 0.65098039, 1. , 0.96862745, 0.49803922, 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.11764706,
0.14117647, 0.36862745, 0.60392157, 0.66666667, 0.99215686, 0.99215686,
0.99215686, 0.99215686, 0.99215686, 0.88235294, 0.6745098 , 0.99215686,
0.94901961, 0.76470588, 0.25098039, 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0.19215686, 0.93333333, 0.99215686, 0.99215686,
0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686,
0.98431373, 0.36470588, 0.32156863, 0.32156863, 0.21960784, 0.15294118,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.07058824,
0.85882353, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686,
0.77647059, 0.71372549, 0.96862745, 0.94509804, 0. , 0. , 0. , 0. , 0.
, 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0.31372549, 0.61176471, 0.41960784, 0.99215686, 0.99215686, 0.80392157,
0.04313725, 0. , 0.16862745, 0.60392157, 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0.05490196, 0.00392157, 0.60392157, 0.99215686, 0.35294118, 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.54509804, 0.99215686,
0.74509804, 0.00784314, 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.
, 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0.04313725, 0.74509804, 0.99215686, 0.2745098 , 0. , 0. , 0. , 0. , 0.
, 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0.1372549 , 0.94509804, 0.88235294,
0.62745098, 0.42352941, 0.00392157, 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0.31764706, 0.94117647, 0.99215686, 0.99215686, 0.46666667,
0.09803922, 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.17647059,
0.72941176, 0.99215686, 0.99215686, 0.58823529, 0.10588235, 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.0627451 , 0.36470588, 0.98823529,
0.99215686, 0.73333333, 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.
, 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0.97647059, 0.99215686, 0.97647059, 0.25098039, 0. , 0. , 0. , 0.
```

```
, 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0.18039216, 0.50980392, 0.71764706, 0.99215686,
0.99215686, 0.81176471, 0.00784314, 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.15294118,
0.58039216, 0.89803922, 0.99215686, 0.99215686, 0.99215686, 0.98039216,
0.71372549, 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0.09411765, 0.44705882, 0.86666667,
0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.78823529, 0.30588235,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0.09019608, 0.25882353, 0.83529412, 0.99215686,
0.99215686, 0.99215686, 0.99215686, 0.77647059, 0.31764706, 0.00784314,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0.07058824, 0.67058824, 0.85882353, 0.99215686, 0.99215686,
0.99215686, 0.99215686, 0.76470588, 0.31372549, 0.03529412, 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0.21568627, 0.6745098 , 0.88627451, 0.99215686, 0.99215686, 0.99215686,
0.99215686, 0.95686275, 0.52156863, 0.04313725, 0. , 0. , 0. , 0. , 0.
, 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0.53333333, 0.99215686, 0.99215686, 0.99215686, 0.83137255, 0.52941176,
0.51764706, 0.0627451 , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.
, 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. ,
0. , 0. , 0. , 0. , 0. , 0. ])
```

```
#modelling data and validating data
model = keras.Sequential([
    keras.layers.Dense(10, input_shape=(784,), activation='sigmoid')
])
```

```
[33] model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

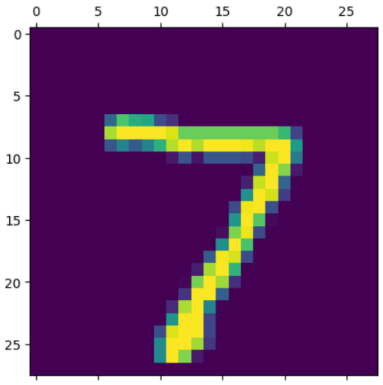
```
[34] model.fit(X_train_flattened, y_train, epochs=5)

Epoch 1/5
1875/1875 [==============================] - 4s 2ms/step - loss: 0.4734 - accuracy: 0.8746
Epoch 2/5
1875/1875 [==============================] - 5s 3ms/step - loss: 0.3047 - accuracy: 0.9153
Epoch 3/5
1875/1875 [==============================] - 3s 2ms/step - loss: 0.2834 - accuracy: 0.9208
Epoch 4/5
1875/1875 [==============================] - 3s 2ms/step - loss: 0.2733 - accuracy: 0.9243
Epoch 5/5
1875/1875 [==============================] - 4s 2ms/step - loss: 0.2661 - accuracy: 0.9263
<keras.callbacks.History at 0x7fcb537fb6a0>
```

```
[35] model.evaluate(X_test_flattened, y_test)

313/313 [==============================] - 2s 5ms/step - loss: 0.2687 - accuracy: 0.9242
[0.26867222785949707, 0.9241999983787537]
```

```
y_predicted = model.predict(X_test_flattened)
y_predicted[0]

313/313 [==============================] - 1s 2ms/step
array([2.1566955e-02, 3.2233328e-07, 6.6303402e-02, 9.6107769e-01,
       2.5136338e-03, 1.3426013e-01, 2.5372722e-06, 9.9976873e-01,
       9.7535647e-02, 6.5442240e-01], dtype=float32)
```

```
plt.matshow(X_test[0])
```

```
<matplotlib.image.AxesImage at 0x7fcb67d20c40>
```




```
[38] np.argmax(y_predicted[0])
```

```
7
```

```
y_predicted_labels = [np.argmax(i) for i in y_predicted]
```

```
[40] y_predicted_labels[:5]
```

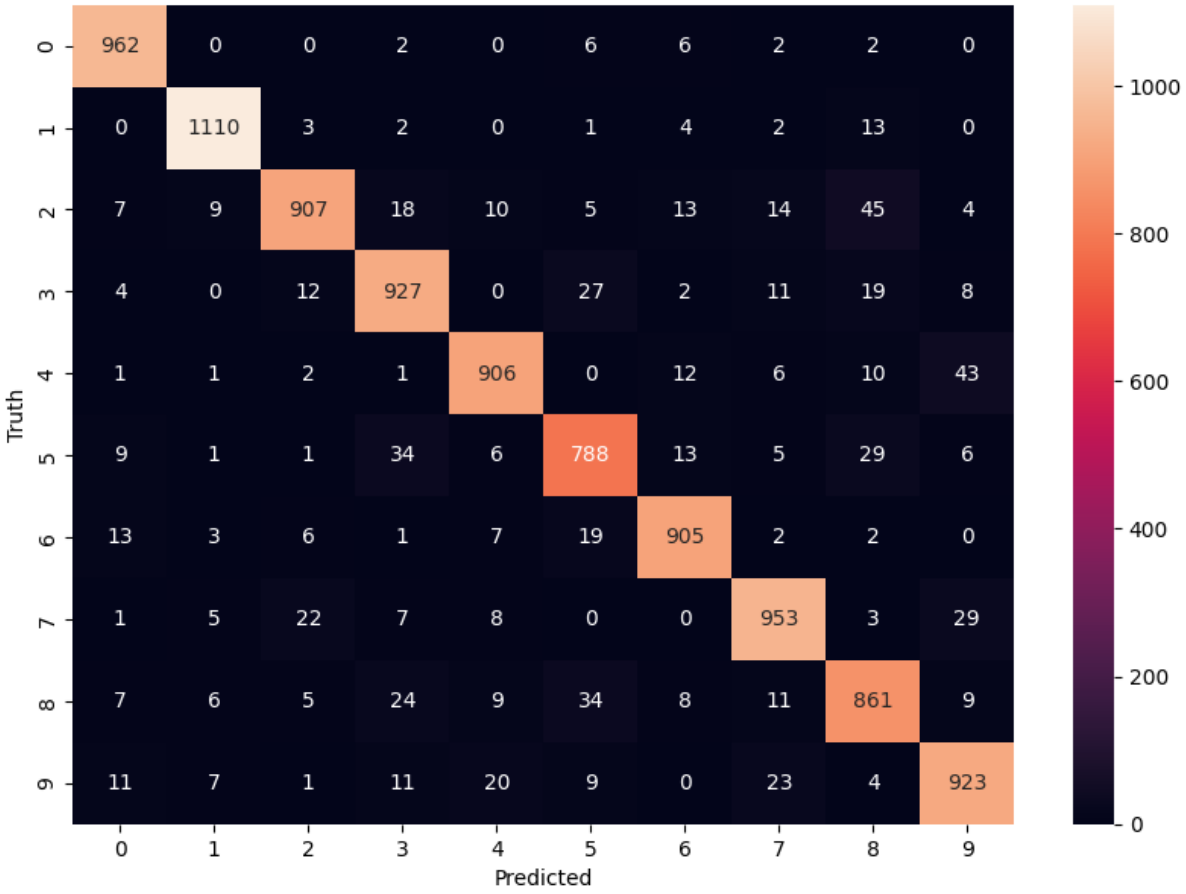
```
[7, 2, 1, 0, 4]
```

```
[41] cm = tf.math.confusion_matrix(labels=y_test,predictions=y_predicted_labels)
     cm
```

```
<tf.Tensor: shape=(10, 10), dtype=int32, numpy=
array([[ 962,    0,    0,    2,    0,    6,    6,    2,    2,    0],
       [   0, 1110,    3,    2,    0,    1,    4,    2,   13,    0],
       [   7,    9,  907,   18,   10,    5,   13,   14,   45,    4],
       [   4,    0,   12,  927,    0,   27,    2,   11,   19,    8],
       [   1,    1,    2,    1,  906,    0,   12,    6,   10,   43],
       [   9,    1,    1,   34,    6,  788,   13,    5,   29,    6],
       [  13,    3,    6,    1,    7,   19,  905,    2,    2,    0],
       [   1,    5,   22,    7,    8,    0,    0,  953,    3,   29],
       [   7,    6,    5,   24,    9,   34,    8,   11,  861,    9],
       [  11,    7,    1,   11,   20,    9,    0,   23,    4,  923]],
      dtype=int32)>
```

```
#Using hidden layer

import seaborn as sn
plt.figure(figsize = (10,7))
sn.heatmap(cm, annot=True, fmt='d')
plt.xlabel('Predicted')
plt.ylabel('Truth')
```

```
Text(95.72222222222221, 0.5, 'Truth')
```




```
model = keras.Sequential([
    keras.layers.Dense(100, input_shape=(784,), activation='relu'),
    keras.layers.Dense(10, activation='sigmoid')
])
```

```
[44] model.compile(optimizer='adam',
                   loss='sparse_categorical_crossentropy',
                   metrics=['accuracy'])
```

```
[45] model.fit(X_train_flattened, y_train, epochs=5)
```

```
Epoch 1/5
1875/1875 [==============================] - 7s 3ms/step - loss: 0.2739 - accuracy: 0.9228
Epoch 2/5
1875/1875 [==============================] - 8s 4ms/step - loss: 0.1249 - accuracy: 0.9636
Epoch 3/5
1875/1875 [==============================] - 7s 4ms/step - loss: 0.0864 - accuracy: 0.9742
Epoch 4/5
1875/1875 [==============================] - 8s 4ms/step - loss: 0.0664 - accuracy: 0.9804
Epoch 5/5
1875/1875 [==============================] - 6s 3ms/step - loss: 0.0521 - accuracy: 0.9842
<keras.callbacks.History at 0x7fcb532967a0>
```