handwritten-digits-classification

February 11, 2024

1 Handwritten Digits Classification

Handwritten digits classification is a classic machine learning problem that involves identifying and classifying handwritten digits (0-9) from images. This problem has practical applications in various fields such as postal automation, document processing, and security systems.

The goal of handwritten digits classification is to develop a model that can accurately predict the digit represented by a given image. This can be achieved using various machine learning techniques, including:

- **Deep Learning:** Deep learning models, particularly Convolutional Neural Networks (CNNs), have achieved state-of-the-art performance in handwritten digits classification. CNNs are designed to process images and can automatically learn features from the data, eliminating the need for manual feature engineering.

[29]:
```python
import tensorflow as tf
from tensorflow import keras
import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
```

Loading data from keras

[2]:
```python
(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
```

[3]:
```python
len(x_train)
```

[3]:
```
60000
```

[4]:
```python
x_train[0].shape
```

[4]:
```
(28, 28)
```

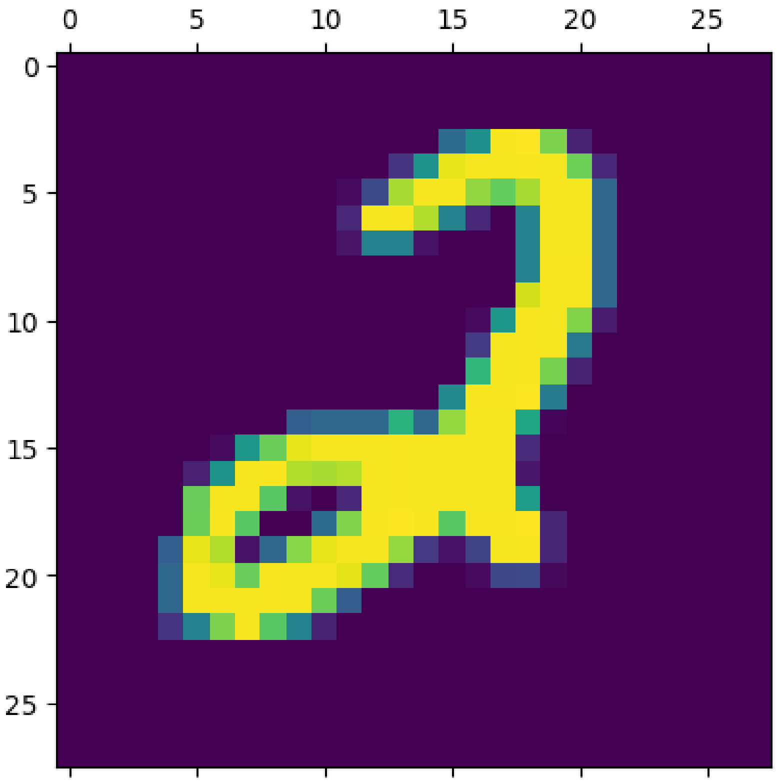
[5]:
```python
x_train[0]
```

```
[5]: 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,
            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)
```

Plotting images from the data using **Matplotlib**

```
[7]: plt.matshow(x_train[233])
```

```
[7]: <matplotlib.image.AxesImage at 0x798bdc9c77c0>
```



Scalling values

[8]:
```
x_train=x_train/255
x_test=x_test/255
```

Converting 2D Array into single dimentional array

[9]:
```
x_train_flattended= x_train.reshape(len(x_train),28*28)
```

[10]:
```
x_test_flattended= x_test.reshape(len(x_test),28*28)
```

[11]:
```
x_train_flattended.shape
```

[11]:
```
(60000, 784)
```

[12]:
```
x_test_flattended
```

[12]:
```
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.]])
```

```
[13]: x_train[0]
```

```
[13]: 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.        ]])
```

#Neural Network model using Keras

[14]:

```
model = keras.Sequential([
    keras.layers.Dense(10, input_shape=(784,), activation='sigmoid')
])
model.compile(
    optimizer='adam',
    loss='sparse_categorical_crossentropy',
    metrics=['accuracy']
)
model.fit(x_train_flattended, y_train, epochs=10)
```

```
Epoch 1/10
1875/1875 [==============================] - 2s 982us/step - loss: 0.4674 -
accuracy: 0.8787
```

```
Epoch 2/10
1875/1875 [==============================] - 2s 966us/step - loss: 0.3043 -
accuracy: 0.9151
Epoch 3/10
1875/1875 [==============================] - 2s 1ms/step - loss: 0.2835 -
accuracy: 0.9204
Epoch 4/10
1875/1875 [==============================] - 2s 979us/step - loss: 0.2737 -
accuracy: 0.9233
Epoch 5/10
1875/1875 [==============================] - 2s 977us/step - loss: 0.2664 -
accuracy: 0.9259
Epoch 6/10
1875/1875 [==============================] - 2s 979us/step - loss: 0.2617 -
accuracy: 0.9268
Epoch 7/10
1875/1875 [==============================] - 2s 984us/step - loss: 0.2584 -
accuracy: 0.9285
Epoch 8/10
1875/1875 [==============================] - 2s 977us/step - loss: 0.2555 -
accuracy: 0.9296
Epoch 9/10
1875/1875 [==============================] - 2s 1ms/step - loss: 0.2529 -
accuracy: 0.9295
Epoch 10/10
1875/1875 [==============================] - 2s 1ms/step - loss: 0.2511 -
accuracy: 0.9302
```

[14]:
```
<keras.src.callbacks.History at 0x798bdf13a980>
```

evaluating the model

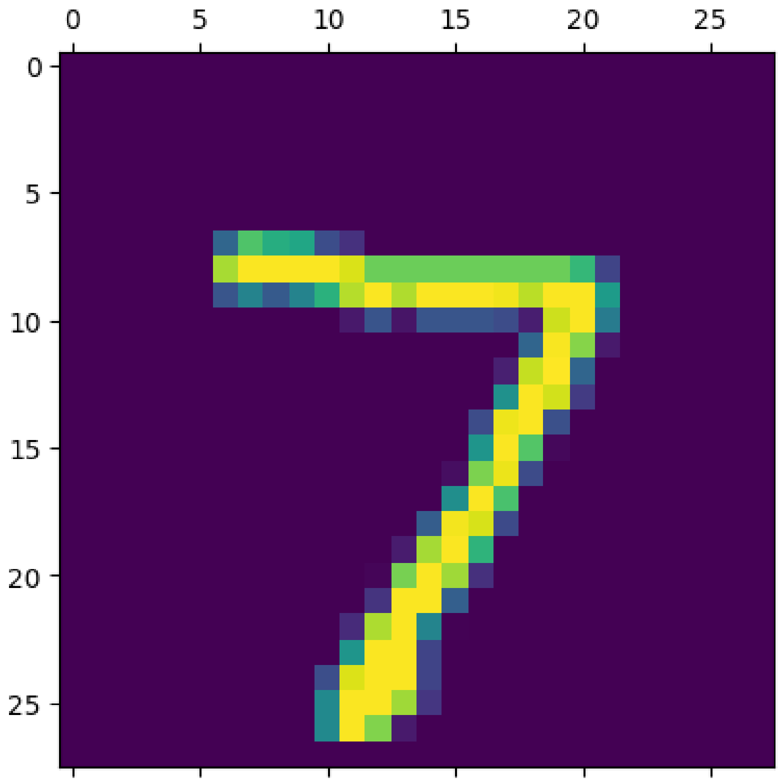
[17]:
```
model.evaluate(x_test_flattended,y_test)
```

```
313/313 [==============================] - 0s 855us/step - loss: 0.2669 -
accuracy: 0.9254
```

[17]:
```
[0.2668640613555908, 0.9254000186920166]
```

[21]:
```
plt.matshow(x_test[0])
```

[21]:
```
<matplotlib.image.AxesImage at 0x798bdf1cada0>
```



2 Prediction of model

[22]:
```
predicted_value=model.predict(x_test_flattended)
predicted_value[0]
```

```
313/313 [==============================] - 0s 778us/step
```

[22]:
```
array([4.5268373e-03, 1.6215136e-08, 2.0346729e-02, 9.5916164e-01,
       1.6172243e-03, 1.8915704e-01, 4.5303238e-08, 9.9989504e-01,
       1.0035723e-01, 6.4365256e-01], dtype=float32)
```

[23]:
```
np.argmax(predicted_value[0])
```

[23]:
```
7
```

[30]:
```
y_predicted_labels=[np.argmax(i) for i in predicted_value ]
y_predicted_labels[:10]
```

[30]:
```
[7, 2, 1, 0, 4, 1, 4, 9, 6, 9]
```

[33]:

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

[33]:

```
<tf.Tensor: shape=(10, 10), dtype=int32, numpy=
array([[ 959,    0,    2,    2,    0,    6,    8,    2,    1,    0],
       [   0, 1111,    3,    2,    0,    1,    4,    2,   12,    0],
       [   4,    8,  924,   12,    8,    5,   14,    8,   46,    3],
       [   3,    0,   21,  904,    1,   31,    4,   12,   28,    6],
       [   1,    1,    5,    2,  913,    0,   11,    4,   10,   35],
       [  10,    3,    2,   21,    9,  782,   17,    8,   35,    5],
       [  10,    3,    9,    1,    7,   10,  914,    2,    2,    0],
       [   1,    6,   26,    2,    7,    1,    0,  957,    3,   25],
       [   6,    8,    6,   14,    9,   25,    9,   12,  879,    6],
       [  10,    8,    1,    7,   26,    8,    0,   29,    9,  911]],
      dtype=int32)>
```

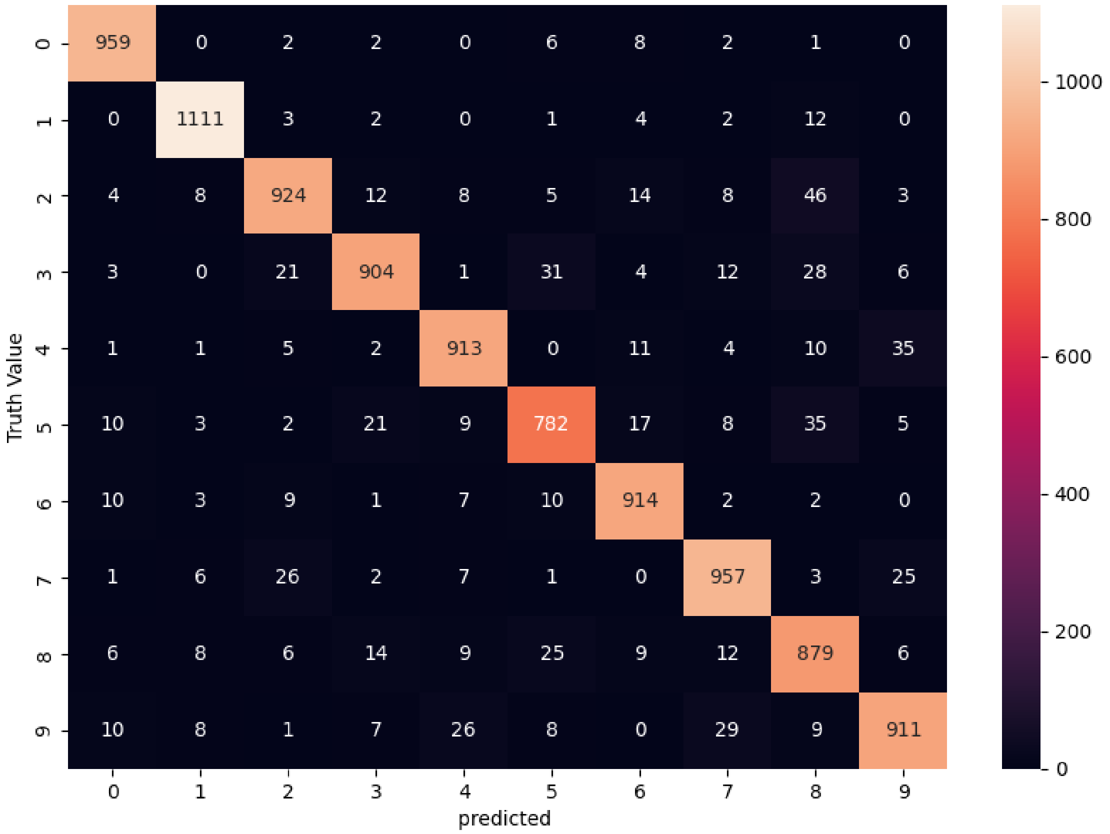
#Adding confusion matrix

[34]:

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

[34]:

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



3 Adding hidden layer

after adding hidden layer accuracy will increase.

[35]:
```
model = keras.Sequential([
    keras.layers.Dense(100, input_shape=(784,), activation='relu'),
    keras.layers.Dense(10, activation='sigmoid')
])
model.compile(
    optimizer='adam',
    loss='sparse_categorical_crossentropy',
    metrics=['accuracy']
)
model.fit(x_train_flattended, y_train, epochs=10)
```

```
Epoch 1/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.2759 -
accuracy: 0.9212
Epoch 2/10
1875/1875 [==============================] - 3s 2ms/step - loss: 0.1275 -
```

```
accuracy: 0.9622
Epoch 3/10
1875/1875 [==============================] - 3s 1ms/step - loss: 0.0887 -
accuracy: 0.9736
Epoch 4/10
1875/1875 [==============================] - 3s 1ms/step - loss: 0.0663 -
accuracy: 0.9796
Epoch 5/10
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0535 -
accuracy: 0.9833
Epoch 6/10
1875/1875 [==============================] - 3s 1ms/step - loss: 0.0419 -
accuracy: 0.9867
Epoch 7/10
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0341 -
accuracy: 0.9894
Epoch 8/10
1875/1875 [==============================] - 3s 1ms/step - loss: 0.0279 -
accuracy: 0.9915
Epoch 9/10
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0230 -
accuracy: 0.9929
Epoch 10/10
1875/1875 [==============================] - 3s 2ms/step - loss: 0.0184 -
accuracy: 0.9943
```

```
[35]: <keras.src.callbacks.History at 0x798bb1706860>
```

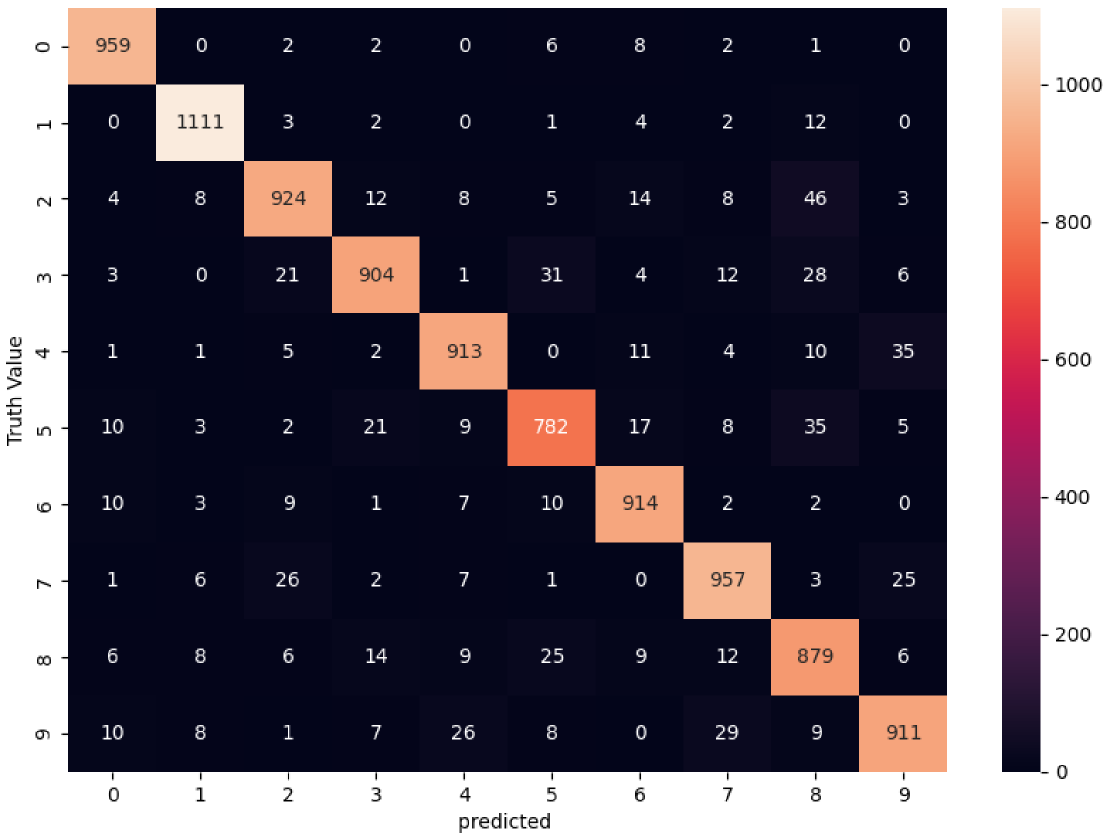
```
[36]: model.evaluate(x_test_flattended,y_test)
```

```
313/313 [==============================] - 1s 2ms/step - loss: 0.0811 -
accuracy: 0.9767
```

```
[36]: [0.08113346993923187, 0.9767000079154968]
```

```
[37]: import seaborn as sn
      plt.figure(figsize=(10,7))
      sn.heatmap(cm,annot=True,fmt='d')
      plt.xlabel('predicted ')
      plt.ylabel('Truth Value')
```

```
[37]: Text(95.72222222222221, 0.5, 'Truth Value')
```



	0	1	2	3	4	5	6	7	8	9
0	959	0	2	2	0	6	8	2	1	0
1	0	1111	3	2	0	1	4	2	12	0
2	4	8	924	12	8	5	14	8	46	3
3	3	0	21	904	1	31	4	12	28	6
4	1	1	5	2	913	0	11	4	10	35
5	10	3	2	21	9	782	17	8	35	5
6	10	3	9	1	7	10	914	2	2	0
7	1	6	26	2	7	1	0	957	3	25
8	6	8	6	14	9	25	9	12	879	6
9	10	8	1	7	26	8	0	29	9	911

With flatenned array meanully. using keras Flatten method