classification

November 5, 2023

Experiment No 4 Write a program to perform classification tasks over given data using direct functions and evaluate its performance

Name Atharva Balpande Roll no 313002 Div:C1 PRN 22111143

[ ]:
```
import tensorflow as tf
import numpy as np
mnist = tf.keras.datasets.mnist
(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test=x_train/255.0, x_test/255.0
```

[ ]:
```
print("Training Tensor")
print("number if dimension in tensor: %d"%x_train.ndim)
print("Tensor Shape", x_train.shape)
print("Data type", x_train.dtype)
print()
print("Testing Tensor")
print("number if dimension in tensor: %d"%x_test.ndim)
print("Tensor Shape", x_test.shape)
```

```
Training Tensor
number if dimension in tensor: 3
Tensor Shape (60000, 28, 28)
Data type float64

Testing Tensor
number if dimension in tensor: 3
Tensor Shape (10000, 28, 28)
```

[ ]:
```
print("Data type", x_test.dtype)
```

```
Data type float64
```

[ ]:
```
model=tf.keras.models.Sequential([
tf.keras.layers.Flatten(input_shape=(28, 28)),
tf.keras.layers.Dense(128, activation='relu'),
tf.keras.layers.Dropout(0.2),
tf.keras.layers.Dense(10, activation='softmax')
])
model.compile(optimizer='adam',
```

```
loss='sparse_categorical_crossentropy',
metrics=['accuracy'])
model.fit(x_train, y_train, epochs=5)
model.evaluate(x_test, y_test, verbose=2)
model.summary()
```

```
Epoch 1/5
1875/1875 [==============================] - 5s 2ms/step - loss: 0.2942 -
accuracy: 0.9144
Epoch 2/5
1875/1875 [==============================] - 5s 3ms/step - loss: 0.1411 -
accuracy: 0.9571
Epoch 3/5
1875/1875 [==============================] - 5s 3ms/step - loss: 0.1063 -
accuracy: 0.9674
Epoch 4/5
1875/1875 [==============================] - 4s 2ms/step - loss: 0.0869 -
accuracy: 0.9736
Epoch 5/5
1875/1875 [==============================] - 4s 2ms/step - loss: 0.0745 -
accuracy: 0.9775
313/313 - 1s - loss: 0.0723 - accuracy: 0.9785 - 516ms/epoch - 2ms/step
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 flatten (Flatten)           (None, 784)               0

 dense (Dense)               (None, 128)               100480

 dropout (Dropout)           (None, 128)               0

 dense_1 (Dense)             (None, 10)                1290

=================================================================
Total params: 101770 (397.54 KB)
Trainable params: 101770 (397.54 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

[ ]:
```
x_test[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.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.32941176, 0.7254902 , 0.62352941, 0.59215686,
 0.23529412, 0.14117647, 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.87058824, 0.99607843, 0.99607843, 0.99607843,
 0.99607843, 0.94509804, 0.77647059, 0.77647059, 0.77647059,
 0.77647059, 0.77647059, 0.77647059, 0.77647059, 0.77647059,
 0.66666667, 0.20392157, 0.        , 0.        , 0.        ,
```

```
 0.        , 0.        , 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.2627451 , 0.44705882, 0.28235294, 0.44705882,
 0.63921569, 0.89019608, 0.99607843, 0.88235294, 0.99607843,
 0.99607843, 0.99607843, 0.98039216, 0.89803922, 0.99607843,
 0.99607843, 0.54901961, 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.06666667, 0.25882353, 0.05490196, 0.2627451 ,
 0.2627451 , 0.2627451 , 0.23137255, 0.08235294, 0.9254902 ,
 0.99607843, 0.41568627, 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.3254902 , 0.99215686,
 0.81960784, 0.07058824, 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.08627451, 0.91372549, 1.        ,
 0.3254902 , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.50588235, 0.99607843, 0.93333333,
 0.17254902, 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.23137255, 0.97647059, 0.99607843, 0.24313725,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.52156863, 0.99607843, 0.73333333, 0.01960784,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.03529412, 0.80392157, 0.97254902, 0.22745098, 0.        ,
```

```
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.49411765, 0.99607843, 0.71372549, 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.29411765,
 0.98431373, 0.94117647, 0.22352941, 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.0745098 , 0.86666667,
 0.99607843, 0.65098039, 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, 0.79607843, 0.99607843,
 0.85882353, 0.1372549 , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.14901961, 0.99607843, 0.99607843,
 0.30196078, 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.12156863, 0.87843137, 0.99607843, 0.45098039,
 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.        , 0.52156863, 0.99607843, 0.99607843, 0.20392157,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        ],
[0.        , 0.        , 0.        , 0.        , 0.        ,
 0.        , 0.        , 0.        , 0.        , 0.        ,
 0.23921569, 0.94901961, 0.99607843, 0.99607843, 0.20392157,
```

```
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.4745098 , 0.99607843, 0.99607843, 0.85882353, 0.15686275,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.4745098 , 0.99607843, 0.81176471, 0.07058824, 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ]])
```

[ ]:
```
predictions = model.predict(x_test)
print(predictions[1020])
np.argmax(predictions[1020])
```

```
313/313 [==============================] - 1s 1ms/step
[4.9627691e-10 1.8739981e-09 5.5064406e-06 9.9997544e-01 1.4164112e-10
 1.2414010e-05 1.7222192e-11 4.9364616e-07 5.2688438e-06 9.6760698e-07]
```

[ ]:
```
3
```

[ ]:
```
import matplotlib.pyplot as plt
plt.imshow(x_test[1020], cmap='gray')
pred=[]
for j in range(len(predictions)):
        pred.append(np.argmax(predictions[j]))
pred
```

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

9,
0,
6,
9,
0,
1,
5,
9,
7,
3,
4,
9,
6,
6,
5,
4,
0,
7,
4,
0,
1,
3,
1,
3,
4,
7,
2,
7,
1,
2,
1,
1,
7,
4,
2,
3,
5,
1,
2,
4,
4,
6,
3,
5,
5,
6,
0,

4,
1,
9,
5,
7,
8,
9,
3,
7,
4,
6,
4,
3,
0,
7,
0,
2,
9,
1,
7,
3,
2,
9,
7,
7,
6,
2,
7,
8,
4,
7,
3,
6,
1,
3,
6,
9,
3,
1,
4,
1,
7,
6,
9,
6,
0,
5,

4,
9,
9,
2,
1,
9,
4,
8,
7,
3,
9,
7,
4,
4,
4,
9,
2,
5,
4,
7,
6,
7,
9,
0,
5,
8,
5,
6,
6,
5,
7,
8,
1,
0,
1,
6,
4,
6,
7,
3,
1,
7,
1,
8,
2,
0,
2,

9,
9,
5,
5,
1,
5,
6,
0,
3,
4,
4,
6,
5,
4,
6,
5,
4,
5,
1,
4,
4,
7,
2,
3,
2,
7,
1,
8,
1,
8,
1,
8,
5,
0,
8,
9,
2,
5,
0,
1,
1,
1,
0,
9,
0,
3,
1,

```
6,
4,
2,
3,
6,
1,
1,
1,
3,
9,
5,
2,
9,
4,
5,
9,
3,
9,
0,
3,
6,
5,
5,
7,
2,
2,
7,
1,
2,
8,
4,
1,
7,
3,
3,
8,
8,
7,
9,
2,
2,
4,
1,
5,
9,
8,
7,
```

```
2,
3,
0,
6,
4,
2,
4,
1,
9,
5,
7,
7,
2,
8,
2,
0,
8,
5,
7,
7,
9,
1,
8,
1,
8,
0,
3,
0,
1,
9,
9,
4,
1,
8,
2,
1,
2,
9,
7,
5,
9,
2,
6,
4,
1,
5,
8,
```

```
2,
9,
2,
0,
4,
0,
0,
2,
8,
4,
7,
1,
2,
4,
0,
2,
7,
4,
3,
3,
0,
0,
3,
1,
9,
6,
5,
2,
5,
9,
7,
9,
3,
0,
4,
2,
0,
7,
1,
1,
2,
1,
5,
3,
3,
9,
7,
```

```
8,
6,
5,
6,
1,
3,
8,
1,
0,
5,
1,
3,
1,
5,
5,
6,
1,
8,
5,
1,
7,
9,
4,
6,
2,
2,
5,
0,
6,
5,
6,
3,
7,
2,
0,
8,
8,
5,
4,
1,
1,
4,
0,
3,
3,
7,
6,
```

1,
6,
2,
1,
9,
2,
8,
6,
1,
9,
5,
2,
5,
4,
4,
2,
8,
3,
8,
2,
4,
5,
0,
3,
1,
7,
7,
5,
7,
9,
7,
1,
9,
2,
1,
4,
2,
9,
2,
0,
4,
9,
1,
4,
8,
1,
8,

```
4,
5,
9,
8,
8,
3,
7,
6,
0,
0,
3,
0,
2,
0,
6,
9,
9,
3,
3,
3,
2,
3,
9,
1,
2,
6,
8,
0,
5,
6,
6,
6,
3,
8,
8,
2,
7,
5,
8,
9,
6,
1,
8,
4,
1,
2,
5,
```

```
9,
1,
9,
7,
5,
4,
0,
8,
9,
9,
1,
0,
5,
2,
3,
7,
2,
9,
4,
0,
6,
3,
9,
5,
2,
1,
3,
1,
3,
6,
5,
7,
4,
2,
2,
6,
3,
2,
6,
5,
4,
8,
9,
7,
1,
3,
0,
```

```
3,
8,
3,
1,
9,
3,
4,
4,
6,
4,
2,
1,
8,
2,
5,
4,
8,
5,
4,
0,
0,
2,
3,
2,
7,
7,
0,
8,
7,
4,
4,
7,
9,
6,
9,
0,
9,
8,
0,
4,
6,
0,
6,
3,
5,
4,
8,
```

```
3,
3,
9,
3,
3,
3,
7,
8,
0,
2,
2,
1,
7,
0,
6,
5,
4,
3,
8,
0,
9,
6,
3,
8,
0,
9,
9,
6,
8,
6,
8,
5,
7,
8,
6,
0,
2,
4,
0,
2,
2,
3,
1,
9,
7,
5,
8,
```

```
0,
8,
4,
6,
2,
6,
7,
9,
3,
2,
9,
8,
2,
2,
9,
2,
7,
3,
5,
9,
1,
8,
0,
2,
0,
5,
2,
1,
3,
7,
6,
7,
1,
2,
5,
8,
0,
3,
7,
8,
4,
0,
9,
1,
8,
6,
7,
```

```
7,
4,
3,
4,
9,
1,
9,
5,
1,
7,
3,
9,
7,
6,
9,
1,
3,
3,
8,
3,
3,
6,
7,
2,
4,
5,
8,
5,
1,
1,
4,
4,
3,
1,
0,
7,
7,
0,
7,
9,
9,
4,
8,
5,
5,
4,
0,
```

```
8,
2,
1,
0,
8,
4,
8,
0,
4,
0,
6,
1,
7,
3,
2,
6,
7,
2,
6,
9,
3,
1,
4,
6,
2,
5,
9,
2,
0,
6,
2,
1,
7,
3,
4,
1,
0,
5,
4,
3,
1,
1,
7,
4,
9,
9,
9,
```

```
8,
4,
0,
2,
4,
5,
1,
1,
6,
4,
7,
1,
9,
4,
2,
4,
1,
5,
5,
3,
8,
3,
1,
4,
5,
6,
8,
9,
4,
1,
5,
3,
8,
0,
3,
2,
5,
1,
2,
8,
3,
4,
4,
0,
8,
8,
3,
```

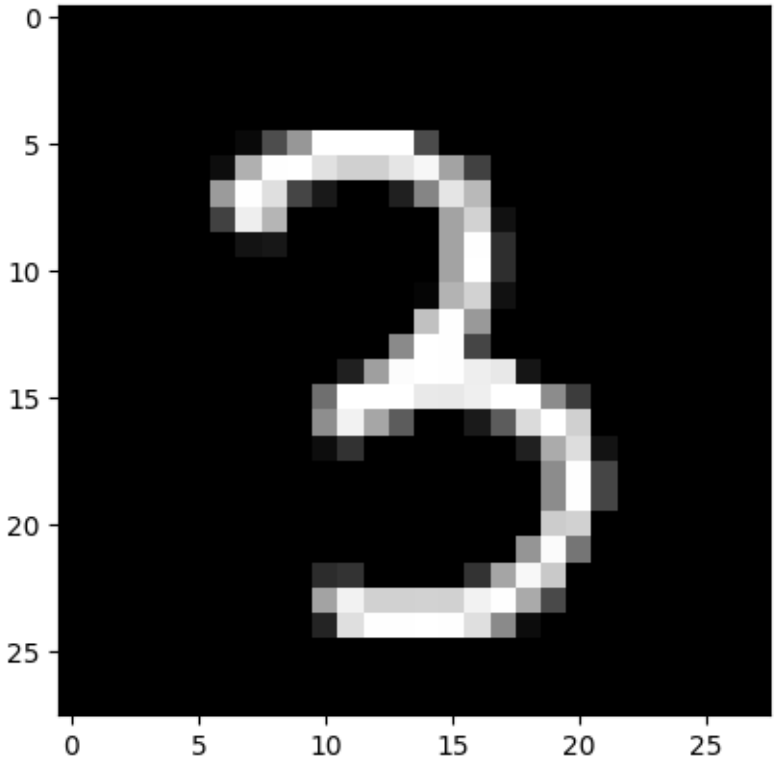
```
3,
1,
2,
3,
5,
9,
6,
3,
2,
6,
1,
3,
6,
0,
7,
2,
1,
7,
1,
4,
2,
4,
2,
1,
7,
9,
6,
1,
1,
2,
4,
3,
1,
7,
7,
4,
8,
0,
7,
3,
1,
3,
1,
0,
7,
7,
0,
```

```
3,
5,
5,
2,
7,
6,
6,
9,
2,
8,
3,
5,
2,
2,
5,
6,
0,
8,
2,
9,
2,
8,
8,
8,
8,
7,
9,
9,
3,
0,
6,
6,
3,
2,
1,
3,
2,
2,
9,
3,
0,
0,
5,
7,
8,
1,
4,
```

```
4,
6,
0,
2,
9,
1,
4,
7,
4,
7,
3,
9,
8,
8,
4,
7,
1,
2,
1,
2,
2,
3,
2,
3,
2,
3,
9,
1,
7,
4,
0,
3,
5,
5,
8,
6,
3,
2,
6,
7,
6,
6,
3,
2,
7,
8,
1,
```

1,
7,
5,
6,
4,
9,
5,
1,
3,
3,
9,
7,
8,
9,
1,
1,
0,
9,
1,
4,
4,
5,
4,
0,
6,
2,
2,
3,
1,
5,
1,
2,
0,
3,
8,
1,
2,
6,
7,
1,
6,
2,
3,
9,
0,
1,
2,

```
 2,
 0,
 8,
 9,
 …]
```

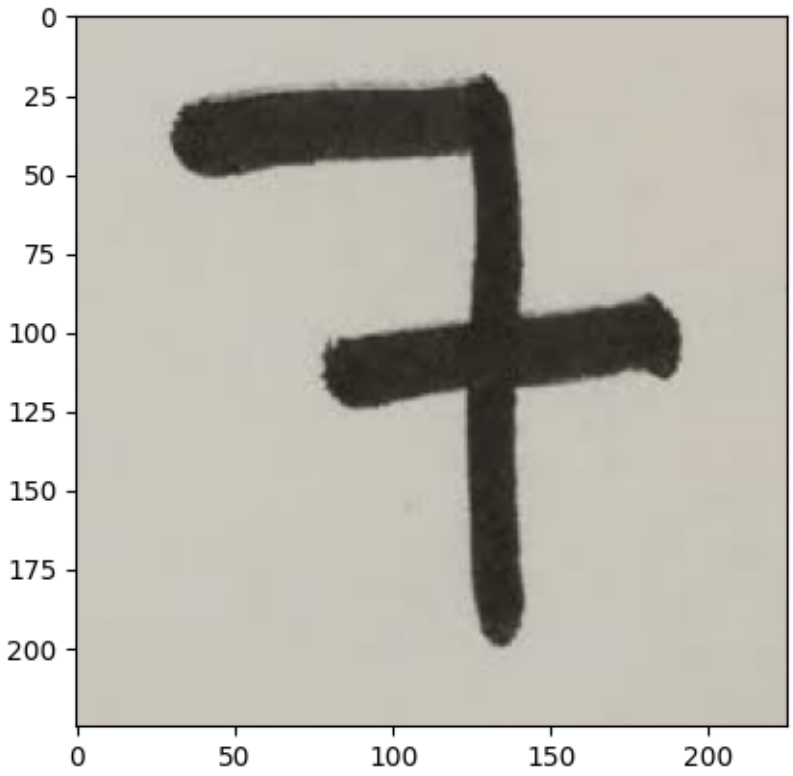


[ ]:
```
import matplotlib.pyplot as plt
import matplotlib.image as im
import cv2
img = im.imread('IMG1.jpg')
print(img)
plt.imshow(img)
img.dtype
img=cv2.resize(img, (28,28), interpolation=cv2.INTER_AREA)
img.shape
```

```
[[[203 199 188]
  [203 199 188]
  [203 199 188]
  …
  [204 197 187]
  [204 197 187]
```

```
 [204 198 186]]

[[203 199 188]
 [203 199 188]
 [203 199 188]
 …
 [204 197 187]
 [204 197 187]
 [204 198 186]]

[[203 199 188]
 [203 199 188]
 [203 199 188]
 …
 [204 197 187]
 [204 197 187]
 [203 197 185]]

…

[[203 199 188]
 [203 199 188]
 [203 199 188]
 …
 [198 194 183]
 [198 194 183]
 [198 194 183]]

[[203 199 188]
 [203 199 188]
 [203 199 188]
 …
 [198 194 183]
 [198 194 183]
 [198 194 183]]

[[203 199 188]
 [203 199 188]
 [203 199 188]
 …
 [200 193 183]
 [200 193 183]
 [198 194 183]]]
```

[ ]: (28, 28, 3)



[ ]:
```
class_names=['0', '1', '2', '3', '4',
'5', '6', '7', '8', '9']
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, pred)
```

[ ]:
```
array([[ 974,    0,    0,    0,    0,    1,    1,    2,    2,    0],
       [   0, 1120,    3,    1,    0,    0,    2,    2,    7,    0],
       [   7,    1, 1002,    1,    2,    0,    3,    8,    8,    0],
       [   0,    0,    0,  990,    0,    8,    0,    4,    3,    5],
       [   1,    0,    2,    0,  943,    0,    4,    3,    3,   26],
       [   2,    0,    0,    2,    0,  882,    2,    1,    2,    1],
       [   7,    2,    0,    1,    2,    4,  939,    0,    3,    0],
       [   2,    3,    9,    5,    0,    0,    0, 1003,    0,    6],
       [   2,    0,    3,    6,    4,    4,    0,    4,  949,    2],
       [   3,    2,    0,    7,    5,    3,    1,    4,    1,  983]],
      dtype=int64)
```

[ ]:
```
import sklearn
from sklearn.metrics import classification_report
print(classification_report(y_test, pred, target_names=class_names))
```

```
              precision    recall  f1-score   support
```

```
              0       0.98      0.99      0.98       980
              1       0.99      0.99      0.99      1135
              2       0.98      0.97      0.98      1032
              3       0.98      0.98      0.98      1010
              4       0.99      0.96      0.97       982
              5       0.98      0.99      0.98       892
              6       0.99      0.98      0.98       958
              7       0.97      0.98      0.97      1028
              8       0.97      0.97      0.97       974
              9       0.96      0.97      0.97      1009

    accuracy                           0.98     10000
   macro avg       0.98      0.98      0.98     10000
weighted avg       0.98      0.98      0.98     10000
```

[ ]:
```
print('Accuracy :', sklearn.metrics.accuracy_score(y_test, pred))
print('F1 score :', sklearn.metrics.f1_score(y_test, pred,
average='weighted'))
print('Recall :', sklearn.metrics.recall_score(y_test, pred,
average='weighted'))
print('Precision :', sklearn.metrics.precision_score(y_test, pred,
average='weighted'))
```

```
Accuracy : 0.9785
F1 score : 0.9784989698978481
Recall : 0.9785
Precision : 0.9785768189074296
```

[ ]:
```

```