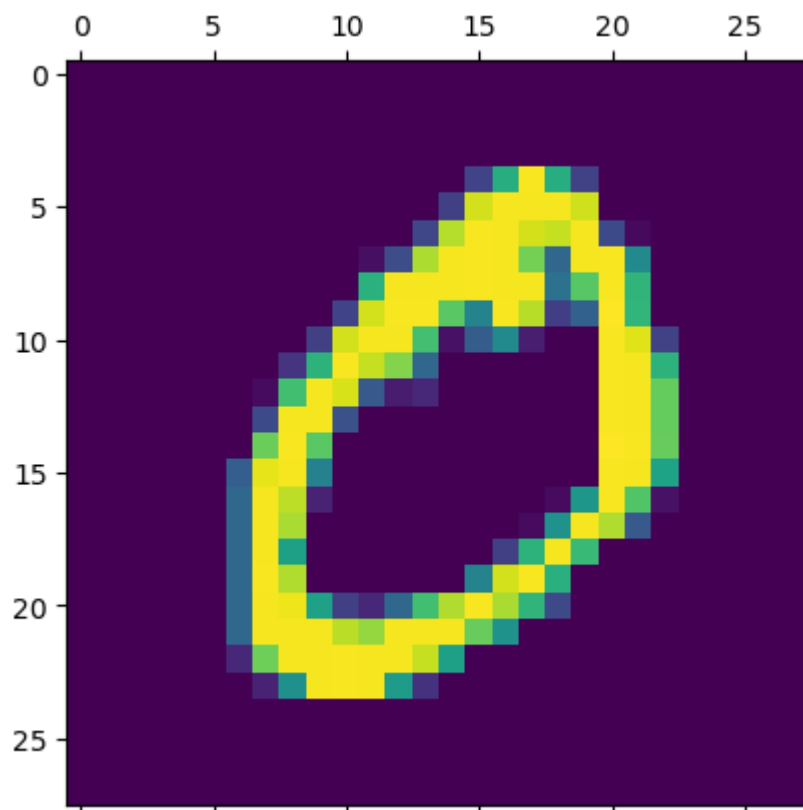


```
In [8]: ▶ #importing necessary packages
import tensorflow as tf
from tensorflow import keras
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import random
%matplotlib inline
```

```
In [9]: ▶ #import dataset and split into train and test data
mnist = tf.keras.datasets.mnist
(x_train, y_train), (x_test, y_test) = mnist.load_data()
```

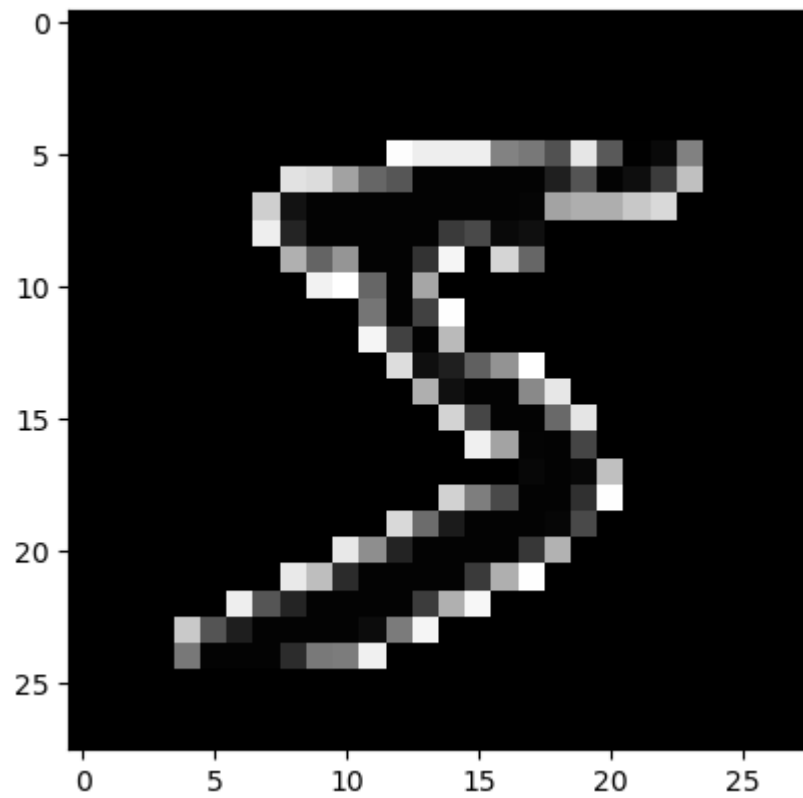
```
In [10]: ▶ plt.matshow(x_train[1])
```

Out[10]: <matplotlib.image.AxesImage at 0x24730311090>



```
In [11]: ▶ plt.imshow(-x_train[0], cmap="gray")
```

```
Out[11]: <matplotlib.image.AxesImage at 0x24730370650>
```



```
In [12]: ▶ x_train = x_train / 255  
x_test = x_test / 255
```

```
In [13]: #define network architecture using keras
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(128, activation="relu"),
    keras.layers.Dense(10, activation="softmax")
])
model.summary()
```

WARNING:tensorflow:From C:\Users\shrey\AppData\Roaming\Python\Python311\site-packages\keras\src\backend.py:873: The name tf.get\_default\_graph is deprecated. Please use tf.compat.v1.get\_default\_graph instead.

Model: "sequential"

Layer (type)	Output Shape	Param #
flatten (Flatten)	(None, 784)	0
dense (Dense)	(None, 128)	100480
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)		

```
In [14]: #train model using sgd
model.compile(optimizer="sgd",
    loss="sparse_categorical_crossentropy",
    metrics=['accuracy'])
```

WARNING:tensorflow:From C:\Users\shrey\AppData\Roaming\Python\Python311\site-packages\keras\src\optimizers\\_\_init\_\_.py:309: The name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.

```
In [15]: history=model.fit(x_train,
y_train,validation_data=(x_test,y_test),epochs=10)
```

Epoch 1/10

WARNING:tensorflow:From C:\Users\shrey\AppData\Roaming\Python\Python311\site-packages\keras\src\utils\tf\_utils.py:492: The name tf.ragged.RaggedTensorValue is deprecated. Please use tf.compat.v1.ragged.RaggedTensorValue instead.

WARNING:tensorflow:From C:\Users\shrey\AppData\Roaming\Python\Python311\site-packages\keras\src\engine\base\_layer\_utils.py:384: The name tf.executing\_eagerly\_outside\_functions is deprecated. Please use tf.compat.v1.executing\_eagerly\_outside\_functions instead.

1875/1875 [=====] - 8s 4ms/step - loss: 0.6486 - accuracy: 0.8342 - val\_loss: 0.3616 - val\_accuracy: 0.9028

Epoch 2/10

1875/1875 [=====] - 5s 3ms/step - loss: 0.3394 - accuracy: 0.9050 - val\_loss: 0.2959 - val\_accuracy: 0.9163

Epoch 3/10

1875/1875 [=====] - 6s 3ms/step - loss: 0.2904 - accuracy: 0.9182 - val\_loss: 0.2635 - val\_accuracy: 0.9257

Epoch 4/10

1875/1875 [=====] - 5s 3ms/step - loss: 0.2591 - accuracy: 0.9269 - val\_loss: 0.2404 - val\_accuracy: 0.9329

Epoch 5/10

1875/1875 [=====] - 5s 3ms/step - loss: 0.2354 - accuracy: 0.9341 - val\_loss: 0.2201 - val\_accuracy: 0.9351

Epoch 6/10

1875/1875 [=====] - 6s 3ms/step - loss: 0.2165 - accuracy: 0.9391 - val\_loss: 0.2038 - val\_accuracy: 0.9417

Epoch 7/10

1875/1875 [=====] - 6s 3ms/step - loss: 0.2007 - accuracy: 0.9436 - val\_loss: 0.1922 - val\_accuracy: 0.9432

Epoch 8/10

1875/1875 [=====] - 5s 3ms/step - loss: 0.1871 - accuracy: 0.9475 - val\_loss: 0.1806 - val\_accuracy: 0.9471

Epoch 9/10

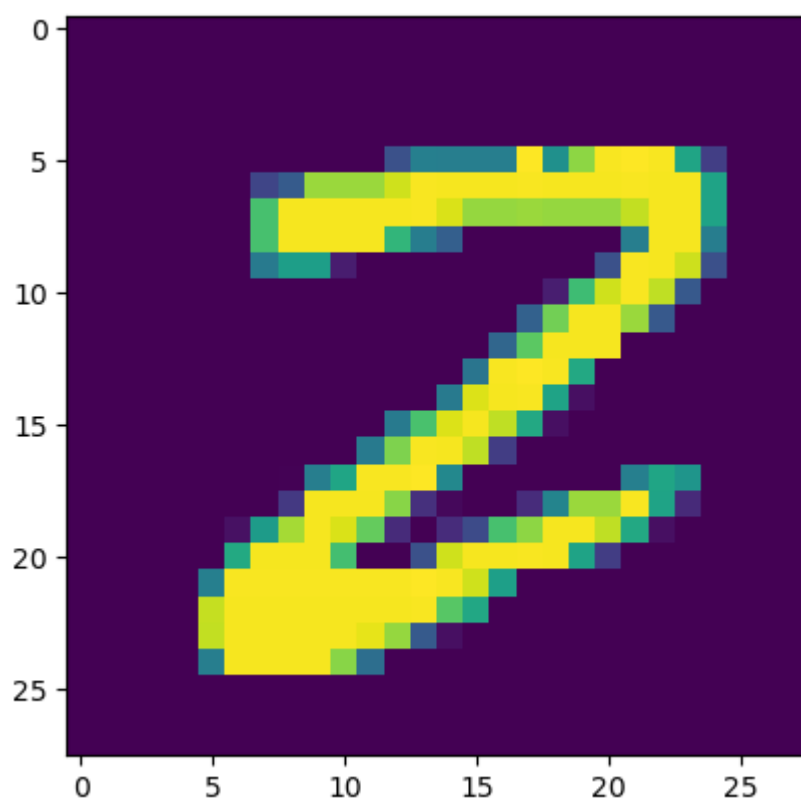
1875/1875 [=====] - 6s 3ms/step - loss: 0.1757 - accuracy: 0.9510 - val\_loss: 0.1694 - val\_accuracy: 0.9490

Epoch 10/10

1875/1875 [=====] - 5s 3ms/step - loss: 0.1651 - accuracy: 0.9536 - val\_loss: 0.1616 - val\_accuracy: 0.9528

```
In [ ]: #evaluate the network
test_loss,test_acc=model.evaluate(x_test,y_test)
print("Loss=%.3f" %test_loss)
print("Accuracy=%.3f" %test_acc)
```

```
In [17]: n=random.randint(0,9999)
plt.imshow(x_test[n])
plt.show()
```

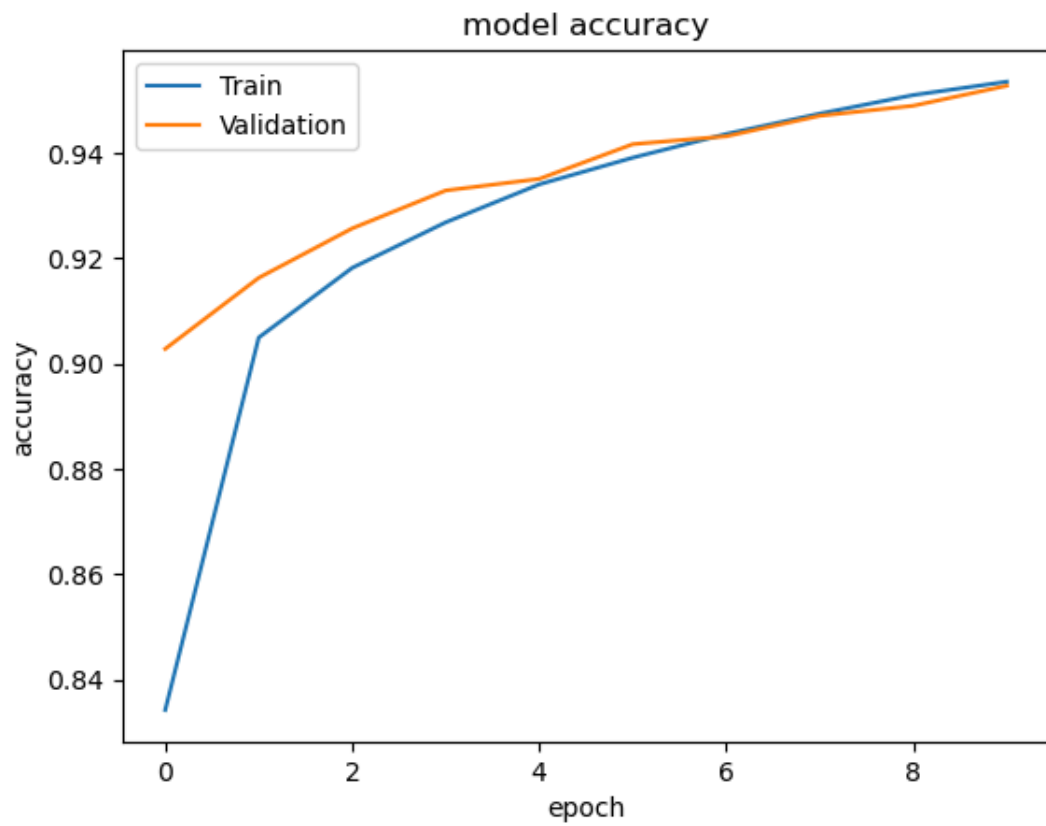


```
In [18]: predicted_value=model.predict(x_test)
print("Handwritten number in the image is=%d" %np.argmax(predicted_valu
```



```
313/313 [=====] - 1s 2ms/step
Handwritten number in the image is=2
```

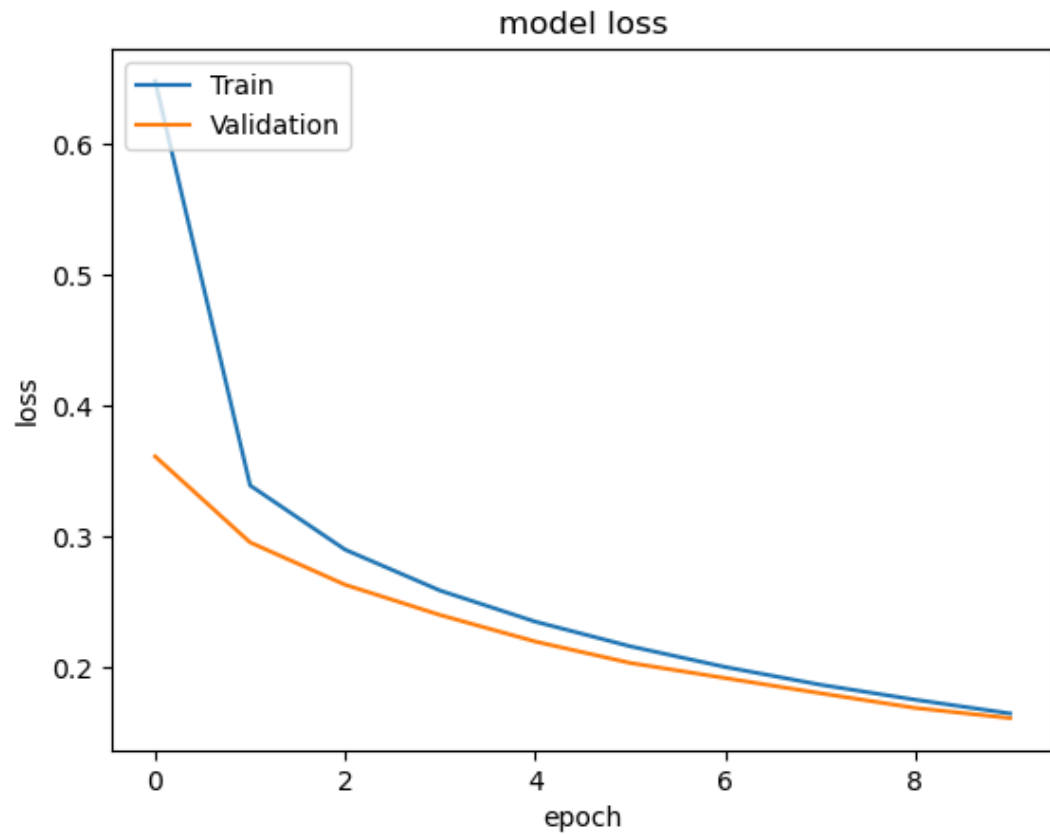
```
In [29]: # history.history()
history.history.keys()
# dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()
```



In [30]:

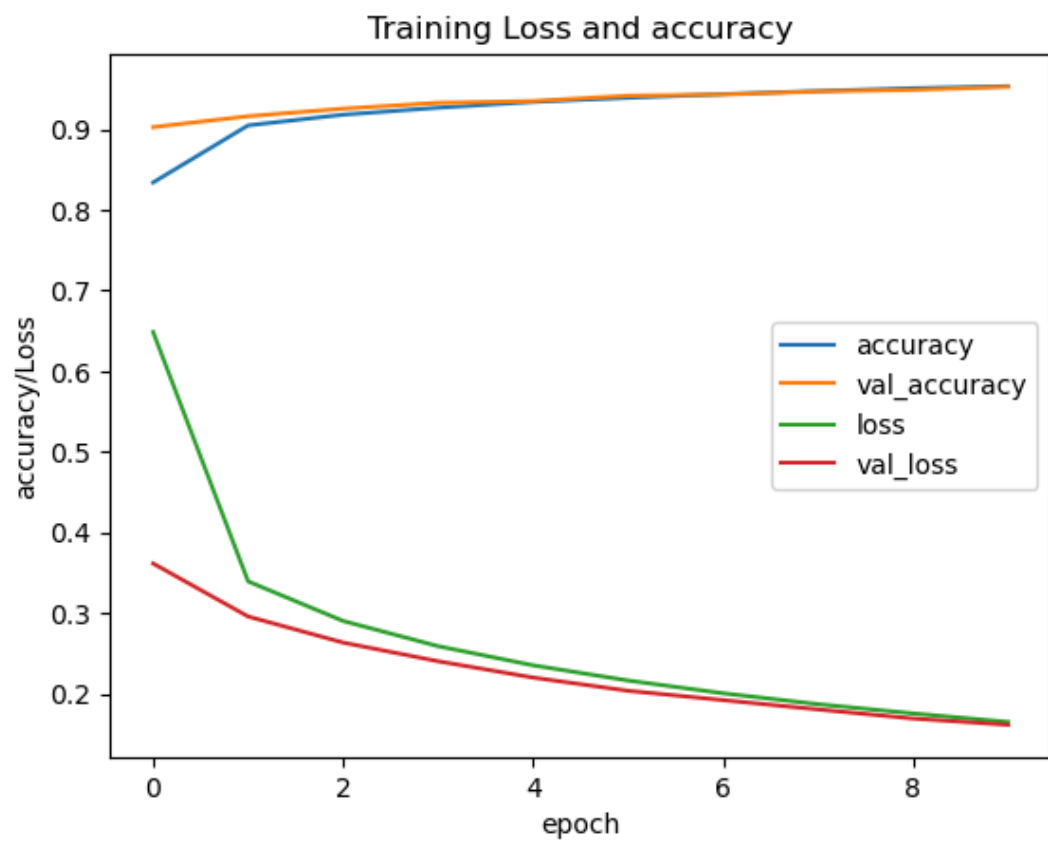
```
# history.history()

history.history.keys()
# dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()
```



```
In [34]: ▶ plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Training Loss and accuracy')
plt.ylabel('accuracy/Loss')
plt.xlabel('epoch')
plt.legend(['accuracy', 'val_accuracy', 'loss', 'val_loss'])
```

Out[34]: <matplotlib.legend.Legend at 0x2474f4d3a90>



In [ ]: ▶

In [ ]: ▶