NNGurmukhi Handwritten Digit Classification

April 29, 2023

[2]:
```
import os
import cv2
import numpy as np
from google.colab import drive
from matplotlib import pyplot as plt
import pandas as pd
from sklearn.metrics import confusion_matrix
import seaborn as sns
```

[3]:
```
# Mount your Google Drive
drive.mount('/content/drive')

# Define the path to the train and test data directories
train_path = "/content/drive/MyDrive/dataset/train/train"

# Get the list of all folders in the train directory
folders = os.listdir(train_path)

# Define the image size and the number of channels
img_size = 28
num_channels = 1

# Define the empty lists to store the images and their respective labels
x_train = []
y_train = []

# Loop through each folder and read the images
for folder in folders:
    # Define the path to the current folder
    folder_path = os.path.join(train_path, folder)
    # Get the list of all image files in the current folder
    files = os.listdir(folder_path)
    # Loop through each image and read it
    for file in files:
        # Define the path to the current image
        img_path = os.path.join(folder_path, file)
        # Read the image and resize it to the defined size
```

```
        img = cv2.imread(img_path,cv2.IMREAD_GRAYSCALE)
        img = cv2.resize(img, (img_size, img_size))
        x_train.append(img.flatten() / 255)
        # Append the image and its respective label to the lists
        #x_train.append(img)
        y_train.append(int(folder))

# Convert the lists to numpy arrays
x_train = np.array(x_train)
y_train = np.array(y_train)

# Print the shape of the training data and target labels
print("Training data shape:", x_train.shape)
print("Training labels shape:", y_train.shape)
```

```
Mounted at /content/drive
Training data shape: (1000, 784)
Training labels shape: (1000,)
```

[4]:
```
# Define the path to the train and test data directories
test_path = "/content/drive/MyDrive/dataset/val/val"

# Get the list of all folders in the train directory
folders = os.listdir(test_path)

# Define the image size and the number of channels
img_size = 28
num_channels = 1

# Define the empty lists to store the images and their respective labels
x_test = []
y_test = []

# Loop through each folder and read the images
for folder in folders:
    # Define the path to the current folder
    folder_path = os.path.join(test_path, folder)
    # Get the list of all image files in the current folder
    files = os.listdir(folder_path)
    # Loop through each image and read it
    for file in files:
        # Define the path to the current image
        img_path = os.path.join(folder_path, file)
        # Read the image and resize it to the defined size
        img = cv2.imread(img_path,cv2.IMREAD_GRAYSCALE)
        img = cv2.resize(img, (img_size, img_size))
        x_test.append(img.flatten() / 255)
```

```
        y_test.append(int(folder))

# Convert the lists to numpy arrays
x_test = np.array(x_test)
y_test = np.array(y_test)

# Print the shape of the training data and target labels
print("Training data shape:", x_test.shape)
print("Training labels shape:", y_test.shape)
```

```
Training data shape: (178, 784)
Training labels shape: (178,)
```

[5]:
```
import numpy as np
# create a list of column names
# create a list of column names for the Xtrain data
column_names = ["pixel" + str(i) for i in range(784)]

# stack the Ytrain and Xtrain arrays horizontally
combined_data = np.column_stack((y_train, x_train))

# save the combined data to a CSV file
np.savetxt("train.csv", combined_data, delimiter=",", header="label," + ",".
 ↪join(column_names), comments="")
```

[6]:
```
import numpy as np
# create a list of column names for the Xtrain data
column_names = ["pixel" + str(i) for i in range(784)]

# stack the Ytrain and Xtrain arrays horizontally
combined_data = np.column_stack((y_test, x_test))

# save the combined data to a CSV file
np.savetxt("test.csv", combined_data, delimiter=",", header="label," + ",".
 ↪join(column_names), comments="")
```

[7]:
```
data = pd.read_csv('train.csv')
data['label'] = data['label'].astype(int)
data.head()
```

[7]:

	label	pixel0	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	\
0	5	0.996078	0.929412	0.929412	0.462745	0.0	0.000000	0.000000	
1	5	0.996078	0.729412	0.000000	0.000000	0.0	0.000000	0.862745	
2	5	0.929412	0.729412	0.000000	0.000000	0.0	0.729412	0.929412	
3	5	0.862745	0.000000	0.000000	0.000000	0.0	0.729412	0.929412	
4	5	1.000000	0.984314	0.929412	0.462745	0.0	0.000000	0.000000	

```
    pixel7    pixel8  …  pixel774  pixel775  pixel776  pixel777  pixel778  \
0 0.066667  0.945098  …       1.0       1.0       1.0       0.0  0.200000
1 0.933333  1.000000  …       1.0       1.0       1.0       1.0  1.000000
2 1.000000  1.000000  …       1.0       1.0       1.0       1.0  1.000000
3 1.000000  1.000000  …       1.0       1.0       1.0       1.0  0.984314
4 0.000000  0.200000  …       1.0       1.0       1.0       1.0  1.000000

  pixel779  pixel780  pixel781  pixel782  pixel783
0 0.952941  1.000000  1.000000  1.000000  1.000000
1 0.972549  0.462745  0.000000  0.784314  1.000000
2 1.000000  1.000000  1.000000  0.215686  0.000000
3 0.596078  0.000000  0.596078  0.984314  1.000000
4 1.000000  0.964706  0.333333  0.000000  0.862745

[5 rows x 785 columns]
```

[8]:
```python
testdata = pd.read_csv('test.csv')
testdata['label'] = testdata['label'].astype(int)
testdata.head(178)
```

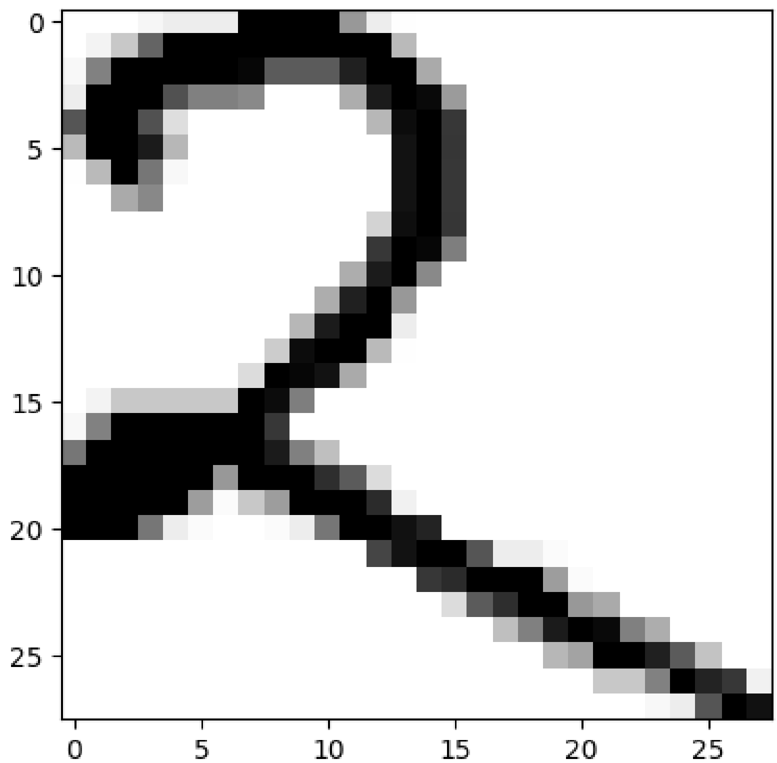
[8]:
```
     label  pixel0  pixel1  pixel2  pixel3  pixel4  pixel5  pixel6    pixel7  \
0        9     1.0     1.0     1.0     1.0     1.0     1.0     1.0  1.000000
1        9     1.0     1.0     1.0     1.0     1.0     1.0     1.0  1.000000
2        9     1.0     1.0     1.0     1.0     1.0     1.0     1.0  1.000000
3        9     1.0     1.0     1.0     1.0     1.0     1.0     1.0  1.000000
4        9     1.0     1.0     1.0     1.0     1.0     1.0     1.0  1.000000
..     …       …       …       …       …       …       …       …         …
173      0     1.0     1.0     1.0     1.0     1.0     1.0     1.0  1.000000
174      0     1.0     1.0     1.0     1.0     1.0     1.0     1.0  0.862745
175      0     1.0     1.0     1.0     1.0     1.0     1.0     1.0  1.000000
176      0     1.0     1.0     1.0     1.0     1.0     1.0     1.0  1.000000
177      0     1.0     1.0     1.0     1.0     1.0     1.0     1.0  0.996078

       pixel8  …  pixel774  pixel775  pixel776  pixel777  pixel778  \
0    1.000000  …  1.000000  1.000000  1.000000  1.000000  1.000000
1    1.000000  …  1.000000  0.945098  0.929412  0.929412  0.929412
2    1.000000  …  0.000000  0.000000  0.929412  0.000000  0.000000
3    1.000000  …  0.070588  0.070588  0.070588  0.070588  0.054902
4    1.000000  …  0.929412  0.200000  0.000000  0.000000  0.000000
..        …  …       …         …         …         …         …
173  1.000000  …  1.000000  1.000000  1.000000  1.000000  1.000000
174  0.000000  …  0.929412  0.200000  0.000000  0.066667  0.929412
175  1.000000  …  0.972549  1.000000  1.000000  1.000000  1.000000
176  1.000000  …  1.000000  1.000000  1.000000  1.000000  1.000000
177  0.929412  …  0.929412  0.929412  0.996078  1.000000  1.000000

     pixel779  pixel780  pixel781  pixel782  pixel783
```

```
0    1.000000  0.498039  0.000000  0.784314  1.000000
1    0.929412  0.462745  0.000000  0.000000  0.000000
2    0.000000  0.000000  0.596078  0.984314  1.000000
3    0.000000  0.462745  0.929412  0.929412  0.929412
4    0.000000  0.000000  0.000000  0.000000  0.000000
..        …         …         …         …         …
173  1.000000  1.000000  1.000000  1.000000  1.000000
174  0.952941  1.000000  1.000000  1.000000  1.000000
175  1.000000  1.000000  1.000000  1.000000  1.000000
176  1.000000  1.000000  1.000000  1.000000  1.000000
177  1.000000  1.000000  1.000000  1.000000  1.000000

[178 rows x 785 columns]
```

[9]:

```python
data = np.array(data)
m, n = data.shape
np.random.shuffle(data)

data_train = data[0:m].T
Y_train = data_train[0].astype(int)
X_train = data_train[1:n]
_,m_train = X_train.shape

label = Y_train[1]
print(label)
current_image = X_train[:, 1, None]
current_image = current_image.reshape((28, 28)) * 255
plt.gray()
plt.imshow(current_image, interpolation='nearest')
plt.show()
```

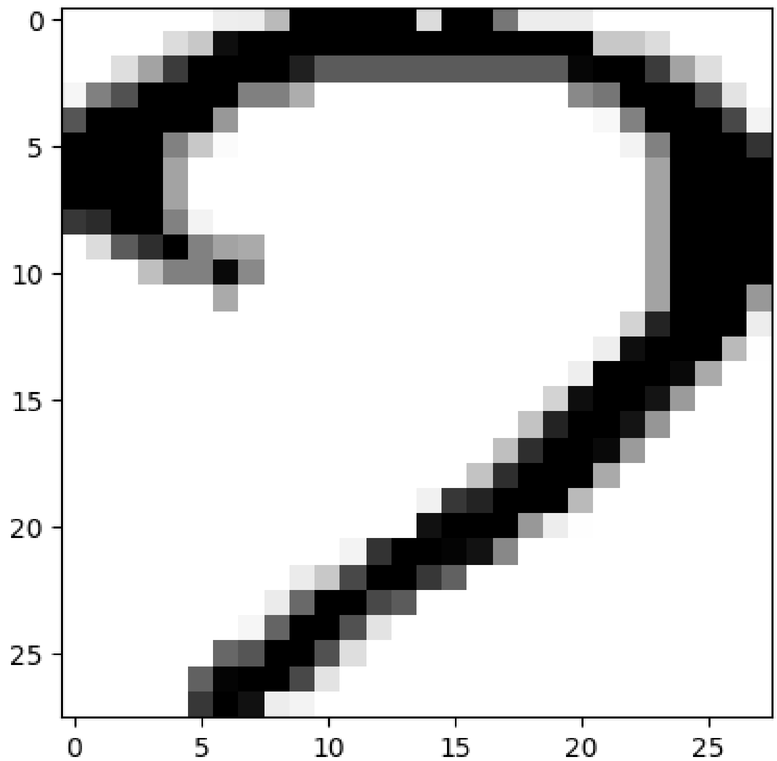
```
2
```



[10]:

```
testdata = np.array(testdata)
m, n = testdata.shape
np.random.shuffle(testdata)


data_dev = testdata[0:m].T
Y_dev = data_dev[0].astype(int)
X_dev = data_dev[1:n]


label = Y_dev[1]
print(label)
current_image = X_dev[:, 1, None]
current_image = current_image.reshape((28, 28)) * 255
plt.gray()
plt.imshow(current_image, interpolation='nearest')
plt.show()
```

7



[11]:
```
def init_params():
    W1 = np.random.rand(10, 784) - 0.5
    b1 = np.random.rand(10, 1) - 0.5
    W2 = np.random.rand(10, 10) - 0.5
    b2 = np.random.rand(10, 1) - 0.5
    return W1, b1, W2, b2

def ReLU(Z):
    return np.maximum(Z, 0)

def softmax(Z):
    A = np.exp(Z) / sum(np.exp(Z))
    return A

def forward_prop(W1, b1, W2, b2, X):
    Z1 = W1.dot(X) + b1
    A1 = ReLU(Z1)
    Z2 = W2.dot(A1) + b2
    A2 = softmax(Z2)
    return Z1, A1, Z2, A2
def ReLU_deriv(Z):
    return Z > 0
```

```
def one_hot(Y):
    one_hot_Y = np.zeros((Y.size, Y.max() + 1))
    one_hot_Y[np.arange(Y.size), Y] = 1
    one_hot_Y = one_hot_Y.T
    return one_hot_Y

def backward_prop(Z1, A1, Z2, A2, W1, W2, X, Y):
    one_hot_Y = one_hot(Y)
    dZ2 = A2 - one_hot_Y
    dW2 = 1 / m * dZ2.dot(A1.T)
    db2 = 1 / m * np.sum(dZ2)
    dZ1 = W2.T.dot(dZ2) * ReLU_deriv(Z1)
    dW1 = 1 / m * dZ1.dot(X.T)
    db1 = 1 / m * np.sum(dZ1)
    return dW1, db1, dW2, db2

def update_params(W1, b1, W2, b2, dW1, db1, dW2, db2, alpha):
    W1 = W1 - alpha * dW1
    b1 = b1 - alpha * db1
    W2 = W2 - alpha * dW2
    b2 = b2 - alpha * db2
    return W1, b1, W2, b2
```

[12]:
```
def get_predictions(A2):
    print("#"*100)
    return np.argmax(A2, 0)

def get_accuracy(predictions, Y):
    return np.sum(predictions == Y) / Y.size

def gradient_descent(X, Y, alpha, iterations):
    W1, b1, W2, b2 = init_params()
    for i in range(iterations):
        Z1, A1, Z2, A2 = forward_prop(W1, b1, W2, b2, X)
        dW1, db1, dW2, db2 = backward_prop(Z1, A1, Z2, A2, W1, W2, X, Y)
        W1, b1, W2, b2 = update_params(W1, b1, W2, b2, dW1, db1, dW2, db2,␣
↪alpha)
        if i % 10 == 0:
            print("Iteration: ", i)
            predictions = get_predictions(A2)
            print("ACCURACY -----> ",get_accuracy(predictions, Y),"\n")
    return W1, b1, W2, b2
```

[13]:
```
W1, b1, W2, b2 = gradient_descent(X_train, Y_train, 0.01,   3000)
print("Final weights","\n")
print("w1 : ",W1,"\n")
```

```python
print("w2 : ",W2,"\n")
print("Final bias","\n")
print("b1 : ",b1,"\n")
print("b2 : ",b2,"\n")
```

```
Iteration:  0
################################################################################
####################
ACCURACY ----->  0.109

Iteration:  10
################################################################################
####################
ACCURACY ----->  0.14

Iteration:  20
################################################################################
####################
ACCURACY ----->  0.219

Iteration:  30
################################################################################
####################
ACCURACY ----->  0.399

Iteration:  40
################################################################################
####################
ACCURACY ----->  0.489

Iteration:  50
################################################################################
####################
ACCURACY ----->  0.542

Iteration:  60
################################################################################
####################
ACCURACY ----->  0.577

Iteration:  70
################################################################################
####################
ACCURACY ----->  0.599

Iteration:  80
################################################################################
```

```
####################
ACCURACY ----->  0.643

Iteration:  90
################################################################################
####################
ACCURACY ----->  0.692

Iteration:  100
################################################################################
####################
ACCURACY ----->  0.723

Iteration:  110
################################################################################
####################
ACCURACY ----->  0.75

Iteration:  120
################################################################################
####################
ACCURACY ----->  0.784

Iteration:  130
################################################################################
####################
ACCURACY ----->  0.812

Iteration:  140
################################################################################
####################
ACCURACY ----->  0.825

Iteration:  150
################################################################################
####################
ACCURACY ----->  0.841

Iteration:  160
################################################################################
####################
ACCURACY ----->  0.85

Iteration:  170
################################################################################
####################
ACCURACY ----->  0.863
```

```
Iteration:  180
################################################################################
####################
ACCURACY ----->  0.874

Iteration:  190
################################################################################
####################
ACCURACY ----->  0.879

Iteration:  200
################################################################################
####################
ACCURACY ----->  0.884

Iteration:  210
################################################################################
####################
ACCURACY ----->  0.888

Iteration:  220
################################################################################
####################
ACCURACY ----->  0.889

Iteration:  230
################################################################################
####################
ACCURACY ----->  0.894

Iteration:  240
################################################################################
####################
ACCURACY ----->  0.898

Iteration:  250
################################################################################
####################
ACCURACY ----->  0.899

Iteration:  260
################################################################################
####################
ACCURACY ----->  0.904

Iteration:  270
################################################################################
####################
```

```
ACCURACY ----->  0.908

Iteration:  280
################################################################################
####################
ACCURACY ----->  0.913

Iteration:  290
################################################################################
####################
ACCURACY ----->  0.916

Iteration:  300
################################################################################
####################
ACCURACY ----->  0.917

Iteration:  310
################################################################################
####################
ACCURACY ----->  0.919

Iteration:  320
################################################################################
####################
ACCURACY ----->  0.923

Iteration:  330
################################################################################
####################
ACCURACY ----->  0.927

Iteration:  340
################################################################################
####################
ACCURACY ----->  0.929

Iteration:  350
################################################################################
####################
ACCURACY ----->  0.931

Iteration:  360
################################################################################
####################
ACCURACY ----->  0.935

Iteration:  370
```

```
################################################################################
####################
ACCURACY ----->  0.937

Iteration:  380
################################################################################
####################
ACCURACY ----->  0.938

Iteration:  390
################################################################################
####################
ACCURACY ----->  0.939

Iteration:  400
################################################################################
####################
ACCURACY ----->  0.943

Iteration:  410
################################################################################
####################
ACCURACY ----->  0.944

Iteration:  420
################################################################################
####################
ACCURACY ----->  0.944

Iteration:  430
################################################################################
####################
ACCURACY ----->  0.944

Iteration:  440
################################################################################
####################
ACCURACY ----->  0.945

Iteration:  450
################################################################################
####################
ACCURACY ----->  0.947

Iteration:  460
################################################################################
####################
ACCURACY ----->  0.949
```

```
Iteration:  470
################################################################################
####################
ACCURACY ----->  0.95

Iteration:  480
################################################################################
####################
ACCURACY ----->  0.95

Iteration:  490
################################################################################
####################
ACCURACY ----->  0.951

Iteration:  500
################################################################################
####################
ACCURACY ----->  0.953

Iteration:  510
################################################################################
####################
ACCURACY ----->  0.953

Iteration:  520
################################################################################
####################
ACCURACY ----->  0.954

Iteration:  530
################################################################################
####################
ACCURACY ----->  0.955

Iteration:  540
################################################################################
####################
ACCURACY ----->  0.955

Iteration:  550
################################################################################
####################
ACCURACY ----->  0.957

Iteration:  560
################################################################################
```

```
####################
ACCURACY ----->  0.957

Iteration:  570
################################################################################
####################
ACCURACY ----->  0.957

Iteration:  580
################################################################################
####################
ACCURACY ----->  0.958

Iteration:  590
################################################################################
####################
ACCURACY ----->  0.959

Iteration:  600
################################################################################
####################
ACCURACY ----->  0.961

Iteration:  610
################################################################################
####################
ACCURACY ----->  0.962

Iteration:  620
################################################################################
####################
ACCURACY ----->  0.962

Iteration:  630
################################################################################
####################
ACCURACY ----->  0.962

Iteration:  640
################################################################################
####################
ACCURACY ----->  0.963

Iteration:  650
################################################################################
####################
ACCURACY ----->  0.964
```

```
Iteration:  660
################################################################################
####################
ACCURACY ----->  0.965

Iteration:  670
################################################################################
####################
ACCURACY ----->  0.965

Iteration:  680
################################################################################
####################
ACCURACY ----->  0.967

Iteration:  690
################################################################################
####################
ACCURACY ----->  0.968

Iteration:  700
################################################################################
####################
ACCURACY ----->  0.968

Iteration:  710
################################################################################
####################
ACCURACY ----->  0.969

Iteration:  720
################################################################################
####################
ACCURACY ----->  0.97

Iteration:  730
################################################################################
####################
ACCURACY ----->  0.971

Iteration:  740
################################################################################
####################
ACCURACY ----->  0.971

Iteration:  750
################################################################################
####################
```

```
ACCURACY ----->  0.973

Iteration:  760
################################################################################
####################
ACCURACY ----->  0.973

Iteration:  770
################################################################################
####################
ACCURACY ----->  0.974

Iteration:  780
################################################################################
####################
ACCURACY ----->  0.974

Iteration:  790
################################################################################
####################
ACCURACY ----->  0.975

Iteration:  800
################################################################################
####################
ACCURACY ----->  0.976

Iteration:  810
################################################################################
####################
ACCURACY ----->  0.977

Iteration:  820
################################################################################
####################
ACCURACY ----->  0.977

Iteration:  830
################################################################################
####################
ACCURACY ----->  0.977

Iteration:  840
################################################################################
####################
ACCURACY ----->  0.978

Iteration:  850
```

```
################################################################################
####################
ACCURACY ----->  0.98

Iteration:  860
################################################################################
####################
ACCURACY ----->  0.982

Iteration:  870
################################################################################
####################
ACCURACY ----->  0.984

Iteration:  880
################################################################################
####################
ACCURACY ----->  0.986

Iteration:  890
################################################################################
####################
ACCURACY ----->  0.986

Iteration:  900
################################################################################
####################
ACCURACY ----->  0.986

Iteration:  910
################################################################################
####################
ACCURACY ----->  0.986

Iteration:  920
################################################################################
####################
ACCURACY ----->  0.986

Iteration:  930
################################################################################
####################
ACCURACY ----->  0.986

Iteration:  940
################################################################################
####################
ACCURACY ----->  0.986
```

```
Iteration:  950
################################################################################
####################
ACCURACY ----->  0.987

Iteration:  960
################################################################################
####################
ACCURACY ----->  0.989

Iteration:  970
################################################################################
####################
ACCURACY ----->  0.989

Iteration:  980
################################################################################
####################
ACCURACY ----->  0.989

Iteration:  990
################################################################################
####################
ACCURACY ----->  0.989

Iteration:  1000
################################################################################
####################
ACCURACY ----->  0.991

Iteration:  1010
################################################################################
####################
ACCURACY ----->  0.991

Iteration:  1020
################################################################################
####################
ACCURACY ----->  0.991

Iteration:  1030
################################################################################
####################
ACCURACY ----->  0.991

Iteration:  1040
################################################################################
```

```
####################
ACCURACY ----->  0.991

Iteration:  1050
################################################################################
####################
ACCURACY ----->  0.993

Iteration:  1060
################################################################################
####################
ACCURACY ----->  0.993

Iteration:  1070
################################################################################
####################
ACCURACY ----->  0.995

Iteration:  1080
################################################################################
####################
ACCURACY ----->  0.995

Iteration:  1090
################################################################################
####################
ACCURACY ----->  0.995

Iteration:  1100
################################################################################
####################
ACCURACY ----->  0.995

Iteration:  1110
################################################################################
####################
ACCURACY ----->  0.995

Iteration:  1120
################################################################################
####################
ACCURACY ----->  0.995

Iteration:  1130
################################################################################
####################
ACCURACY ----->  0.995
```

```
Iteration:  1140
################################################################################
####################
ACCURACY ----->  0.996

Iteration:  1150
################################################################################
####################
ACCURACY ----->  0.996

Iteration:  1160
################################################################################
####################
ACCURACY ----->  0.996

Iteration:  1170
################################################################################
####################
ACCURACY ----->  0.996

Iteration:  1180
################################################################################
####################
ACCURACY ----->  0.996

Iteration:  1190
################################################################################
####################
ACCURACY ----->  0.996

Iteration:  1200
################################################################################
####################
ACCURACY ----->  0.996

Iteration:  1210
################################################################################
####################
ACCURACY ----->  0.996

Iteration:  1220
################################################################################
####################
ACCURACY ----->  0.996

Iteration:  1230
################################################################################
####################
```

```
ACCURACY ----->  0.996

Iteration:  1240
################################################################################
####################
ACCURACY ----->  0.996

Iteration:  1250
################################################################################
####################
ACCURACY ----->  0.996

Iteration:  1260
################################################################################
####################
ACCURACY ----->  0.997

Iteration:  1270
################################################################################
####################
ACCURACY ----->  0.997

Iteration:  1280
################################################################################
####################
ACCURACY ----->  0.997

Iteration:  1290
################################################################################
####################
ACCURACY ----->  0.997

Iteration:  1300
################################################################################
####################
ACCURACY ----->  0.998

Iteration:  1310
################################################################################
####################
ACCURACY ----->  0.998

Iteration:  1320
################################################################################
####################
ACCURACY ----->  0.998

Iteration:  1330
```

```
################################################################################
####################
ACCURACY ----->  0.998

Iteration:  1340
################################################################################
####################
ACCURACY ----->  0.998

Iteration:  1350
################################################################################
####################
ACCURACY ----->  0.998

Iteration:  1360
################################################################################
####################
ACCURACY ----->  0.998

Iteration:  1370
################################################################################
####################
ACCURACY ----->  0.998

Iteration:  1380
################################################################################
####################
ACCURACY ----->  0.998

Iteration:  1390
################################################################################
####################
ACCURACY ----->  0.998

Iteration:  1400
################################################################################
####################
ACCURACY ----->  0.998

Iteration:  1410
################################################################################
####################
ACCURACY ----->  0.998

Iteration:  1420
################################################################################
####################
ACCURACY ----->  0.998
```

```
Iteration:  1430
################################################################################
####################
ACCURACY ----->  0.998

Iteration:  1440
################################################################################
####################
ACCURACY ----->  0.998

Iteration:  1450
################################################################################
####################
ACCURACY ----->  0.998

Iteration:  1460
################################################################################
####################
ACCURACY ----->  0.998

Iteration:  1470
################################################################################
####################
ACCURACY ----->  0.998

Iteration:  1480
################################################################################
####################
ACCURACY ----->  0.998

Iteration:  1490
################################################################################
####################
ACCURACY ----->  0.998

Iteration:  1500
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1510
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1520
################################################################################
```

```
####################
ACCURACY ----->  0.999

Iteration:  1530
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1540
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1550
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1560
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1570
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1580
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1590
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1600
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1610
################################################################################
####################
ACCURACY ----->  0.999
```

```
Iteration:  1620
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1630
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1640
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1650
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1660
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1670
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1680
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1690
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1700
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1710
################################################################################
####################
```

```
ACCURACY ----->  0.999

Iteration:  1720
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1730
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1740
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1750
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1760
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1770
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1780
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1790
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1800
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1810
```

```
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1820
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1830
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1840
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1850
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1860
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1870
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1880
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1890
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1900
################################################################################
####################
ACCURACY ----->  0.999
```

```
Iteration:  1910
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1920
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1930
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1940
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1950
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1960
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1970
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1980
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  1990
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  2000
################################################################################
```

```
####################
ACCURACY ----->  0.999

Iteration:  2010
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  2020
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  2030
################################################################################
####################
ACCURACY ----->  0.999

Iteration:  2040
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2050
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2060
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2070
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2080
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2090
################################################################################
####################
ACCURACY ----->  1.0
```

```
Iteration:  2100
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2110
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2120
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2130
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2140
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2150
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2160
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2170
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2180
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2190
################################################################################
####################
```

```
ACCURACY ----->  1.0

Iteration:  2200
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2210
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2220
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2230
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2240
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2250
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2260
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2270
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2280
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2290
```

```
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2300
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2310
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2320
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2330
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2340
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2350
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2360
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2370
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2380
################################################################################
####################
ACCURACY ----->  1.0
```

```
Iteration:  2390
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2400
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2410
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2420
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2430
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2440
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2450
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2460
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2470
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2480
################################################################################
```

```
####################
ACCURACY ----->  1.0

Iteration:  2490
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2500
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2510
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2520
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2530
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2540
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2550
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2560
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2570
################################################################################
####################
ACCURACY ----->  1.0
```

```
Iteration:  2580
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2590
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2600
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2610
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2620
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2630
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2640
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2650
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2660
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2670
################################################################################
####################
```

```
ACCURACY ----->  1.0

Iteration:  2680
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2690
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2700
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2710
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2720
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2730
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2740
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2750
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2760
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2770
```

```
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2780
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2790
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2800
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2810
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2820
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2830
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2840
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2850
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2860
################################################################################
####################
ACCURACY ----->  1.0
```

```
Iteration:  2870
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2880
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2890
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2900
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2910
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2920
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2930
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2940
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2950
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2960
################################################################################
```

```
####################
ACCURACY ----->  1.0

Iteration:  2970
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2980
################################################################################
####################
ACCURACY ----->  1.0

Iteration:  2990
################################################################################
####################
ACCURACY ----->  1.0

Final weights

w1 :  [[ 0.03157671  0.09150234  0.16384779 …  0.36008017 -0.22055458
  -0.0214013 ]
 [-0.17483829 -0.29715748  0.05446306 …  0.23132973  0.38213701
   0.39398468]
 [-0.33946228 -0.20704644  0.26437539 … -0.06204053 -0.11186042
  -0.55739576]
 …
 [ 0.22955531 -0.34764684 -0.39713152 … -0.10243345 -0.29766212
   0.11012308]
 [ 0.30734613 -0.04641468 -0.08323898 …  0.27633018 -0.35511033
   0.19380217]
 [-0.28970803  0.03726549  0.19520246 …  0.24060071  0.29178854
  -0.20479002]]

w2 :  [[-0.54629158 -0.25173185 -0.54886856  0.48663787 -0.06732886 -0.7815455
   0.63894462 -0.5215383  -0.13305639 -0.35471676]
 [-0.3940902   1.39783049 -0.71853494  0.50014784 -0.23701339  0.2817396
  -0.42653296 -0.31525903 -0.16531905  0.36168918]
 [-0.07847176 -0.23698494  1.09495916  0.34153637  0.24302988 -0.06740911
  -0.63456708 -1.24728759 -0.65606869 -0.04396559]
 [ 0.42302272  0.48840761  0.76835509 -0.03299167  0.15148522 -0.7049692
  -0.98460713  0.29125591  0.538783   -0.27182699]
 [-0.1685696  -0.61436901  0.34052812 -0.4475098   0.21033658  0.85422315
   0.12733159 -0.48555097  0.37390613 -0.38980334]
 [ 0.30264791 -0.57166     0.64274798 -1.4090898  -0.37603213  0.2957559
  -0.56735164  0.6088053  -0.47283069  0.04744121]
 [-0.12890302 -0.53292168  0.3323005   0.17390274 -0.47030478 -0.96686872
   0.15035677  0.40235812 -0.3427039   0.14263146]
```

```
 [-0.13236111  0.42403264  0.18293707 -0.12124835 -0.11841466  0.3163107
   0.37729766 -0.72157846 -0.15262949 -0.48119942]
 [ 0.03190978 -0.58116046 -1.16604016  0.24519975  0.2584575   0.32470274
  -0.41084061  0.66284945 -0.23597826 -0.01026353]
 [ 0.02116192  0.21126859 -1.12224133 -0.46551947 -0.24535618 -0.48942691
   0.597237    0.87029328 -0.4639276   0.21857362]]

Final bias

b1 :  [[ 0.35721738]
 [ 0.05209061]
 [-0.3329564 ]
 [-0.46155889]
 [-0.2404678 ]
 [ 0.04900926]
 [ 0.42975876]
 [ 0.00716055]
 [ 0.01652929]
 [ 0.06328138]]

b2 :  [[-0.37314731]
 [ 0.29774897]
 [-0.18568368]
 [ 0.45882918]
 [ 0.07766576]
 [ 0.18839537]
 [-0.24435732]
 [-0.46381676]
 [ 0.04581421]
 [ 0.21786476]]
```

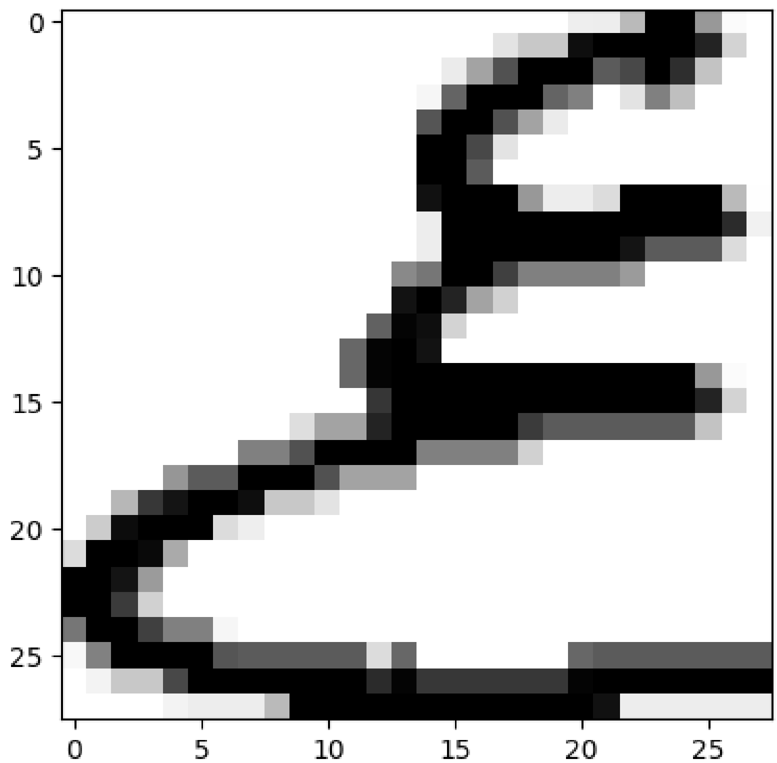
[14]:

```python
def make_predictions(X, W1, b1, W2, b2):
    _, _, _,A2  = forward_prop(W1, b1, W2, b2, X)
    predictions = get_predictions(A2)
    return predictions

def test_prediction(index, W1, b1, W2, b2):
    current_image = X_train[:, index, None]
    prediction = make_predictions(X_train[:, index, None], W1, b1, W2, b2)
    label = Y_train[index]
    print("Prediction: ", prediction)
    print("Label: ", label)

    current_image = current_image.reshape((28, 28)) * 255
    plt.gray()
    plt.imshow(current_image, interpolation='nearest')
```

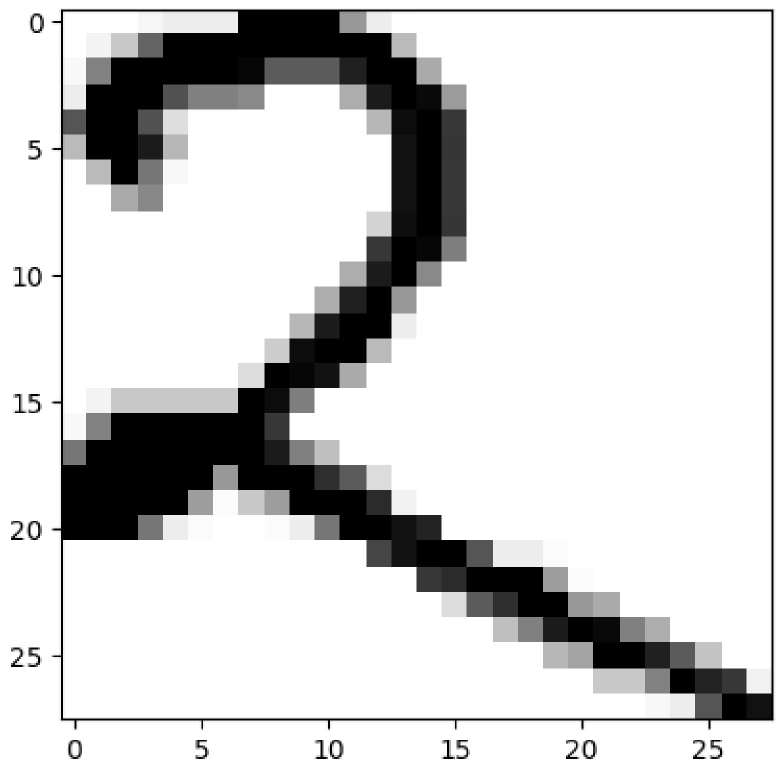
```
    plt.show()

def test_prediction1(index, W1, b1, W2, b2):
    current_image = X_dev[:, index, None]
    prediction = make_predictions(X_dev[:, index, None], W1, b1, W2, b2)
    label = Y_dev[index]
    print("Prediction: ", prediction)
    print("Label: ", label)

    current_image = current_image.reshape((28, 28)) * 255
    plt.gray()
    plt.imshow(current_image, interpolation='nearest')
    plt.show()
```

[15]:
```
test_prediction(0, W1, b1, W2, b2)
test_prediction(1, W1, b1, W2, b2)
test_prediction(2, W1, b1, W2, b2)
test_prediction(3, W1, b1, W2, b2)

test_prediction1(0, W1, b1, W2, b2)
test_prediction1(1, W1, b1, W2, b2)
test_prediction1(2, W1, b1, W2, b2)
test_prediction1(3, W1, b1, W2, b2)
```

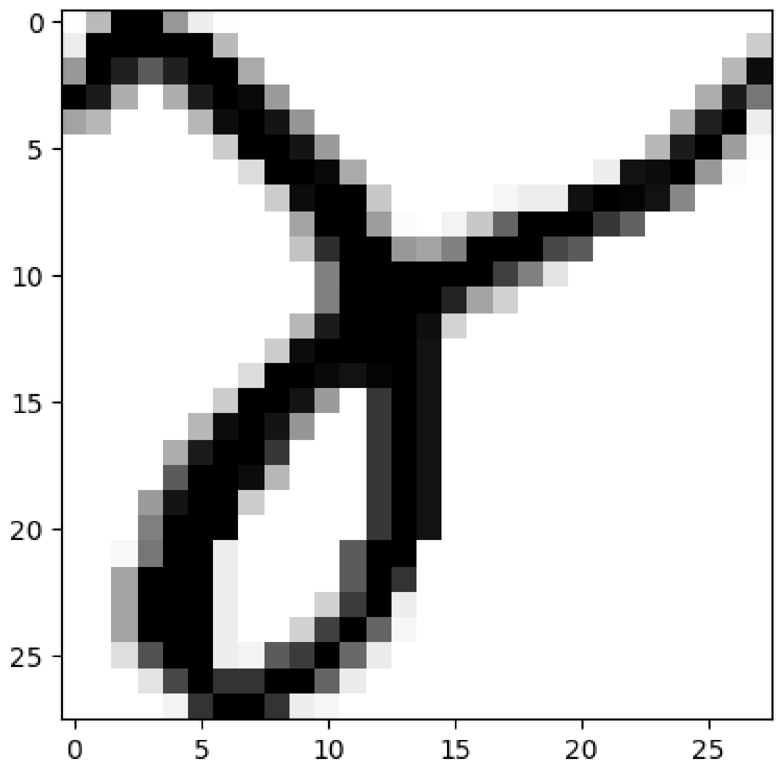
```
################################################################################
####################
Prediction:  [6]
Label:  6
```



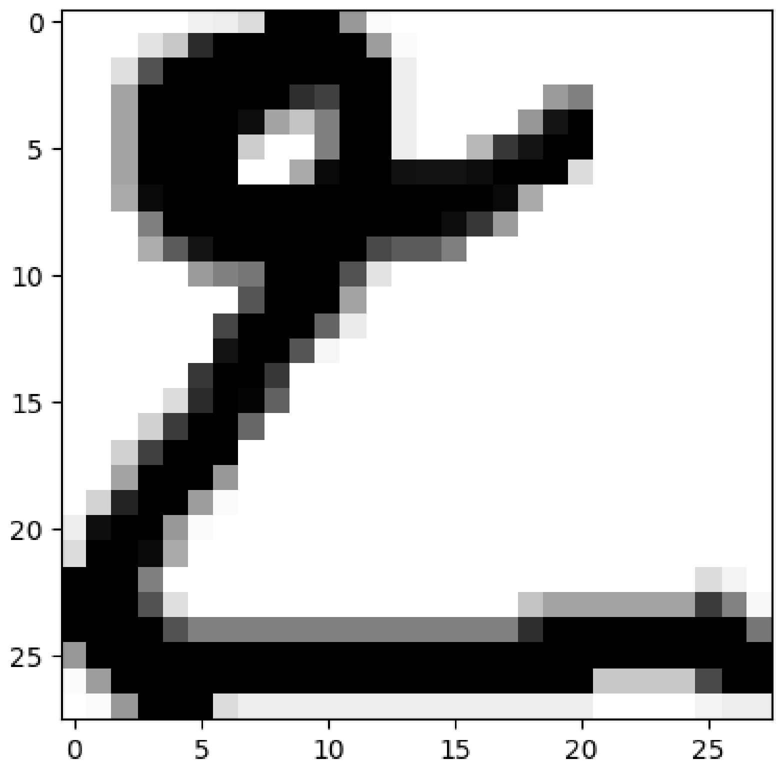
##
####################
Prediction: [2]
Label: 2



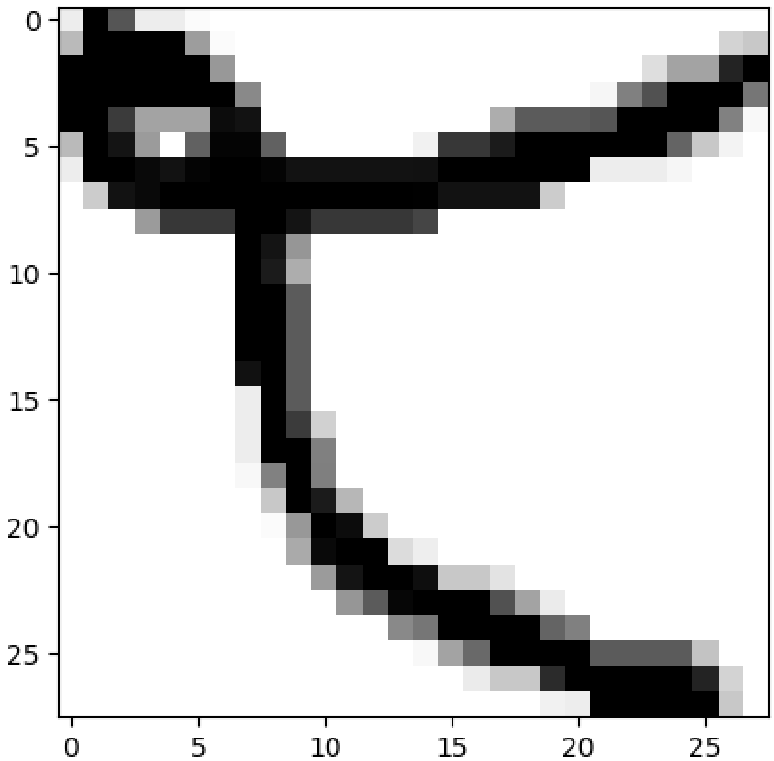

```
################################################################################
####################
Prediction:  [4]
Label:  4
```



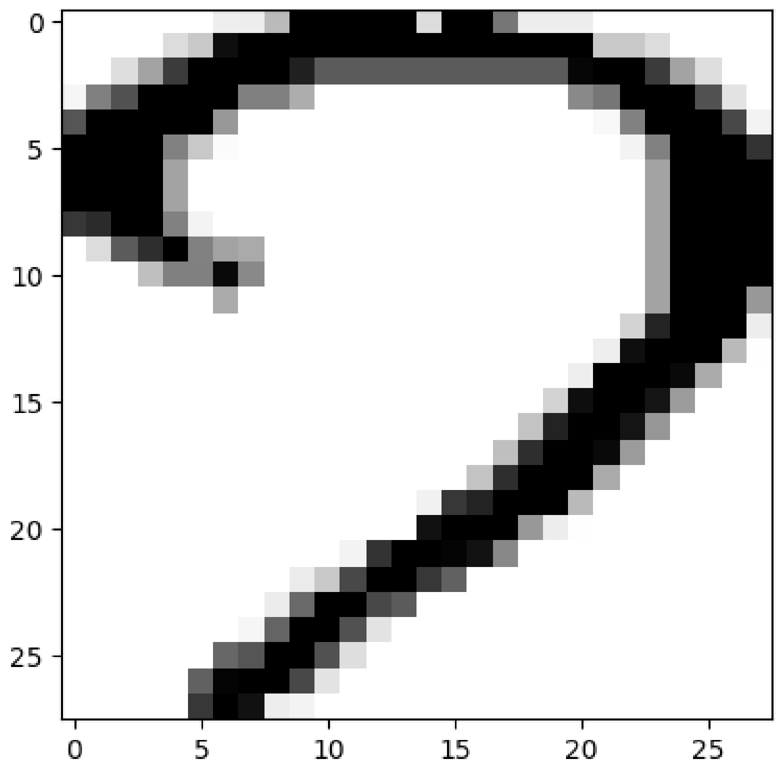

```
################################################################################
####################
Prediction:  [8]
Label:  8
```



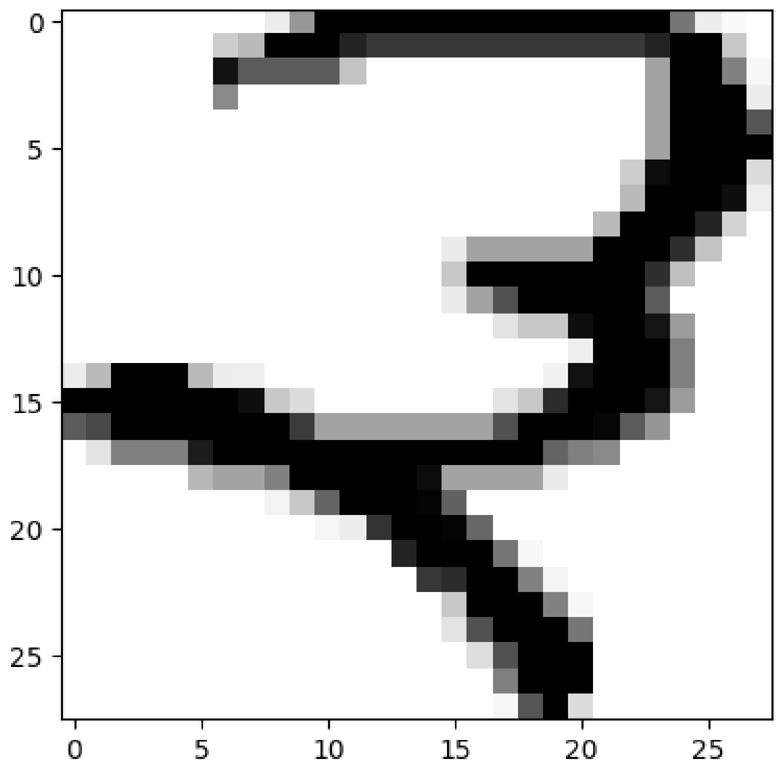

```
################################################################################
####################
Prediction:  [8]
Label:  8
```



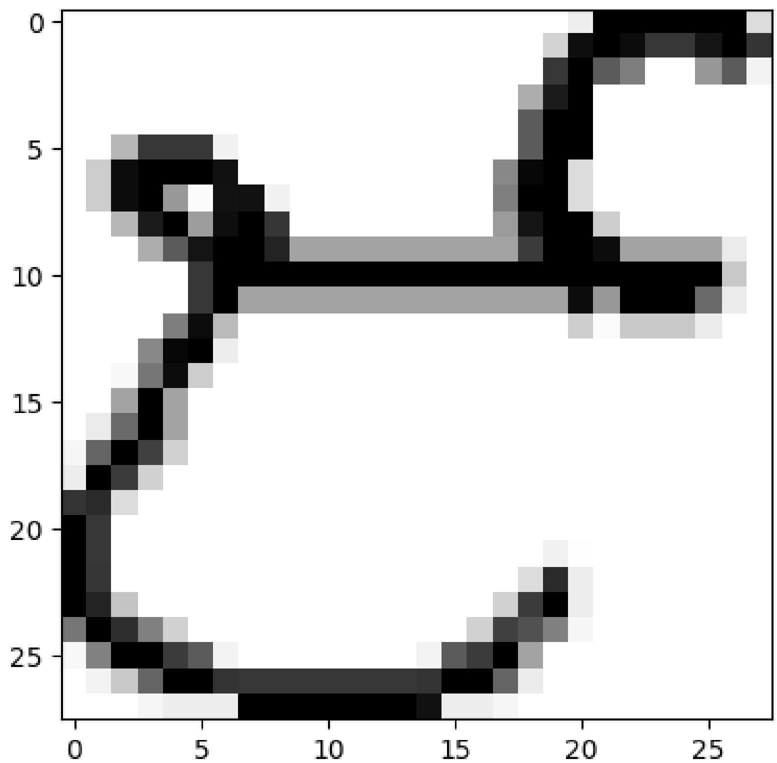

```
####################################################################################
####################
Prediction:  [7]
Label:  7
```




```
################################################################################
####################
Prediction:  [3]
Label:  3
```



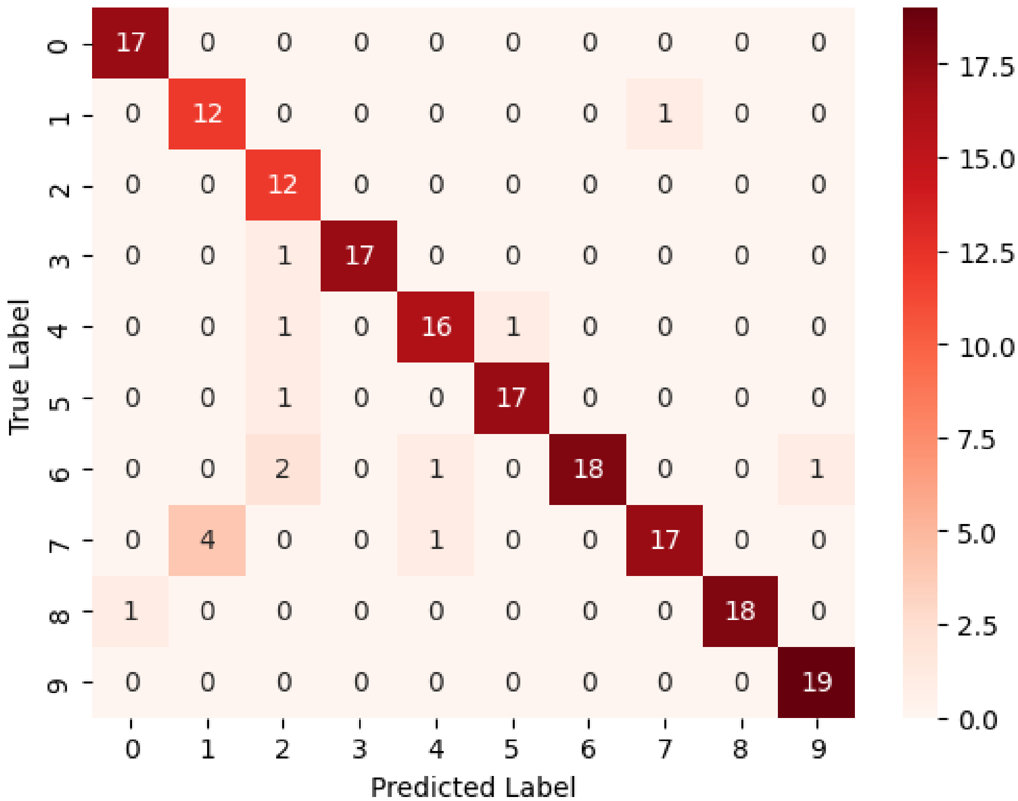
##
####################
Prediction: [9]
Label: 9



[16]:
```
dev_predictions = make_predictions(X_dev, W1, b1, W2, b2)
print("Accuracy",get_accuracy(dev_predictions, Y_dev))
# Generate confusion matrix and heatmap
conf_matrix = confusion_matrix(dev_predictions, Y_dev)
print("conf_matrix : ","\n")
print(conf_matrix,"\n")
sns.heatmap(conf_matrix, annot=True, cmap="Reds", fmt="d")
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.show()
```

```
################################################################################
####################
Accuracy 0.9157303370786517
conf_matrix :

[[17  0  0  0  0  0  0  0  0  0]
 [ 0 12  0  0  0  0  0  1  0  0]
 [ 0  0 12  0  0  0  0  0  0  0]
 [ 0  0  1 17  0  0  0  0  0  0]
 [ 0  0  1  0 16  1  0  0  0  0]
 [ 0  0  1  0  0 17  0  0  0  0]
```

```
[ 0  0  2  0  1  0 18  0  0  1]
[ 0  4  0  0  1  0  0 17  0  0]
[ 1  0  0  0  0  0  0  0 18  0]
[ 0  0  0  0  0  0  0  0  0 19]]
```



[17]:
```python
import pandas as pd
import numpy as np
import tensorflow as tf
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
import matplotlib.pyplot as plt

# Load the data from CSV files
train_data = pd.read_csv("train.csv")
test_data = pd.read_csv("test.csv")

# Split the data into input features (pixels) and target variable (labels)
X_train = train_data.iloc[:, 1:].values.astype('float32') / 255.0
y_train = train_data.iloc[:, 0].values.astype('int32')
```

```python
X_test = test_data.iloc[:, 1:].values.astype('float32') / 255.0
y_test = test_data.iloc[:, 0].values.astype('int32')

# Split the training data into train and validation sets
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=0.2, random_state=42)

# Define the model architecture
model = Sequential([
    Dense(512, activation='relu', input_shape=(784,)),
    Dropout(0.2),
    Dense(256, activation='relu'),
    Dropout(0.2),
    Dense(10, activation='softmax')
])

# Compile the model
model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

# Train the model
history = model.fit(X_train, y_train, epochs=200, batch_size=32, validation_data=(X_val, y_val))

# Evaluate the model on the test set
test_loss, test_acc = model.evaluate(X_test, y_test)

print('Test loss:', test_loss)
print('Test accuracy:', test_acc)

# Make predictions on the test set
predictions = model.predict(X_test)
y_pred = np.argmax(predictions, axis=1)

# Display a random image from the test set along with its predicted and actual labels
random_index = np.random.randint(0, len(X_test))
plt.imshow(X_test[random_index].reshape((28, 28)), cmap='gray')
plt.title("Predicted label: " + str(np.argmax(predictions[random_index])) + ", Actual label: " + str(y_test[random_index]))
plt.show()
random_index = np.random.randint(1, len(X_test))
plt.imshow(X_test[random_index].reshape((28, 28)), cmap='gray')
plt.title("Predicted label: " + str(np.argmax(predictions[random_index])) + ", Actual label: " + str(y_test[random_index]))
```

```
plt.show()
random_index = np.random.randint(3, len(X_test))
plt.imshow(X_test[random_index].reshape((28, 28)), cmap='gray')
plt.title("Predicted label: " + str(np.argmax(predictions[random_index])) + ",␣
 ↪Actual label: " + str(y_test[random_index]))
plt.show()
random_index = np.random.randint(4, len(X_test))
plt.imshow(X_test[random_index].reshape((28, 28)), cmap='gray')
plt.title("Predicted label: " + str(np.argmax(predictions[random_index])) + ",␣
 ↪Actual label: " + str(y_test[random_index]))
plt.show()
```

```
Epoch 1/200
25/25 [==============================] - 2s 23ms/step - loss: 2.2850 - accuracy:
0.1800 - val_loss: 2.2579 - val_accuracy: 0.2250
Epoch 2/200
25/25 [==============================] - 0s 12ms/step - loss: 2.1844 - accuracy:
0.3925 - val_loss: 2.0841 - val_accuracy: 0.5250
Epoch 3/200
25/25 [==============================] - 0s 11ms/step - loss: 1.8911 - accuracy:
0.5450 - val_loss: 1.6882 - val_accuracy: 0.5800
Epoch 4/200
25/25 [==============================] - 0s 12ms/step - loss: 1.4154 - accuracy:
0.6212 - val_loss: 1.2106 - val_accuracy: 0.7600
Epoch 5/200
25/25 [==============================] - 0s 11ms/step - loss: 1.0105 - accuracy:
0.7350 - val_loss: 0.8821 - val_accuracy: 0.7450
Epoch 6/200
25/25 [==============================] - 0s 12ms/step - loss: 0.7903 - accuracy:
0.7575 - val_loss: 0.6707 - val_accuracy: 0.8400
Epoch 7/200
25/25 [==============================] - 0s 13ms/step - loss: 0.6165 - accuracy:
0.8087 - val_loss: 0.5486 - val_accuracy: 0.8600
Epoch 8/200
25/25 [==============================] - 0s 12ms/step - loss: 0.5188 - accuracy:
0.8487 - val_loss: 0.4309 - val_accuracy: 0.9100
Epoch 9/200
25/25 [==============================] - 0s 11ms/step - loss: 0.4236 - accuracy:
0.8750 - val_loss: 0.3620 - val_accuracy: 0.9200
Epoch 10/200
25/25 [==============================] - 0s 11ms/step - loss: 0.3752 - accuracy:
0.8938 - val_loss: 0.3338 - val_accuracy: 0.9150
Epoch 11/200
25/25 [==============================] - 0s 12ms/step - loss: 0.3280 - accuracy:
0.9187 - val_loss: 0.2767 - val_accuracy: 0.9350
Epoch 12/200
25/25 [==============================] - 0s 11ms/step - loss: 0.2760 - accuracy:
```

```
0.9275 - val_loss: 0.2396 - val_accuracy: 0.9350
Epoch 13/200
25/25 [==============================] - 0s 12ms/step - loss: 0.2514 - accuracy:
0.9413 - val_loss: 0.2318 - val_accuracy: 0.9200
Epoch 14/200
25/25 [==============================] - 0s 12ms/step - loss: 0.2244 - accuracy:
0.9337 - val_loss: 0.1952 - val_accuracy: 0.9400
Epoch 15/200
25/25 [==============================] - 0s 13ms/step - loss: 0.1998 - accuracy:
0.9450 - val_loss: 0.1892 - val_accuracy: 0.9400
Epoch 16/200
25/25 [==============================] - 0s 13ms/step - loss: 0.1785 - accuracy:
0.9488 - val_loss: 0.1686 - val_accuracy: 0.9450
Epoch 17/200
25/25 [==============================] - 0s 11ms/step - loss: 0.1651 - accuracy:
0.9525 - val_loss: 0.1833 - val_accuracy: 0.9450
Epoch 18/200
25/25 [==============================] - 0s 13ms/step - loss: 0.1551 - accuracy:
0.9563 - val_loss: 0.1721 - val_accuracy: 0.9300
Epoch 19/200
25/25 [==============================] - 0s 11ms/step - loss: 0.1484 - accuracy:
0.9575 - val_loss: 0.1793 - val_accuracy: 0.9200
Epoch 20/200
25/25 [==============================] - 0s 12ms/step - loss: 0.1345 - accuracy:
0.9613 - val_loss: 0.1800 - val_accuracy: 0.9350
Epoch 21/200
25/25 [==============================] - 0s 11ms/step - loss: 0.1228 - accuracy:
0.9725 - val_loss: 0.1558 - val_accuracy: 0.9450
Epoch 22/200
25/25 [==============================] - 0s 11ms/step - loss: 0.1220 - accuracy:
0.9675 - val_loss: 0.1358 - val_accuracy: 0.9500
Epoch 23/200
25/25 [==============================] - 0s 11ms/step - loss: 0.1007 - accuracy:
0.9762 - val_loss: 0.1393 - val_accuracy: 0.9500
Epoch 24/200
25/25 [==============================] - 0s 14ms/step - loss: 0.1020 - accuracy:
0.9725 - val_loss: 0.1220 - val_accuracy: 0.9550
Epoch 25/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0892 - accuracy:
0.9800 - val_loss: 0.1225 - val_accuracy: 0.9550
Epoch 26/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0879 - accuracy:
0.9762 - val_loss: 0.1232 - val_accuracy: 0.9500
Epoch 27/200
25/25 [==============================] - 0s 13ms/step - loss: 0.0841 - accuracy:
0.9762 - val_loss: 0.1161 - val_accuracy: 0.9700
Epoch 28/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0802 - accuracy:
```

```
0.9850 - val_loss: 0.1075 - val_accuracy: 0.9700
Epoch 29/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0831 - accuracy:
0.9812 - val_loss: 0.1166 - val_accuracy: 0.9500
Epoch 30/200
25/25 [==============================] - 0s 13ms/step - loss: 0.0654 - accuracy:
0.9837 - val_loss: 0.1184 - val_accuracy: 0.9650
Epoch 31/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0630 - accuracy:
0.9875 - val_loss: 0.1141 - val_accuracy: 0.9600
Epoch 32/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0556 - accuracy:
0.9862 - val_loss: 0.1214 - val_accuracy: 0.9550
Epoch 33/200
25/25 [==============================] - 0s 14ms/step - loss: 0.0524 - accuracy:
0.9862 - val_loss: 0.1203 - val_accuracy: 0.9550
Epoch 34/200
25/25 [==============================] - 0s 17ms/step - loss: 0.0556 - accuracy:
0.9875 - val_loss: 0.1124 - val_accuracy: 0.9700
Epoch 35/200
25/25 [==============================] - 0s 17ms/step - loss: 0.0461 - accuracy:
0.9887 - val_loss: 0.1211 - val_accuracy: 0.9550
Epoch 36/200
25/25 [==============================] - 0s 18ms/step - loss: 0.0471 - accuracy:
0.9912 - val_loss: 0.0993 - val_accuracy: 0.9600
Epoch 37/200
25/25 [==============================] - 0s 17ms/step - loss: 0.0553 - accuracy:
0.9862 - val_loss: 0.0924 - val_accuracy: 0.9750
Epoch 38/200
25/25 [==============================] - 0s 19ms/step - loss: 0.0408 - accuracy:
0.9912 - val_loss: 0.0915 - val_accuracy: 0.9750
Epoch 39/200
25/25 [==============================] - 0s 17ms/step - loss: 0.0498 - accuracy:
0.9850 - val_loss: 0.1021 - val_accuracy: 0.9600
Epoch 40/200
25/25 [==============================] - 0s 19ms/step - loss: 0.0459 - accuracy:
0.9887 - val_loss: 0.1058 - val_accuracy: 0.9550
Epoch 41/200
25/25 [==============================] - 0s 15ms/step - loss: 0.0406 - accuracy:
0.9925 - val_loss: 0.0927 - val_accuracy: 0.9650
Epoch 42/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0299 - accuracy:
0.9962 - val_loss: 0.0867 - val_accuracy: 0.9700
Epoch 43/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0300 - accuracy:
0.9962 - val_loss: 0.0855 - val_accuracy: 0.9600
Epoch 44/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0326 - accuracy:
```

```
0.9950 - val_loss: 0.0905 - val_accuracy: 0.9600
Epoch 45/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0257 - accuracy:
0.9975 - val_loss: 0.0830 - val_accuracy: 0.9750
Epoch 46/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0268 - accuracy:
0.9962 - val_loss: 0.0845 - val_accuracy: 0.9750
Epoch 47/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0273 - accuracy:
0.9962 - val_loss: 0.0827 - val_accuracy: 0.9750
Epoch 48/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0258 - accuracy:
0.9962 - val_loss: 0.0812 - val_accuracy: 0.9700
Epoch 49/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0257 - accuracy:
0.9987 - val_loss: 0.0875 - val_accuracy: 0.9700
Epoch 50/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0202 - accuracy:
0.9987 - val_loss: 0.0790 - val_accuracy: 0.9750
Epoch 51/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0197 - accuracy:
0.9950 - val_loss: 0.0831 - val_accuracy: 0.9650
Epoch 52/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0239 - accuracy:
0.9962 - val_loss: 0.1035 - val_accuracy: 0.9600
Epoch 53/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0221 - accuracy:
0.9962 - val_loss: 0.0829 - val_accuracy: 0.9700
Epoch 54/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0151 - accuracy:
1.0000 - val_loss: 0.0817 - val_accuracy: 0.9650
Epoch 55/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0171 - accuracy:
0.9975 - val_loss: 0.0960 - val_accuracy: 0.9550
Epoch 56/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0139 - accuracy:
1.0000 - val_loss: 0.0839 - val_accuracy: 0.9650
Epoch 57/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0148 - accuracy:
0.9987 - val_loss: 0.0801 - val_accuracy: 0.9700
Epoch 58/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0184 - accuracy:
0.9962 - val_loss: 0.0849 - val_accuracy: 0.9600
Epoch 59/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0179 - accuracy:
0.9975 - val_loss: 0.0912 - val_accuracy: 0.9750
Epoch 60/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0196 - accuracy:
```

```
0.9975 - val_loss: 0.0948 - val_accuracy: 0.9600
Epoch 61/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0145 - accuracy:
1.0000 - val_loss: 0.0728 - val_accuracy: 0.9750
Epoch 62/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0154 - accuracy:
0.9975 - val_loss: 0.0836 - val_accuracy: 0.9750
Epoch 63/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0106 - accuracy:
0.9975 - val_loss: 0.0990 - val_accuracy: 0.9600
Epoch 64/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0137 - accuracy:
1.0000 - val_loss: 0.0724 - val_accuracy: 0.9800
Epoch 65/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0104 - accuracy:
1.0000 - val_loss: 0.0811 - val_accuracy: 0.9650
Epoch 66/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0104 - accuracy:
1.0000 - val_loss: 0.0797 - val_accuracy: 0.9650
Epoch 67/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0128 - accuracy:
0.9987 - val_loss: 0.1013 - val_accuracy: 0.9650
Epoch 68/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0128 - accuracy:
1.0000 - val_loss: 0.0739 - val_accuracy: 0.9700
Epoch 69/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0085 - accuracy:
1.0000 - val_loss: 0.0765 - val_accuracy: 0.9700
Epoch 70/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0082 - accuracy:
1.0000 - val_loss: 0.0781 - val_accuracy: 0.9700
Epoch 71/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0081 - accuracy:
1.0000 - val_loss: 0.0810 - val_accuracy: 0.9750
Epoch 72/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0067 - accuracy:
1.0000 - val_loss: 0.0794 - val_accuracy: 0.9650
Epoch 73/200
25/25 [==============================] - 0s 13ms/step - loss: 0.0085 - accuracy:
1.0000 - val_loss: 0.0782 - val_accuracy: 0.9650
Epoch 74/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0092 - accuracy:
1.0000 - val_loss: 0.0706 - val_accuracy: 0.9700
Epoch 75/200
25/25 [==============================] - 0s 14ms/step - loss: 0.0088 - accuracy:
0.9987 - val_loss: 0.0782 - val_accuracy: 0.9750
Epoch 76/200
25/25 [==============================] - 0s 17ms/step - loss: 0.0097 - accuracy:
```

```
0.9987 - val_loss: 0.0893 - val_accuracy: 0.9600
Epoch 77/200
25/25 [==============================] - 0s 18ms/step - loss: 0.0100 - accuracy:
0.9987 - val_loss: 0.0827 - val_accuracy: 0.9750
Epoch 78/200
25/25 [==============================] - 0s 18ms/step - loss: 0.0100 - accuracy:
0.9987 - val_loss: 0.0965 - val_accuracy: 0.9700
Epoch 79/200
25/25 [==============================] - 0s 18ms/step - loss: 0.0082 - accuracy:
1.0000 - val_loss: 0.0977 - val_accuracy: 0.9600
Epoch 80/200
25/25 [==============================] - 0s 17ms/step - loss: 0.0061 - accuracy:
1.0000 - val_loss: 0.0744 - val_accuracy: 0.9700
Epoch 81/200
25/25 [==============================] - 0s 17ms/step - loss: 0.0053 - accuracy:
1.0000 - val_loss: 0.0795 - val_accuracy: 0.9600
Epoch 82/200
25/25 [==============================] - 0s 18ms/step - loss: 0.0051 - accuracy:
1.0000 - val_loss: 0.0782 - val_accuracy: 0.9750
Epoch 83/200
25/25 [==============================] - 0s 18ms/step - loss: 0.0055 - accuracy:
1.0000 - val_loss: 0.0905 - val_accuracy: 0.9650
Epoch 84/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0058 - accuracy:
1.0000 - val_loss: 0.0748 - val_accuracy: 0.9700
Epoch 85/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0056 - accuracy:
1.0000 - val_loss: 0.0851 - val_accuracy: 0.9750
Epoch 86/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0044 - accuracy:
1.0000 - val_loss: 0.0820 - val_accuracy: 0.9700
Epoch 87/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0064 - accuracy:
1.0000 - val_loss: 0.0848 - val_accuracy: 0.9750
Epoch 88/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0041 - accuracy:
1.0000 - val_loss: 0.0786 - val_accuracy: 0.9700
Epoch 89/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0045 - accuracy:
1.0000 - val_loss: 0.0919 - val_accuracy: 0.9700
Epoch 90/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0060 - accuracy:
1.0000 - val_loss: 0.0775 - val_accuracy: 0.9700
Epoch 91/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0054 - accuracy:
1.0000 - val_loss: 0.0759 - val_accuracy: 0.9750
Epoch 92/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0061 - accuracy:
```

```
1.0000 - val_loss: 0.0993 - val_accuracy: 0.9600
Epoch 93/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0058 - accuracy:
0.9987 - val_loss: 0.0839 - val_accuracy: 0.9700
Epoch 94/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0060 - accuracy:
1.0000 - val_loss: 0.0729 - val_accuracy: 0.9750
Epoch 95/200
25/25 [==============================] - 0s 13ms/step - loss: 0.0047 - accuracy:
1.0000 - val_loss: 0.0823 - val_accuracy: 0.9750
Epoch 96/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0040 - accuracy:
1.0000 - val_loss: 0.0722 - val_accuracy: 0.9750
Epoch 97/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0046 - accuracy:
1.0000 - val_loss: 0.0753 - val_accuracy: 0.9700
Epoch 98/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0043 - accuracy:
1.0000 - val_loss: 0.0879 - val_accuracy: 0.9600
Epoch 99/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0036 - accuracy:
1.0000 - val_loss: 0.0785 - val_accuracy: 0.9750
Epoch 100/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0045 - accuracy:
1.0000 - val_loss: 0.0850 - val_accuracy: 0.9750
Epoch 101/200
25/25 [==============================] - 0s 13ms/step - loss: 0.0075 - accuracy:
0.9987 - val_loss: 0.1000 - val_accuracy: 0.9550
Epoch 102/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0041 - accuracy:
1.0000 - val_loss: 0.0780 - val_accuracy: 0.9700
Epoch 103/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0033 - accuracy:
1.0000 - val_loss: 0.0982 - val_accuracy: 0.9600
Epoch 104/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0038 - accuracy:
1.0000 - val_loss: 0.0796 - val_accuracy: 0.9750
Epoch 105/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0032 - accuracy:
1.0000 - val_loss: 0.0758 - val_accuracy: 0.9750
Epoch 106/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0032 - accuracy:
1.0000 - val_loss: 0.0853 - val_accuracy: 0.9750
Epoch 107/200
25/25 [==============================] - 0s 13ms/step - loss: 0.0037 - accuracy:
1.0000 - val_loss: 0.1192 - val_accuracy: 0.9600
Epoch 108/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0045 - accuracy:
```

```
1.0000 - val_loss: 0.0712 - val_accuracy: 0.9750
Epoch 109/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0036 - accuracy:
1.0000 - val_loss: 0.0794 - val_accuracy: 0.9750
Epoch 110/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0036 - accuracy:
1.0000 - val_loss: 0.0790 - val_accuracy: 0.9750
Epoch 111/200
25/25 [==============================] - 0s 13ms/step - loss: 0.0023 - accuracy:
1.0000 - val_loss: 0.0765 - val_accuracy: 0.9750
Epoch 112/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0053 - accuracy:
0.9987 - val_loss: 0.0998 - val_accuracy: 0.9700
Epoch 113/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0043 - accuracy:
1.0000 - val_loss: 0.0738 - val_accuracy: 0.9750
Epoch 114/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0029 - accuracy:
1.0000 - val_loss: 0.0713 - val_accuracy: 0.9700
Epoch 115/200
25/25 [==============================] - 0s 13ms/step - loss: 0.0023 - accuracy:
1.0000 - val_loss: 0.0748 - val_accuracy: 0.9800
Epoch 116/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0025 - accuracy:
1.0000 - val_loss: 0.0753 - val_accuracy: 0.9750
Epoch 117/200
25/25 [==============================] - 0s 17ms/step - loss: 0.0018 - accuracy:
1.0000 - val_loss: 0.0782 - val_accuracy: 0.9800
Epoch 118/200
25/25 [==============================] - 0s 15ms/step - loss: 0.0020 - accuracy:
1.0000 - val_loss: 0.0798 - val_accuracy: 0.9700
Epoch 119/200
25/25 [==============================] - 0s 16ms/step - loss: 0.0029 - accuracy:
1.0000 - val_loss: 0.0762 - val_accuracy: 0.9750
Epoch 120/200
25/25 [==============================] - 0s 19ms/step - loss: 0.0017 - accuracy:
1.0000 - val_loss: 0.0755 - val_accuracy: 0.9700
Epoch 121/200
25/25 [==============================] - 0s 18ms/step - loss: 0.0018 - accuracy:
1.0000 - val_loss: 0.0754 - val_accuracy: 0.9750
Epoch 122/200
25/25 [==============================] - 0s 18ms/step - loss: 0.0017 - accuracy:
1.0000 - val_loss: 0.0737 - val_accuracy: 0.9750
Epoch 123/200
25/25 [==============================] - 0s 18ms/step - loss: 0.0019 - accuracy:
1.0000 - val_loss: 0.0736 - val_accuracy: 0.9750
Epoch 124/200
25/25 [==============================] - 0s 17ms/step - loss: 0.0030 - accuracy:
```

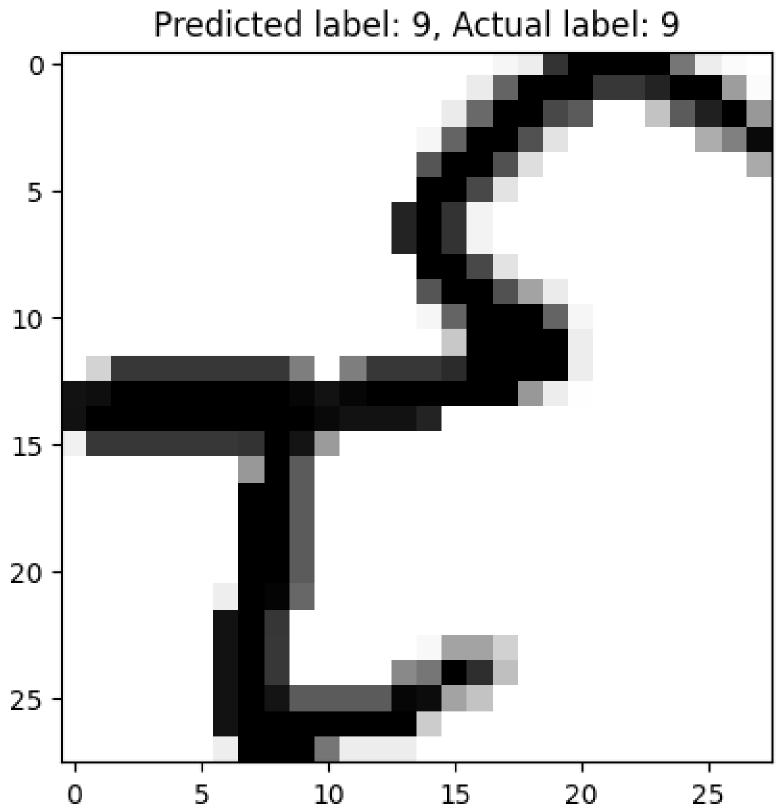
```
1.0000 - val_loss: 0.0755 - val_accuracy: 0.9700
Epoch 125/200
25/25 [==============================] - 0s 14ms/step - loss: 0.0026 - accuracy:
1.0000 - val_loss: 0.0668 - val_accuracy: 0.9750
Epoch 126/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0018 - accuracy:
1.0000 - val_loss: 0.0596 - val_accuracy: 0.9800
Epoch 127/200
25/25 [==============================] - 0s 14ms/step - loss: 0.0019 - accuracy:
1.0000 - val_loss: 0.0676 - val_accuracy: 0.9800
Epoch 128/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0021 - accuracy:
1.0000 - val_loss: 0.0819 - val_accuracy: 0.9650
Epoch 129/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0014 - accuracy:
1.0000 - val_loss: 0.0770 - val_accuracy: 0.9750
Epoch 130/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0011 - accuracy:
1.0000 - val_loss: 0.0777 - val_accuracy: 0.9750
Epoch 131/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0019 - accuracy:
1.0000 - val_loss: 0.0737 - val_accuracy: 0.9700
Epoch 132/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0018 - accuracy:
1.0000 - val_loss: 0.0800 - val_accuracy: 0.9650
Epoch 133/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0025 - accuracy:
1.0000 - val_loss: 0.0766 - val_accuracy: 0.9750
Epoch 134/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0032 - accuracy:
1.0000 - val_loss: 0.0697 - val_accuracy: 0.9750
Epoch 135/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0024 - accuracy:
1.0000 - val_loss: 0.0818 - val_accuracy: 0.9700
Epoch 136/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0018 - accuracy:
1.0000 - val_loss: 0.0760 - val_accuracy: 0.9750
Epoch 137/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0012 - accuracy:
1.0000 - val_loss: 0.0774 - val_accuracy: 0.9700
Epoch 138/200
25/25 [==============================] - 0s 12ms/step - loss: 9.3421e-04 -
accuracy: 1.0000 - val_loss: 0.0806 - val_accuracy: 0.9700
Epoch 139/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0015 - accuracy:
1.0000 - val_loss: 0.0832 - val_accuracy: 0.9750
Epoch 140/200
25/25 [==============================] - 0s 13ms/step - loss: 0.0010 - accuracy:
```

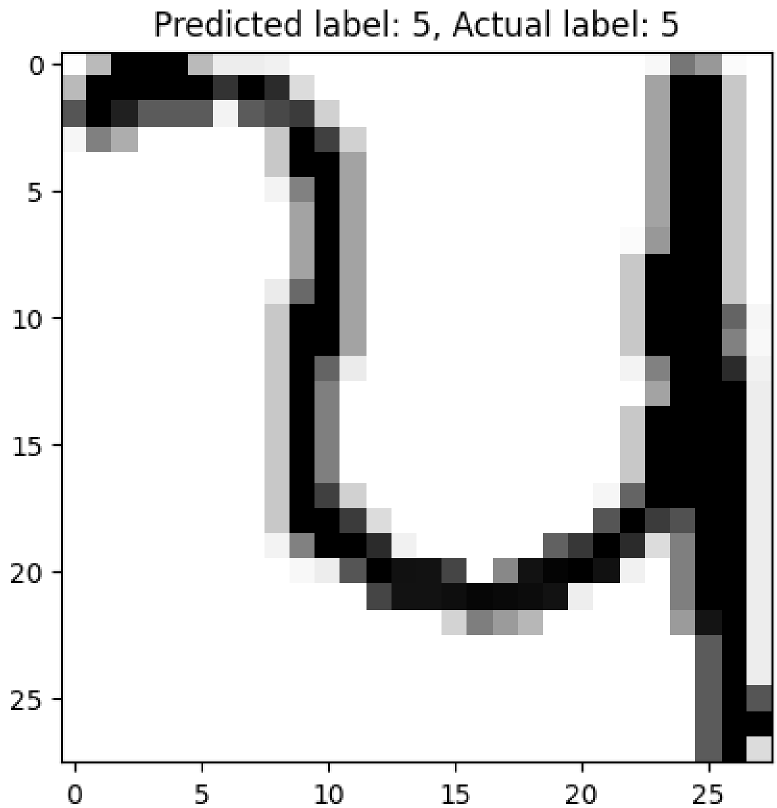
```
1.0000 - val_loss: 0.0861 - val_accuracy: 0.9700
Epoch 141/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0012 - accuracy:
1.0000 - val_loss: 0.0854 - val_accuracy: 0.9700
Epoch 142/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0011 - accuracy:
1.0000 - val_loss: 0.0872 - val_accuracy: 0.9700
Epoch 143/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0017 - accuracy:
1.0000 - val_loss: 0.0770 - val_accuracy: 0.9700
Epoch 144/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0017 - accuracy:
1.0000 - val_loss: 0.0825 - val_accuracy: 0.9750
Epoch 145/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0017 - accuracy:
1.0000 - val_loss: 0.0706 - val_accuracy: 0.9700
Epoch 146/200
25/25 [==============================] - 0s 13ms/step - loss: 0.0014 - accuracy:
1.0000 - val_loss: 0.0702 - val_accuracy: 0.9750
Epoch 147/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0014 - accuracy:
1.0000 - val_loss: 0.0682 - val_accuracy: 0.9750
Epoch 148/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0014 - accuracy:
1.0000 - val_loss: 0.0760 - val_accuracy: 0.9750
Epoch 149/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0016 - accuracy:
1.0000 - val_loss: 0.0923 - val_accuracy: 0.9650
Epoch 150/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0030 - accuracy:
1.0000 - val_loss: 0.0968 - val_accuracy: 0.9550
Epoch 151/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0022 - accuracy:
1.0000 - val_loss: 0.0848 - val_accuracy: 0.9750
Epoch 152/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0019 - accuracy:
1.0000 - val_loss: 0.0942 - val_accuracy: 0.9650
Epoch 153/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0016 - accuracy:
1.0000 - val_loss: 0.0793 - val_accuracy: 0.9800
Epoch 154/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0011 - accuracy:
1.0000 - val_loss: 0.0763 - val_accuracy: 0.9700
Epoch 155/200
25/25 [==============================] - 0s 12ms/step - loss: 9.0063e-04 -
accuracy: 1.0000 - val_loss: 0.0751 - val_accuracy: 0.9850
Epoch 156/200
25/25 [==============================] - 0s 14ms/step - loss: 0.0012 - accuracy:
```

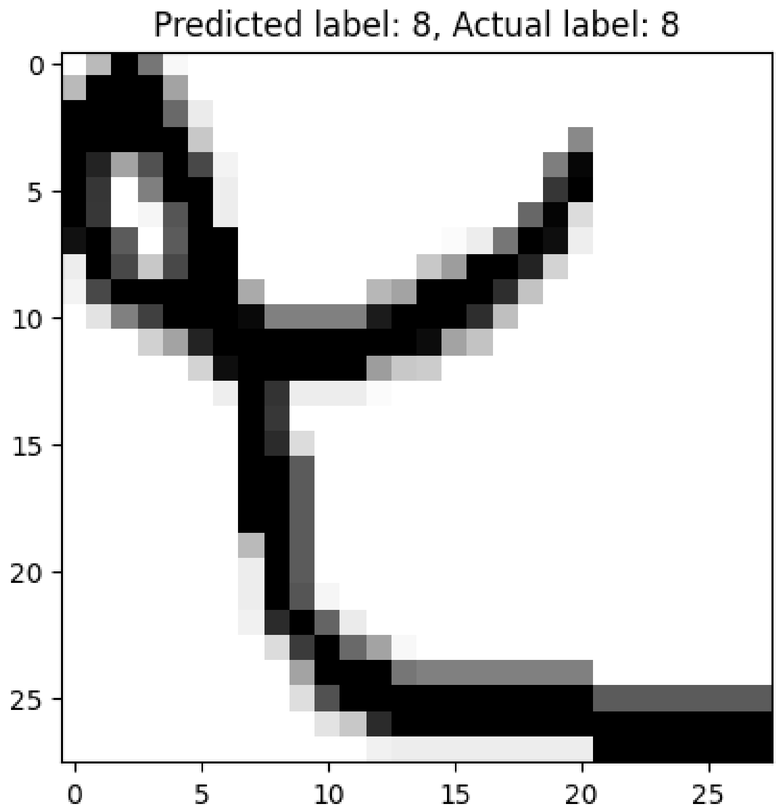
```
1.0000 - val_loss: 0.0714 - val_accuracy: 0.9700
Epoch 157/200
25/25 [==============================] - 0s 13ms/step - loss: 0.0012 - accuracy:
1.0000 - val_loss: 0.0681 - val_accuracy: 0.9750
Epoch 158/200
25/25 [==============================] - 0s 14ms/step - loss: 8.1820e-04 -
accuracy: 1.0000 - val_loss: 0.0822 - val_accuracy: 0.9700
Epoch 159/200
25/25 [==============================] - 1s 20ms/step - loss: 0.0014 - accuracy:
1.0000 - val_loss: 0.0818 - val_accuracy: 0.9750
Epoch 160/200
25/25 [==============================] - 0s 18ms/step - loss: 8.9058e-04 -
accuracy: 1.0000 - val_loss: 0.0726 - val_accuracy: 0.9750
Epoch 161/200
25/25 [==============================] - 0s 18ms/step - loss: 8.3580e-04 -
accuracy: 1.0000 - val_loss: 0.0783 - val_accuracy: 0.9700
Epoch 162/200
25/25 [==============================] - 0s 18ms/step - loss: 8.7571e-04 -
accuracy: 1.0000 - val_loss: 0.0807 - val_accuracy: 0.9700
Epoch 163/200
25/25 [==============================] - 0s 17ms/step - loss: 9.4772e-04 -
accuracy: 1.0000 - val_loss: 0.0780 - val_accuracy: 0.9850
Epoch 164/200
25/25 [==============================] - 0s 16ms/step - loss: 9.8796e-04 -
accuracy: 1.0000 - val_loss: 0.0921 - val_accuracy: 0.9600
Epoch 165/200
25/25 [==============================] - 0s 18ms/step - loss: 0.0011 - accuracy:
1.0000 - val_loss: 0.0860 - val_accuracy: 0.9700
Epoch 166/200
25/25 [==============================] - 0s 16ms/step - loss: 6.9946e-04 -
accuracy: 1.0000 - val_loss: 0.0697 - val_accuracy: 0.9750
Epoch 167/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0010 - accuracy:
1.0000 - val_loss: 0.0779 - val_accuracy: 0.9850
Epoch 168/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0017 - accuracy:
1.0000 - val_loss: 0.1024 - val_accuracy: 0.9800
Epoch 169/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0034 - accuracy:
1.0000 - val_loss: 0.0883 - val_accuracy: 0.9700
Epoch 170/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0085 - accuracy:
0.9962 - val_loss: 0.0883 - val_accuracy: 0.9800
Epoch 171/200
25/25 [==============================] - 0s 14ms/step - loss: 0.0034 - accuracy:
1.0000 - val_loss: 0.0943 - val_accuracy: 0.9650
Epoch 172/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0028 - accuracy:
```

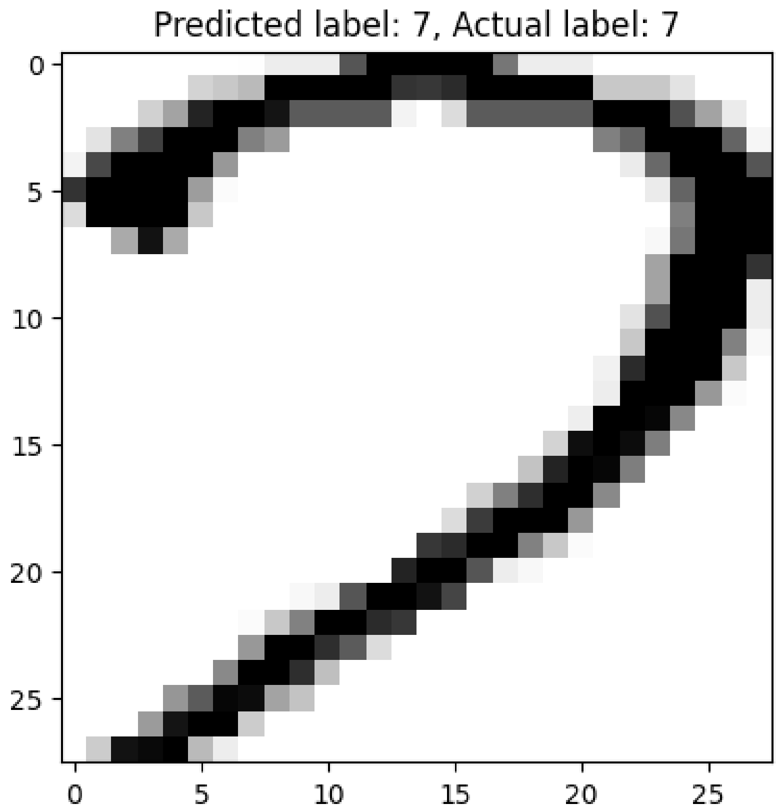
```
0.9987 - val_loss: 0.0944 - val_accuracy: 0.9600
Epoch 173/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0036 - accuracy:
1.0000 - val_loss: 0.1092 - val_accuracy: 0.9750
Epoch 174/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0031 - accuracy:
0.9987 - val_loss: 0.0865 - val_accuracy: 0.9650
Epoch 175/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0010 - accuracy:
1.0000 - val_loss: 0.0761 - val_accuracy: 0.9750
Epoch 176/200
25/25 [==============================] - 0s 14ms/step - loss: 8.8810e-04 -
accuracy: 1.0000 - val_loss: 0.0802 - val_accuracy: 0.9700
Epoch 177/200
25/25 [==============================] - 0s 12ms/step - loss: 7.1445e-04 -
accuracy: 1.0000 - val_loss: 0.0852 - val_accuracy: 0.9700
Epoch 178/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0066 - accuracy:
0.9975 - val_loss: 0.0936 - val_accuracy: 0.9650
Epoch 179/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0032 - accuracy:
1.0000 - val_loss: 0.1406 - val_accuracy: 0.9450
Epoch 180/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0077 - accuracy:
0.9975 - val_loss: 0.0825 - val_accuracy: 0.9700
Epoch 181/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0294 - accuracy:
0.9912 - val_loss: 0.1833 - val_accuracy: 0.9500
Epoch 182/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0168 - accuracy:
0.9962 - val_loss: 0.1114 - val_accuracy: 0.9750
Epoch 183/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0030 - accuracy:
1.0000 - val_loss: 0.1083 - val_accuracy: 0.9550
Epoch 184/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0017 - accuracy:
1.0000 - val_loss: 0.0868 - val_accuracy: 0.9750
Epoch 185/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0012 - accuracy:
1.0000 - val_loss: 0.0824 - val_accuracy: 0.9750
Epoch 186/200
25/25 [==============================] - 0s 12ms/step - loss: 6.7356e-04 -
accuracy: 1.0000 - val_loss: 0.0834 - val_accuracy: 0.9700
Epoch 187/200
25/25 [==============================] - 0s 11ms/step - loss: 0.0011 - accuracy:
1.0000 - val_loss: 0.0904 - val_accuracy: 0.9650
Epoch 188/200
25/25 [==============================] - 0s 12ms/step - loss: 7.7525e-04 -
```

```
accuracy: 1.0000 - val_loss: 0.0857 - val_accuracy: 0.9700
Epoch 189/200
25/25 [==============================] - 0s 12ms/step - loss: 5.5066e-04 -
accuracy: 1.0000 - val_loss: 0.0840 - val_accuracy: 0.9700
Epoch 190/200
25/25 [==============================] - 0s 12ms/step - loss: 7.4334e-04 -
accuracy: 1.0000 - val_loss: 0.0889 - val_accuracy: 0.9750
Epoch 191/200
25/25 [==============================] - 0s 12ms/step - loss: 7.5265e-04 -
accuracy: 1.0000 - val_loss: 0.0791 - val_accuracy: 0.9850
Epoch 192/200
25/25 [==============================] - 0s 12ms/step - loss: 9.8193e-04 -
accuracy: 1.0000 - val_loss: 0.0851 - val_accuracy: 0.9700
Epoch 193/200
25/25 [==============================] - 0s 12ms/step - loss: 6.1868e-04 -
accuracy: 1.0000 - val_loss: 0.0823 - val_accuracy: 0.9750
Epoch 194/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0041 - accuracy:
0.9987 - val_loss: 0.1033 - val_accuracy: 0.9600
Epoch 195/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0079 - accuracy:
0.9987 - val_loss: 0.1107 - val_accuracy: 0.9650
Epoch 196/200
25/25 [==============================] - 0s 12ms/step - loss: 0.0020 - accuracy:
1.0000 - val_loss: 0.0798 - val_accuracy: 0.9850
Epoch 197/200
25/25 [==============================] - 0s 12ms/step - loss: 8.4830e-04 -
accuracy: 1.0000 - val_loss: 0.0877 - val_accuracy: 0.9750
Epoch 198/200
25/25 [==============================] - 0s 12ms/step - loss: 6.7996e-04 -
accuracy: 1.0000 - val_loss: 0.0811 - val_accuracy: 0.9850
Epoch 199/200
25/25 [==============================] - 0s 16ms/step - loss: 7.5595e-04 -
accuracy: 1.0000 - val_loss: 0.0748 - val_accuracy: 0.9850
Epoch 200/200
25/25 [==============================] - 0s 19ms/step - loss: 8.3946e-04 -
accuracy: 1.0000 - val_loss: 0.0763 - val_accuracy: 0.9750
6/6 [==============================] - 0s 5ms/step - loss: 0.2474 - accuracy:
0.9607
Test loss: 0.24740873277187347
Test accuracy: 0.9606741666793823
6/6 [==============================] - 0s 4ms/step
```


Predicted label: 9, Actual label: 9
0
5
10
15
20
25
0
5
10
15
20
25


Predicted label: 5, Actual label: 5
0
5
10
15
20
25
0
5
10
15
20
25


Predicted label: 8, Actual label: 8
0
5
10
15
20
25
0
5
10
15
20
25



[18]:
```
# Generate confusion matrix and heatmap
conf_matrix = confusion_matrix(y_test, y_pred)
print("conf_matrix : ","\n")
print(conf_matrix,"\n")
```

```
conf_matrix :

[[18  0  0  0  0  0  0  0  0  0]
 [ 0 13  0  0  0  0  0  3  0  0]
 [ 0  0 14  0  1  2  0  0  0  0]
 [ 0  0  0 17  0  0  0  0  0  0]
 [ 0  0  0  0 18  0  0  0  0  0]
 [ 0  0  0  0  0 18  0  0  0  0]
 [ 0  0  0  0  0  0 18  0  0  0]
 [ 0  1  0  0  0  0  0 17  0  0]
 [ 0  0  0  0  0  0  0  0 18  0]
 [ 0  0  0  0  0  0  0  0  0 20]]
```

[19]:
```
sns.heatmap(conf_matrix, annot=True, cmap="Blues", fmt="d")
plt.xlabel('Predicted Label')
```

```
plt.ylabel('True Label')
plt.show()
```