**MNIST ALGORITHM**

**CODE AND OUTPUTS:**

import tensorflow as tf

(x\_train, y\_train), (x\_test, y\_test) = tf.keras.datasets.mnist.load\_data()

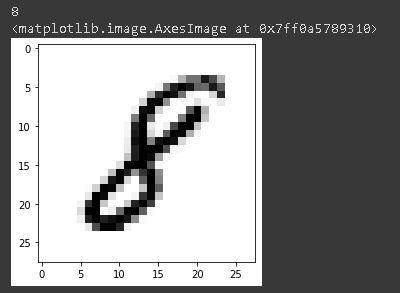
import matplotlib.pyplot as plt

%matplotlib inline

image\_index = 7777

print(y\_train[image\_index])

plt.imshow(x\_train[image\_index], cmap='Greys')



x\_train.shape



x\_train = x\_train.reshape(x\_train.shape[0], 28, 28, 1)

x\_test = x\_test.reshape(x\_test.shape[0], 28, 28, 1)

input\_shape = (28, 28, 1)

x\_train = x\_train.astype('float32')

x\_test = x\_test.astype('float32')

x\_train /= 255

x\_test /= 255

print('x\_train shape:', x\_train.shape)



from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, Conv2D, Dropout, Flatten, MaxPooling2D

model = Sequential()

model.add(Conv2D(28, kernel\_size=(3,3), input\_shape=input\_shape))

model.add(MaxPooling2D(pool\_size=(2, 2)))

model.add(Flatten())

model.add(Dense(128, activation=tf.nn.relu))

model.add(Dropout(0.2))

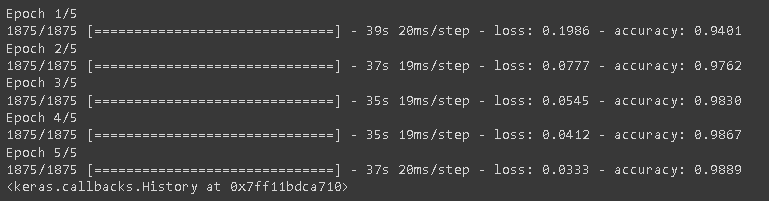
model.add(Dense(10,activation=tf.nn.softmax))

model.compile(optimizer='adam',

              loss='sparse\_categorical\_crossentropy',

              metrics=['accuracy'])

model.fit(x=x\_train,y=y\_train, epochs=5)



model.evaluate(x\_test, y\_test)



image\_index = 50

plt.imshow(x\_test[image\_index].reshape(28, 28),cmap='Greys')

pred = model.predict(x\_test[image\_index].reshape(1, 28, 28, 1))

print(pred.argmax())

