Image Classification using CNN on MNIST Dataset

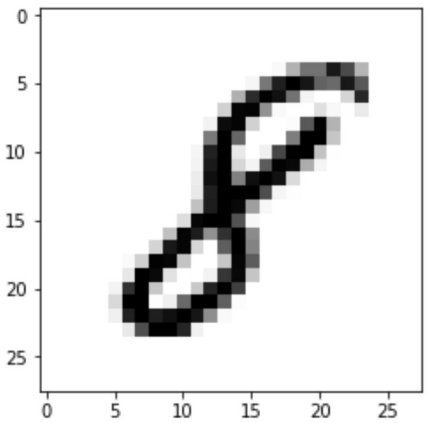
In [1]:
```
import tensorflow as tf
import keras
from keras.models import Sequential
from keras.layers import Dense, Conv2D, Dropout, Flatten, MaxPooling2D
(x_train, y_train), (x_test, y_test) = tf.keras.datasets.mnist.load_data()
```

```
Using TensorFlow backend.
```

In [3]:
```
import matplotlib.pyplot as plt
image_index = 7777 # You may select anything up to 60,000
print(y_train[image_index]) # The label is 8
plt.imshow(x_train[image_index], cmap='Greys')
```

```
8
```

Out[3]:
```
<matplotlib.image.AxesImage at 0x2cf1547d978>
```



In [4]:
```
x_train.shape
```

Out[4]:
```
(60000, 28, 28)
```

In [5]:
```
# Reshaping the array to 4-dims so that it can work with the Keras API
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)
# Making sure that the values are float so that we can get decimal points after div
ision
x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
# Normalizing the RGB codes by dividing it to the max RGB value.
x_train /= 255
x_test /= 255
print('x_train shape:', x_train.shape)
print('Number of images in x_train', x_train.shape[0])
print('Number of images in x_test', x_test.shape[0])
```

```
x_train shape: (60000, 28, 28, 1)
Number of images in x_train 60000
Number of images in x_test 10000
```

In [6]:
```
# Creating a Sequential Model and adding the layers
model = Sequential()
model.add(Conv2D(28, kernel_size=(3,3), input_shape=input_shape))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten()) # Flattening the 2D arrays for fully connected layers
model.add(Dense(128, activation=tf.nn.relu))
model.add(Dropout(0.2))
model.add(Dense(10,activation=tf.nn.softmax))
```

```
WARNING:tensorflow:From d:\ProgramData\Anaconda3\lib\site-packages\tensorflow\py
thon\framework\op_def_library.py:263: colocate_with (from tensorflow.python.fram
ework.ops) is deprecated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
WARNING:tensorflow:From d:\ProgramData\Anaconda3\lib\site-packages\keras\backend
\tensorflow_backend.py:3445: calling dropout (from tensorflow.python.ops.nn_ops)
with keep_prob is deprecated and will be removed in a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - keep
_prob`.
```

In [7]:
```
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
model.fit(x=x_train,y=y_train, epochs=3)
```

```
WARNING:tensorflow:From d:\ProgramData\Anaconda3\lib\site-packages\tensorflow\py
thon\ops\math_ops.py:3066: to_int32 (from tensorflow.python.ops.math_ops) is dep
recated and will be removed in a future version.
Instructions for updating:
Use tf.cast instead.
Epoch 1/3
60000/60000 [==============================] - 69s 1ms/step - loss: 0.2069 - acc
: 0.9383
Epoch 2/3
60000/60000 [==============================] - 68s 1ms/step - loss: 0.0867 - acc
: 0.9734
Epoch 3/3
60000/60000 [==============================] - 69s 1ms/step - loss: 0.0598 - acc
: 0.9814
```

Out[7]:
```
<keras.callbacks.History at 0x2cf1544add8>
```

In [8]:
```
image_index = 4444
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())
```

```
9
```