Author:Muhammad Bilal

Image Classification in 10 Minutes with MNIST Dataset

Using Convolutional Neural Networks to Classify Handwritten Digits with TensorFlow and Keras | Supervised Deep Learning

In [62]:
```
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
from tensorflow import keras
# Importing MNIST Dataset
mnist = tf.keras.datasets.mnist
(X_train, Y_train), (X_test, Y_test) = mnist.load_data()
print(X_train.shape)

#Reshaping and Normalizing the Images
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

#Building the Convolutional Neural Network
model=tf.keras.models.Sequential()
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D, Dropout, Flatten, MaxPooling2D
# 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))
```

```
(60000, 28, 28)
```

In [65]:
```
#Compiling and Fitting the Model
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
model.fit(x=X_train,y=Y_train, epochs=10)
```

```
Epoch 1/10
1875/1875 [==============================] - 35s 18ms/step - loss: 0.3468 - accuracy: 0.8977
Epoch 2/10
1875/1875 [==============================] - 36s 19ms/step - loss: 0.0809 - accuracy: 0.9747
Epoch 3/10
1875/1875 [==============================] - 35s 19ms/step - loss: 0.0551 - accuracy: 0.9822
Epoch 4/10
1875/1875 [==============================] - 35s 19ms/step - loss: 0.0435 - accuracy: 0.9856
Epoch 5/10
1875/1875 [==============================] - 35s 18ms/step - loss: 0.0323 - accuracy: 0.9892
Epoch 6/10
1875/1875 [==============================] - 35s 18ms/step - loss: 0.0263 - accuracy: 0.9918
Epoch 7/10
1875/1875 [==============================] - 40s 21ms/step - loss: 0.0243 - accuracy: 0.9918
Epoch 8/10
1875/1875 [==============================] - 40s 22ms/step - loss: 0.0201 - accuracy: 0.9931
Epoch 9/10
1875/1875 [==============================] - 55s 29ms/step - loss: 0.0166 - accuracy: 0.9945
Epoch 10/10
1875/1875 [==============================] - 70s 37ms/step - loss: 0.0184 - accuracy: 0.9933
```

Out[65]:
```
<tensorflow.python.keras.callbacks.History at 0x1e99007c3a0>
```

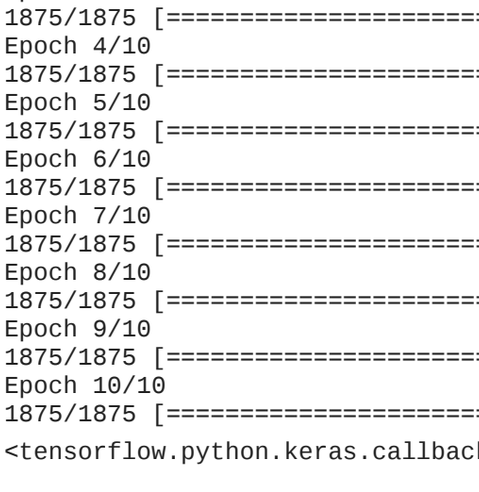
In [67]:
```
#Evaluating the Model
model.evaluate(X_test, Y_test)
```

```
313/313 [==============================] - 2s 6ms/step - loss: 0.0657 - accuracy: 0.9841
```

Out[67]:
```
[0.06565799564123154, 0.9840999841690063]
```

In [69]:
```
#testing our prediction with the model
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
```



In [ ]: