

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from keras.utils import np_utils
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, Dense, Flatten
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.models import load_model
from PIL import Image, ImageOps
import numpy

(X_train, y_train), (X_test, y_test) = mnist.load_data()

Downloading data from https://storage.googleapis.com/tensorflow/tf-
keras-datasets/mnist.npz
11490434/11490434 [=====] - 0s 0us/step

print(X_train.shape)
print(X_test.shape)

(60000, 28, 28)
(10000, 28, 28)

X_train[0]
```

[illegible]



```

0,      [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
0,      0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
0,      0,  0],
0,      [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
0,      0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
0,      0,  0],
0,      [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
0,      0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
0,      0,  0]], dtype=uint8)

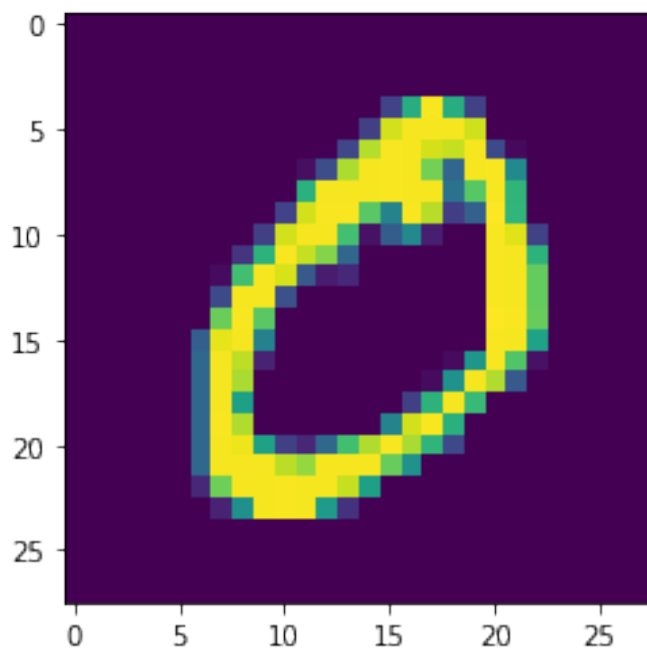
```

```
y_train[0]
```

```
5
```

```
plt.imshow(X_train[1])
```

```
<matplotlib.image.AxesImage at 0x7f7679f22b90>
```



```
X_train = X_train.reshape(60000, 28, 28, 1).astype('float32')
```

```
X_test = X_test.reshape(10000, 28, 28, 1).astype('float32')
```

```
number_of_classes = 10
```

```
Y_train = np_utils.to_categorical(y_train, number_of_classes)
```

```
Y_test = np_utils.to_categorical(y_test, number_of_classes)
```

```

Y_train[0]

array([0., 0., 0., 0., 0., 1., 0., 0., 0., 0.], dtype=float32)

model = Sequential()
model.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1),
activation="relu"))
model.add(Conv2D(32, (3, 3), activation="relu"))
model.add(Flatten())
model.add(Dense(number_of_classes, activation="softmax"))

model.compile(loss='categorical_crossentropy', optimizer="Adam",
metrics=["accuracy"])
model.fit(X_train, Y_train, batch_size=32, epochs=5,
validation_data=(X_test,Y_test))

Epoch 1/5
1875/1875 [=====] - 202s 107ms/step - loss:
0.1927 - accuracy: 0.9542 - val_loss: 0.0764 - val_accuracy: 0.9769
Epoch 2/5
1875/1875 [=====] - 198s 106ms/step - loss:
0.0631 - accuracy: 0.9805 - val_loss: 0.0838 - val_accuracy: 0.9752
Epoch 3/5
1875/1875 [=====] - 208s 111ms/step - loss:
0.0421 - accuracy: 0.9862 - val_loss: 0.0839 - val_accuracy: 0.9783
Epoch 4/5
1875/1875 [=====] - 201s 107ms/step - loss:
0.0361 - accuracy: 0.9889 - val_loss: 0.0956 - val_accuracy: 0.9770
Epoch 5/5
1875/1875 [=====] - 201s 107ms/step - loss:
0.0261 - accuracy: 0.9917 - val_loss: 0.0900 - val_accuracy: 0.9807

<keras.callbacks.History at 0x7f767a4f4190>

metrics = model.evaluate(X_test, Y_test, verbose=0)
print("Metrics (Test Loss & Test Accuracy): ")
print(metrics)

Metrics (Test Loss & Test Accuracy):
[0.0900242030620575, 0.9807000160217285]

prediction = model.predict(X_test[:4])
print(prediction)

1/1 [=====] - 0s 103ms/step
[[2.1198963e-12 2.5143053e-16 2.9080804e-08 2.6923164e-08 1.0787076e-
15
 1.0072105e-12 3.8033966e-17 1.0000000e+00 2.4907893e-11 1.4726306e-
08]
 [1.6187166e-09 3.0472007e-12 9.9999833e-01 5.8330483e-09 6.1154418e-
13
 2.1986678e-15 1.6307678e-06 5.6883907e-17 1.5857615e-09 2.2175238e-

```

```

18]
[3.7463948e-07 9.8546553e-01 1.8378964e-03 2.5092066e-09 1.2474166e-
02
1.3331343e-07 9.2518391e-08 3.8755319e-05 1.8302012e-04 1.7161810e-
08]
[9.9964738e-01 6.1215101e-13 6.1620986e-10 4.1869831e-11 2.5735805e-
10
2.0356998e-09 3.5265239e-04 1.3145517e-10 9.6421496e-11 3.6439904e-
08]]

```

```

print(numpy.argmax(prediction, axis=1))
print(Y_test[:4])

```

```

[7 2 1 0]
[[0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]

```

```

model.save("model.h5")
model=load_model("model.h5")

```

```

from keras.datasets import mnist
from matplotlib import pyplot
(X_train,y_train),(X_test,y_test)=mnist.load_data()
print('X_train:' +str(X_train.shape))
print('y_train:' +str(y_train.shape))
print('X_test:' +str(X_test.shape))
print('y_test:' +str(y_test.shape))
from matplotlib import pyplot
for i in range(9):
    pyplot.subplot(330+1+i)
    pyplot.imshow(X_train[i],cmap=pyplot.get_cmap('gray'))
    pyplot.show()

```

```

X_train:(60000, 28, 28)
y_train:(60000,)
X_test:(10000, 28, 28)
y_test:(10000,)

```

