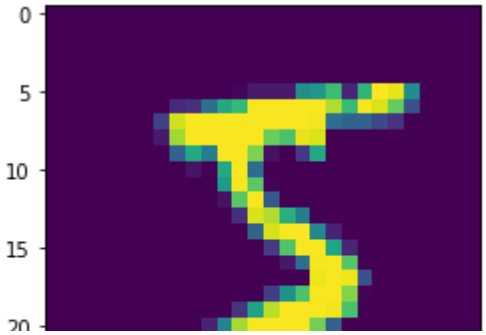
```
<matplotlib.image.AxesImage at 0x7ff6515c8a10>
```



Reshaping the data

```
0  5  10  15  20  25
```

```
X_train = X_train.reshape(60000,28,28,1).astype('float32')
X_test =X_test.reshape(10000,28,28,1).astype('float32')
```

One Hot Encoding

```
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 BUILDING

ADD CNN LAYERS

```
# CREATING THE MODEL
model = Sequential()
#adding model layer
number_of_classes = 10
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'))
```

Compiling the model

```
model.compile(loss='categorical_crossentropy',optimizer='Adam',metrics=['accuracy'])
```

Train the model

*Fitting the model *

```
model.fit(X_train,y_train,epochs= 2,validation_data=(X_test,y_test),batch_size=32)
```

```
Epoch 1/2
1875/1875 [==============================] - 188s 100ms/step - loss: 0.1276 - accuracy:
Epoch 2/2
1875/1875 [==============================] - 189s 101ms/step - loss: 0.0689 - accuracy:
<keras.callbacks.History at 0x7ff651c9f110>
```

OBSERVING THE METRICS

```
metrics= model.evaluate(X_test,y_test,verbose=0)
print("Metrics(Test loss & Test Accuracy): ")
print(metrics)
```

```
Metrics(Test loss & Test Accuracy):
[0.07622674852609634, 0.9776999950408936]
```

PREDICTING THE OUTPUT

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

```
1/1 [==============================] - 0s 92ms/step
[[6.5748842e-09 6.3890655e-12 1.3739229e-07 5.2780575e-07 2.2003690e-14
  3.7750227e-11 2.5512497e-15 9.9999893e-01 3.5263005e-07 8.7892829e-08]
 [4.9200077e-10 2.0602839e-10 1.0000000e+00 1.1946745e-11 5.0145482e-14
  6.3856547e-14 9.7302599e-10 2.1848258e-15 7.6805357e-10 1.4469144e-17]
 [1.1064673e-07 9.9972981e-01 6.4760707e-06 1.3149617e-07 9.1082089e-05
  6.4459650e-06 6.3640905e-06 6.1268469e-07 1.5884437e-04 1.1938099e-07]
 [9.9998248e-01 3.5105444e-08 3.3236324e-07 4.4056489e-10 2.5360112e-09
  1.1630171e-08 1.8082093e-06 8.8792412e-10 2.5261832e-08 1.5380074e-05]]
```

```
import numpy as np
print(np.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.]]
```

OBSERVING THE METRICS

```
metrics= model.evaluate(X_test,y_test,verbose=0)
print("Metrics(Test loss & Test Accuracy): ")
print(metrics)
```

```
Metrics(Test loss & Test Accuracy):
[0.07622674852609634, 0.9776999950408936]
```

TEST THE MODEL

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

```
1/1 [==============================] - 0s 19ms/step
[[6.5748842e-09 6.3890655e-12 1.3739229e-07 5.2780575e-07 2.2003690e-14
  3.7750227e-11 2.5512497e-15 9.9999893e-01 3.5263005e-07 8.7892829e-08]
 [4.9200077e-10 2.0602839e-10 1.0000000e+00 1.1946745e-11 5.0145482e-14
  6.3856547e-14 9.7302599e-10 2.1848258e-15 7.6805357e-10 1.4469144e-17]
 [1.1064673e-07 9.9972981e-01 6.4760707e-06 1.3149617e-07 9.1082089e-05
  6.4459650e-06 6.3640905e-06 6.1268469e-07 1.5884437e-04 1.1938099e-07]
 [9.9998248e-01 3.5105444e-08 3.3236324e-07 4.4056489e-10 2.5360112e-09
  1.1630171e-08 1.8082093e-06 8.8792412e-10 2.5261832e-08 1.5380074e-05]]
```

```
import numpy as np
print(np.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.]]
```

SAVE THE MODEL

```
model.save('model.h5')
```

TEST WITH SAVED MODEL

```
from tensorflow.keras.models import load_model
model = load_model(r'/content/model.h5')
from PIL import Image
import numpy as np
for index in range(4):
 img = Image.open('/content/sample_data/digit.png' + str(index) + '.png').convert("L")
 img = img.resize((28,28))# resizing of input image
 im2arr- np.array(img) #converting to image
 im2arr = im2arr.reshape(1,28,28,1) #reshaping according to our requirement
#Predicting the Test set results
y_pred = model.predict(im2arr) #predicting the results
print(y_pred)
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-28-87cd2cca60f1> in <module>
      4 import numpy as np
      5 for index in range(4):
----> 6  img = Image.open('/content/sample_data/digit.png' + str(index) +
'.png').convert("L")
      7  img = img.resize((28,28))# resizing of input image
      8  im2arr- np.array(img) #converting to image

/usr/local/lib/python3.7/dist-packages/PIL/Image.py in open(fp, mode)
   2841
   2842     if filename:
-> 2843         fp = builtins.open(filename, "rb")
   2844         exclusive_fp = True
   2845

FileNotFoundError: [Errno 2] No such file or directory:
'/content/sample_data/digit.png0.png'
```