```
!pip install tensorflow --upgrade
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: tensorflow in /usr/local/lib/python3.7/dist-packages (2.9.2)
Collecting tensorflow
  Downloading tensorflow-2.11.0-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (588.3 MB)
     |████████████████████████████████| 588.3 MB 7.2 kB/s
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.3.0)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (3.3.0)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.14.1)
Collecting tensorboard<2.12,>=2.11
  Downloading tensorboard-2.11.0-py3-none-any.whl (6.0 MB)
     |████████████████████████████████| 6.0 MB 57.5 MB/s
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (0.4.0)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.6.3)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (2.1.0)
Collecting tensorflow-estimator<2.12,>=2.11.0
  Downloading tensorflow_estimator-2.11.0-py2.py3-none-any.whl (439 kB)
     |████████████████████████████████| 439 kB 82.6 MB/s
Requirement already satisfied: protobuf<3.20,>=3.9.2 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (3.19.6)
Collecting flatbuffers>=2.0
  Downloading flatbuffers-22.10.26-py2.py3-none-any.whl (26 kB)
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (4.1.1)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.15.0)
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (3.1.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages (from tensorflow) (57.4.0)
Collecting keras<2.12,>=2.11.0
  Downloading keras-2.11.0-py2.py3-none-any.whl (1.7 MB)
     |████████████████████████████████| 1.7 MB 59.5 MB/s
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.50.0)
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (14.0.6)
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages (from tensorflow) (21.3)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.21.6)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (0.27.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.7/dist-packages (from astunparse>=1.6.0->tensorflow) (0.38.3)
```

```
Requirement already satisfied: cached-property in /usr/local/lib/python3.7/
dist-packages (from h5py>=2.9.0->tensorflow) (1.5.2)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /us
r/local/lib/python3.7/dist-packages (from tensorboard<2.12,>=2.11->tensorfl
ow) (0.6.1)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local/
lib/python3.7/dist-packages (from tensorboard<2.12,>=2.11->tensorflow) (1.8
.1)
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/pyth
on3.7/dist-packages (from tensorboard<2.12,>=2.11->tensorflow) (2.14.1)
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.7/
dist-packages (from tensorboard<2.12,>=2.11->tensorflow) (1.0.1)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.7/
dist-packages (from tensorboard<2.12,>=2.11->tensorflow) (3.4.1)
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python
3.7/dist-packages (from tensorboard<2.12,>=2.11->tensorflow) (2.23.0)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/loc
al/lib/python3.7/dist-packages (from tensorboard<2.12,>=2.11->tensorflow) (
0.4.6)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/pyth
on3.7/dist-packages (from google-auth<3,>=1.6.3->tensorboard<2.12,>=2.11->t
ensorflow) (0.2.8)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/pyt
hon3.7/dist-packages (from google-auth<3,>=1.6.3->tensorboard<2.12,>=2.11->
tensorflow) (5.2.0)
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.7/di
st-packages (from google-auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflo
w) (4.9)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/p
ython3.7/dist-packages (from google-auth-oauthlib<0.5,>=0.4.1->tensorboard<
2.12,>=2.11->tensorflow) (1.3.1)
Requirement already satisfied: importlib-metadata>=4.4 in /usr/local/lib/py
thon3.7/dist-packages (from markdown>=2.6.8->tensorboard<2.12,>=2.11->tenso
rflow) (4.13.0)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-p
ackages (from importlib-metadata>=4.4->markdown>=2.6.8->tensorboard<2.12,>=
2.11->tensorflow) (3.10.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/pytho
n3.7/dist-packages (from pyasn1-modules>=0.2.1->google-auth<3,>=1.6.3->tens
orboard<2.12,>=2.11->tensorflow) (0.4.8)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dis
t-packages (from requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow)
(2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.
7/dist-packages (from requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorf
low) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /
usr/local/lib/python3.7/dist-packages (from requests<3,>=2.21.0->tensorboar
d<2.12,>=2.11->tensorflow) (1.24.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3
.7/dist-packages (from requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensor
flow) (2022.9.24)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/
dist-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.
4.1->tensorboard<2.12,>=2.11->tensorflow) (3.2.2)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/p
ython3.7/dist-packages (from packaging->tensorflow) (3.0.9)
```

```
Installing collected packages: tensorflow-estimator, tensorboard, keras, fl
atbuffers, tensorflow
  Attempting uninstall: tensorflow-estimator
    Found existing installation: tensorflow-estimator 2.9.0
    Uninstalling tensorflow-estimator-2.9.0:
      Successfully uninstalled tensorflow-estimator-2.9.0
  Attempting uninstall: tensorboard
    Found existing installation: tensorboard 2.9.1
    Uninstalling tensorboard-2.9.1:
      Successfully uninstalled tensorboard-2.9.1
  Attempting uninstall: keras
    Found existing installation: keras 2.9.0
    Uninstalling keras-2.9.0:
      Successfully uninstalled keras-2.9.0
  Attempting uninstall: flatbuffers
    Found existing installation: flatbuffers 1.12
    Uninstalling flatbuffers-1.12:
      Successfully uninstalled flatbuffers-1.12
  Attempting uninstall: tensorflow
    Found existing installation: tensorflow 2.9.2
    Uninstalling tensorflow-2.9.2:
      Successfully uninstalled tensorflow-2.9.2
Successfully installed flatbuffers-22.10.26 keras-2.11.0 tensorboard-2.11.0
tensorflow-2.11.0 tensorflow-estimator-2.11.0
```

In [2]:

```
import numpy as np
import tensorflow #open source used for both ML and DL for computation
from tensorflow.keras.datasets import mnist #mnist dataset
from tensorflow.keras.models import Sequential #it is a plain stack of
layers
from tensorflow.keras import layers #A Layer consists of a tensor- in
tensor-out computat ion funct ion
from tensorflow.keras.layers import Dense, Flatten #Dense-Dense Layer is
the regular deeply connected r
#faltten -used fot flattening the input or change the dimension
from tensorflow.keras.layers import Conv2D #onvoLutiona l Layer
from keras.optimizers import Adam #opt imizer
from keras. utils import np_utils #used for one-hot encoding
import matplotlib.pyplot as plt   #used for data visualization
```

LOAD DATA

In [3]:

```
(x_train, y_train), (x_test, y_test)=mnist.load_data () #splitting the
mnist data into train and test
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-da
tasets/mnist.npz
11490434/11490434 [==============================] - 1s 0us/step
```

In [4]:

```
print (x_train.shape)  #shape is used for give the dimens ion values
#60000-rows 28x28-pixels
print (x_test.shape)
```

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

In [5]:

```
x_train[0]
```

Out[5]:

```
array([[  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],
       [  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,   3,
         18,  18,  18, 126, 136, 175,  26, 166, 255, 247, 127,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,  30,  36,  94, 154, 170,
        253, 253, 253, 253, 253, 225, 172, 253, 242, 195,  64,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,  49, 238, 253, 253, 253, 253,
        253, 253, 253, 253, 251,  93,  82,  82,  56,  39,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,  18, 219, 253, 253, 253, 253,
        253, 198, 182, 247, 241,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,  80, 156, 107, 253, 253,
        205,  11,   0,  43, 154,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,  14,   1, 154, 253,
         90,   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, 139, 253,
        190,   2,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  11, 190,
        253,  70,   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,  35,
        241, 225, 160, 108,   1,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         81, 240, 253, 253, 119,  25,   0,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,  45, 186, 253, 253, 150,  27,   0,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,  16,  93, 252, 253, 187,   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, 249, 253, 249,  64,   0,   0,   0,   0,   0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,  46, 130, 183, 253, 253, 207,   2,   0,   0,   0,   0,   0,
```

```
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  39,
 148, 229, 253, 253, 253, 250, 182,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  24, 114, 221,
 253, 253, 253, 253, 201,  78,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,   0,   0,  23,  66, 213, 253, 253,
 253, 253, 198,  81,   2,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,   0,   0,  18, 171, 219, 253, 253, 253, 253,
 195,  80,   9,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0,  55, 172, 226, 253, 253, 253, 253, 244, 133,
  11,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0],
[  0,   0,   0,   0, 136, 253, 253, 253, 212, 135, 132,  16,   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,   0,
   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0]], dtype=uint8)
```

In [6]:

```
plt.imshow(x_train[5100])     #ploting the index=image
```

Out[6]:

In [7]:

```
np.argmax(y_train[5100])
```

Out[7]:

```
0
```

RESHAPING DATASET

In [8]:

```
#Reshaping to format which CNN expects (batch, height, width, channels)
x_train=x_train.reshape (60000, 28, 28, 1).astype('float32')
x_test=x_test.reshape (10000, 28, 28, 1).astype ('float32')
```

APPLY ONE HOT ENCODING

In [9]:

```
number_of_classes = 10  #storing the no of classes in a variable
y_train = np_utils.to_categorical (y_train, number_of_classes) #converts
the output in binary format
y_test = np_utils.to_categorical (y_test, number_of_classes)
```

ADD CNN LAYER

In [10]:

```
#create model
model=Sequential ()
#adding modeL Layer
model.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1), activation='relu'))
```

```
model.add(Conv2D(32, (3, 3), activation = 'relu'))
#flatten the dimension of the image
model.add(Flatten())
#output layer with 10 neurons
model.add(Dense(number_of_classes,activation = 'softmax'))
```

COMPILE MODEL

In [11]:

```
#Compile model
model.compile(loss= 'categorical_crossentropy', optimizer="Adam",
metrics=['accuracy'])
x_train = np.asarray(x_train)
y_train = np.asarray(y_train)
```

TRAIN THE MODEL

In [12]:

```
model.fit(x_train, y_train, validation_data=(x_test, y_test), epochs=5,
batch_size=32)
```

```
Epoch 1/5
1875/1875 [==============================] - 169s 90ms/step - loss: 0.2799
- accuracy: 0.9426 - val_loss: 0.0934 - val_accuracy: 0.9723
Epoch 2/5
1875/1875 [==============================] - 163s 87ms/step - loss: 0.0754
- accuracy: 0.9771 - val_loss: 0.0868 - val_accuracy: 0.9731
Epoch 3/5
1875/1875 [==============================] - 160s 85ms/step - loss: 0.0518
- accuracy: 0.9842 - val_loss: 0.0730 - val_accuracy: 0.9780
Epoch 4/5
1875/1875 [==============================] - 153s 81ms/step - loss: 0.0409
- accuracy: 0.9872 - val_loss: 0.0901 - val_accuracy: 0.9742
Epoch 5/5
1875/1875 [==============================] - 155s 83ms/step - loss: 0.0286
- accuracy: 0.9911 - val_loss: 0.1128 - val_accuracy: 0.9750
```

Out[12]:

OBSERVING THE METRICES

In [13]:

```
# Final evaluation of the model
metrics = model.evaluate(x_test, y_test, verbose=0)
print("Metrics (Test loss &Test Accuracy) : ")
print(metrics)
```

```
Metrics (Test loss &Test Accuracy) :
[0.11275885254144669, 0.9750000238418579]
```

In [ ]:

```
prediction=model.predict(x_test[6000:6001])
print(prediction)
plt.imshow(x_test[6000])
```

In [ ]:

```
import numpy as np
print(np.argmax(prediction, axis=1)) #printing our Labels from first 4
images
```

In [ ]:

```
np.argmax(y_test[6000:6001]) #printing the actual labels
```

SAVE THE MODEL

In [ ]:

```
# Save the model
model.save('models/mnistCNN.h5')
cd models
```

In [ ]:

```
!tar -zcvf hdr_deployment.tgz mnistCNN.h5
```

In [ ]:

```
ls -1
```

In [ ]:

```
!pip install watson-machine-learning-client --upgrade
```

CLOUD DEPLOY

In [ ]:

```
from ibm_watson_machine_learning import APIClient
credentials ={
    "url":"https://us-south.ml.cloud.ibm.com",
    "apikey":"Qxwy3byu83al_Lvmk05S2xcRhHqeQiy_4BxWzPcxuB9A"
}
client = APIClient(credentials)
client
```

In [ ]:

```
client.spaces.get_details()
```

In [ ]:

```
def guid_from_space_name(client,deploy):
  space = client.spaces.get_details()
  return (next(item for item in space['resources'] if
item['entity']['name']==deploy)['metadata']['id'])
```

In [ ]:

```
space_uid = guid_from_space_name(client,'hdr')
print("Space UID = " + space_uid)
```

In [ ]:

```
client.set.default_space(space_uid)
```

In [ ]:

```
client.software_specifications.list(limit=100)
```

In [ ]:

```
software_space_uid =
client.software_specifications.get_uid_by_name('tensorflow_rt22.1-py3.9')
software_space_uid
```

In [ ]:

```
model_details =
client.repository.store_model(model='hdr_deployment.tgz',meta_props={
    client.repository.ModelMetaNames.NAME:"Digit Recognition System",
    client.repository.ModelMetaNames.TYPE:"tensorflow_2.7",
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_space_uid
})
```

In [ ]:

```
model_details
```

In [ ]:

```
model_id = client.repository.get_model_id(model_details)
model_id
```

In [ ]:

```
client.repository.download(model_id,'DigitRecog_IBM_model.tar.gz')
```

In [ ]:

```
ls
```

TEST THE MODEL

In [ ]:

```
from tensorflow.keras.models import load_model
from keras.preprocessing import image
from PIL import Image
import numpy as np
```

In [ ]:

```
model = load_model("mnistCNN.h5")
```

In [ ]:

```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It
includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='1rEQ4QsDyr45SbIYkkmEXGolFpDjMBjlc1KmxrsH2V1U',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-
storage.appdomain.cloud')

bucket = 'digitrecognition-donotdelete-pr-kvpefjqsoxebrc'
object_key = '4.jpg'

streaming_body_3 = cos_client.get_object(Bucket=bucket,
Key=object_key)['Body']
```

In [ ]:

```
img = Image.open(streaming_body_3).convert("L") # convert image to
monochrome
img = img.resize( (28,28) ) # resizing of input image
```

In [ ]:

```
img
```

In [ ]:

```
im2arr = np.array(img) #converting to image
im2arr = im2arr.reshape(1, 28, 28, 1) #reshaping according to our
requirement
pred = model.predict(im2arr)
print(pred)
```

In [ ]:

```
print(np.argmax(pred, axis=1)) #printing our Labels
```