

```

#TEAM ID:PNT2022TMID33471
#Importing required Libraries

import numpy#for numerical analysis
import tensorflow#open source ml tool by google

from tensorflow.keras.datasets import mnist #mnist dataset
from tensorflow.keras.models import Sequential# stack for layers
from tensorflow.keras import layers#input,middle and output layers
forcnn structure

from tensorflow.keras.layers import Dense,Flatten#dense and flatten
layers
from tensorflow.keras.layers import Conv2D#convolutional layers
from tensorflow import keras#library for building neural networks
built on tensorflow

from tensorflow.keras.optimizers import Adam#optimizers
from keras.utils import np_utils

#Loading dataset

Dataset is available in tensorflow dataset repository

(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

automatically data is splitted for train ,test -70:30 ratio

print(x_train.shape)
print(y_train.shape)

(60000, 28, 28)
(60000,)

Training Dataset has 60000 images & testing has 10000 images

print(x_test.shape)
print(y_test.shape)

(10000, 28, 28)
(10000,)

```

Analyze the data

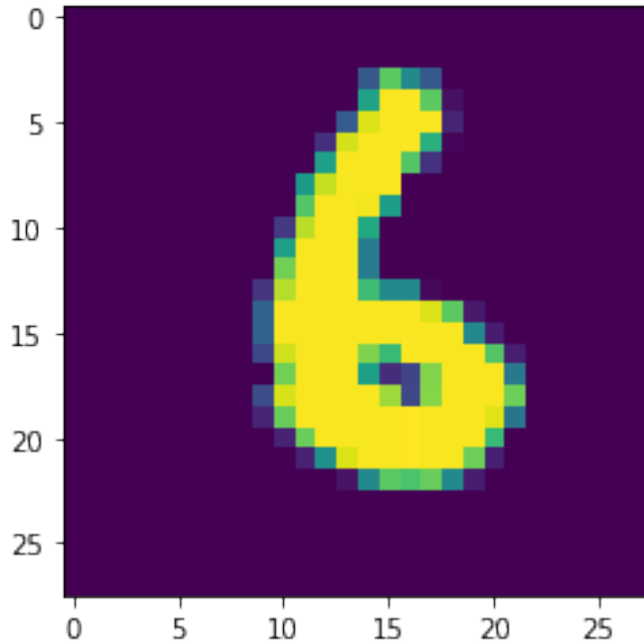
```
x_train[3]
```

```

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,  0,  0,  0,  0,  0,
0,
        0,  0,  0,  0,  0, 124, 253, 255, 63,  0,  0,  0,
0,
        0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
0,
        0,  0,  0,  0, 96, 244, 251, 253, 62,  0,  0,  0,
0,
        0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
0,
        0,  0,  0,  0, 127, 251, 251, 253, 62,  0,  0,  0,
0,
        0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
0,
        0,  0,  0, 68, 236, 251, 211, 31,  8,  0,  0,  0,
0,
        0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
0,
        0,  0, 60, 228, 251, 251, 94,  0,  0,  0,  0,  0,
0,
        0,  0],

```


<matplotlib.image.AxesImage at 0x7fa70fb3b550>



#Reshaping the data

,As we are using Deep learning neural network, the input for this network to get trained on should be of higher dimensional. Our dataset is having three-dimensional images so we have to reshape them too higher dimensions

#(batch,height,width,channel)

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

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

#Applying one hot encoding

One hot encoding to convert numerical values to classes where 0 to 9 are 10 separate classes if value is 5 class 5 is 1 else 0

`no_of_classes=10`

`y_train=np_utils.to_categorical(y_train,no_of_classes)`

`y_test=np_utils.to_categorical(y_test,no_of_classes)`

`y_test[3]`

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