Reshaping the Data

\*Importing the Required libraries\*

from keras.datasets import mnist

import matplotlib.pyplot as plt

from keras.utils import np\_utils

\*Load the Data\*

(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)

\*Analyzing the Data\*

x\_train[1]

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, 51, 159, 253, 159, 50, 0, 0, 0, 0, 0, 0,

0, 0],

[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 48, 238, 252, 252, 252, 237, 0, 0, 0, 0, 0, 0,

0, 0],

[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

54, 227, 253, 252, 239, 233, 252, 57, 6, 0, 0, 0, 0,

0, 0],

[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 10, 60,

224, 252, 253, 252, 202, 84, 252, 253, 122, 0, 0, 0, 0,

0, 0],

[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 163, 252,

252, 252, 253, 252, 252, 96, 189, 253, 167, 0, 0, 0, 0,

0, 0],

[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 51, 238, 253,

253, 190, 114, 253, 228, 47, 79, 255, 168, 0, 0, 0, 0,

0, 0],

[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 48, 238, 252, 252,

179, 12, 75, 121, 21, 0, 0, 253, 243, 50, 0, 0, 0,

0, 0],

[ 0, 0, 0, 0, 0, 0, 0, 0, 38, 165, 253, 233, 208,

84, 0, 0, 0, 0, 0, 0, 253, 252, 165, 0, 0, 0,

0, 0],

[ 0, 0, 0, 0, 0, 0, 0, 7, 178, 252, 240, 71, 19,

28, 0, 0, 0, 0, 0, 0, 253, 252, 195, 0, 0, 0,

0, 0],

[ 0, 0, 0, 0, 0, 0, 0, 57, 252, 252, 63, 0, 0,

0, 0, 0, 0, 0, 0, 0, 253, 252, 195, 0, 0, 0,

0, 0],

[ 0, 0, 0, 0, 0, 0, 0, 198, 253, 190, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 255, 253, 196, 0, 0, 0,

0, 0],

[ 0, 0, 0, 0, 0, 0, 76, 246, 252, 112, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 253, 252, 148, 0, 0, 0,

0, 0],

[ 0, 0, 0, 0, 0, 0, 85, 252, 230, 25, 0, 0, 0,

0, 0, 0, 0, 0, 7, 135, 253, 186, 12, 0, 0, 0,

0, 0],

[ 0, 0, 0, 0, 0, 0, 85, 252, 223, 0, 0, 0, 0,

0, 0, 0, 0, 7, 131, 252, 225, 71, 0, 0, 0, 0,

0, 0],

[ 0, 0, 0, 0, 0, 0, 85, 252, 145, 0, 0, 0, 0,

0, 0, 0, 48, 165, 252, 173, 0, 0, 0, 0, 0, 0,

0, 0],

[ 0, 0, 0, 0, 0, 0, 86, 253, 225, 0, 0, 0, 0,

0, 0, 114, 238, 253, 162, 0, 0, 0, 0, 0, 0, 0,

0, 0],

[ 0, 0, 0, 0, 0, 0, 85, 252, 249, 146, 48, 29, 85,

178, 225, 253, 223, 167, 56, 0, 0, 0, 0, 0, 0, 0,

0, 0],

[ 0, 0, 0, 0, 0, 0, 85, 252, 252, 252, 229, 215, 252,

252, 252, 196, 130, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0],

[ 0, 0, 0, 0, 0, 0, 28, 199, 252, 252, 253, 252, 252,

233, 145, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0],

[ 0, 0, 0, 0, 0, 0, 0, 25, 128, 252, 253, 252, 141,

37, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0],

[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0],

[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0],

[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

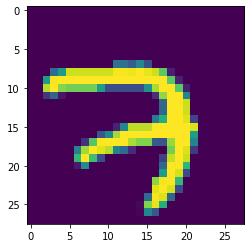
0, 0],

[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

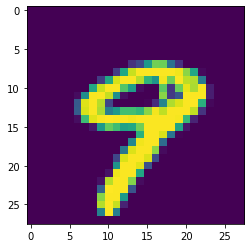
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0]], dtype=uint8)

plt.imshow(x\_train[5000])



plt.imshow(x\_train[45])



print(x\_train.shape)

print(y\_train.shape)

(60000, 28, 28)

(60000,)

import numpy as np

import tensorflow

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

\*Reshaping the Dataset\*

np.argmax(y\_train[5000])

0

\*Applying one Hotcode\*

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)