Applying one Hot Encoding

import cv2

import numpy as np

from keras.datasets import mnist

from keras.layers import Dense, Flatten, MaxPooling2D, Dropout

from keras.layers.convolutional import Conv2D

from keras.models import Sequential

from keras.utils import to\_categorical

import matplotlib.pyplot as plt

(X\_train, y\_train), (X\_test, y\_test) = mnist.load\_data()

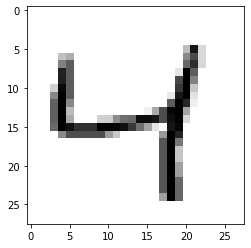
Downloading data from <https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz>

11490434/11490434 [==============================] - 2s 0us/step

plt.imshow(X\_train[2], cmap="binary")

plt.show()

print (y\_train[2])



4

print("---Before reshaping the Data---")

print("Shape of X\_train: {}".format(X\_train.shape))

print("Shape of y\_train: {}".format(y\_train.shape))

print("Shape of X\_test: {}".format(X\_test.shape))

print("Shape of y\_test: {}".format(y\_test.shape))

print("-----------------------------------------")

print()

print("---After reshaping the Data---")

X\_train = X\_train.reshape(60000, 28, 28, 1)

X\_test = X\_test.reshape(10000, 28, 28, 1)

print("Shape of X\_train: {}".format(X\_train.shape))

print("Shape of y\_train: {}".format(y\_train.shape))

print("Shape of X\_test: {}".format(X\_test.shape))

print("Shape of y\_test: {}".format(y\_test.shape))

print("-----------------------------------------")

---Before reshaping the Data---

Shape of X\_train: (60000, 28, 28, 1)

Shape of y\_train: (60000,)

Shape of X\_test: (10000, 28, 28, 1)

Shape of y\_test: (10000,)

-----------------------------------------

---After reshaping the Data---

Shape of X\_train: (60000, 28, 28, 1)

Shape of y\_train: (60000,)

Shape of X\_test: (10000, 28, 28, 1)

Shape of y\_test: (10000,)

-----------------------------------------

y\_train = to\_categorical(y\_train)

y\_test = to\_categorical(y\_test)

print("Shape of y\_train: {}".format(y\_train.shape))

print("Shape of y\_train: {}".format(y\_test.shape))

Shape of y\_train: (60000, 10)

Shape of y\_train: (10000, 10)