{

"nbformat": 4,

"nbformat\_minor": 0,

"metadata": {

"colab": {

"provenance": []

},

"kernelspec": {

"name": "python3",

"display\_name": "Python 3"

},

"language\_info": {

"name": "python"

}

},

"cells": [

{

"cell\_type": "code",

"execution\_count": 1,

"metadata": {

"id": "gy1P-3IuG7o3"

},

"outputs": [],

"source": [

"import cv2\n",

"import numpy as np\n",

"from keras.datasets import mnist\n",

"from keras.layers import Dense, Flatten, MaxPooling2D, Dropout\n",

"from keras.layers.convolutional import Conv2D\n",

"from keras.models import Sequential\n",

"from tensorflow.keras.utils import to\_categorical\n",

"import matplotlib.pyplot as plt"

]

},

{

"cell\_type": "code",

"source": [

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

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "tl9VGKmZHLbb",

"outputId": "bfa1570f-a282-483d-d3b4-58a54bdb4209"

},

"execution\_count": 2,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz\n",

"11490434/11490434 [==============================] - 0s 0us/step\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"plt.imshow(X\_train[0], cmap=\"gray\")\n",

"plt.show()\n",

"print (y\_train[0])"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 282

},

"id": "t3JZPOT-HK9O",

"outputId": "7f4f5e35-0eef-4439-d8d0-bd17b203e06f"

},

"execution\_count": 3,

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {

"needs\_background": "light"

}

},

{

"output\_type": "stream",

"name": "stdout",

"text": [

"5\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

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

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

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

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

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "5G1EiiRIHfkB",

"outputId": "beca6bad-84e1-4d96-c87b-0ad3e2ea09d6"

},

"execution\_count": 4,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Shape of X\_train: (60000, 28, 28)\n",

"Shape of y\_train: (60000,)\n",

"Shape of X\_test: (10000, 28, 28)\n",

"Shape of y\_test: (10000,)\n"

]

}

]

},

{

"cell\_type": "markdown",

"source": [

"# Reshaping so as to covert images for our model "

],

"metadata": {

"id": "gjjhMtimHrze"

}

},

{

"cell\_type": "code",

"source": [

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

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

],

"metadata": {

"id": "FlSNwuWAIBWG"

},

"execution\_count": 5,

"outputs": []

},

{

"cell\_type": "code",

"source": [

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

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

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

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

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "bQ1G6d20ITf0",

"outputId": "545f5dc5-de23-48c4-f4d2-b9e80c7b1612"

},

"execution\_count": 6,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

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

"Shape of y\_train: (60000,)\n",

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

"Shape of y\_test: (10000,)\n"

]

}

]

}

]

}