

```
{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "provenance": []
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    },
    "language_info": {
      "name": "python"
    }
  },
  "cells": [
    {
      "cell_type": "code",
      "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"
      ],
      "metadata": {
        "id": "yJs2eLRLOSHM"
      },
      "execution_count": 1,
      "outputs": []
    },
  ],
}
```

```

{
  "cell_type": "code",
  "source": [
    "(X_train, y_train), (X_test, y_test) = mnist.load_data()"
  ],
  "metadata": {
    "id": "-NoTriNEPBWI"
  },
  "execution_count": 3,
  "outputs": []
},
{
  "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": "s_JYjDk2PGwL",
    "outputId": "b42ae095-36e0-4014-80ac-afabaead8ec7"
  },
  "execution_count": 4,
  "outputs": [
    {
      "output_type": "display_data",
      "data": {
        "text/plain": [
          "<Figure size 432x288 with 1 Axes>"
        ]
      }
    ]
  }
}

```

IVBORw0KGgoAAAAANSUeUGaAAAPsAAAD4CAYAAAAQ5pAIAAAABHNCSVQICAgfAhkiAAAAIwSFlzAAALEgAACl
 B0t1+/AAAADh0RVh0U29mdHdhcmUAAbWF0cGxvdGxpYiB2ZXJzaW9uMy4yLjIsIGh0dHA6Ly9tYXRwbG90bGliLm9y
 Zy+WH4yJAAAN9klEQVR4nO3df4xV9ZnH8c+zWP6QojBrOhKKSyEGg8ZON4gbl6w1hvojGhw1TSexoZE4/YNJaLiHNe
 wf1WwwZBU2S2TNTKMWNl1qEzUgaQouoOzGhDgiKo5LdQ2mTEaowZef/mCHefaPezBTnfu9w7nn3nOZ5/1Kbu69
 57nnnncnfDi/7pnmvubsATH5/VXYDAJqDsANBEHYgCMIOBEHYgSAuaubCzIXT/0CDubuNN72uLbuZ3Wpnmh8zsPTN7sJ
 7vAtBYlvc6u5lNkfRHSUsIHZH0qqQudx9IzMOWHWiwRmzZF0t6z93fd/czkn4raVkd3weggeoj+2xJfxyz/kg27S+YWbe
 Z9ZtZfx3LAlCnHP+gc/c+SX0Su/FamerZsg9KmjPm/bezaQBauD1hf1XSIWb2HTObKulHkrYV0xaAouXejXf3ETPrkbRD0
 hRJT7n724V1BqBQuS+95VoYx+xAwzXkRzUALhyEHQICsANBEHYgCMIOBEHYgSAIOxAEYQeCIOxAEIQdCIKwA0EQdi
 Alwg4EQdiBIAg7EARhB4lg7EAQhB0lgrADQRB2IAjCDgRB2IEgCDsQBGEHgiDsQBCEHqiCsANBEHYgCMIOBJF7yGZcG
 KZMMZKSX3rppQ1dfk9PT9XaxRdfnJx3wYIFyfrKISuT9ccee6xqraurKznv559/nqyvW7cuWX/44YeT9TLUFXYzOyzppKS
 zkkbcfVERTQEOxHfB9pvc/aMCvgdAA3HMDgRRb9hd0k4ze83Musf7gJl1m1m/mfXXuSwAdah3N36Juw+a2bckvWh
 m/+Pue8d+wN37JPVJkpl5ncsDkFNdW3Z3H8yej0l6XtLiipoCULzcYTezaWY2/dxrST+QdLCoxGAUq57d+HZJz5vZue/5
 D3f/QyFdTTJXXHFFsj516tRk/YYbbkjWlyxZuR2Y8aM5Lz33HNpSl6ml0eOJOsbn25M1js7O6vWTP48mZz3jTfeSNZff
 vnlZLOV5Q67u78v6bsF9gKggbj0BgRB2IEgCDsQBGEHgiDsQBDM3rwtU3WX9B1dHqk67t3707WG32baasaHR1N1u
 +///5k/dSpU7mXPTQ0IKx//PHHyfQhQ4dyL7vR3N3Gm86WHQICsANBEHYgCMIOBEHYgSAIOxAEYQeC4Dp7Adra2pL
 1ffv2Jevz5s0rsp1C1ep9eHg4Wb/pppuq1s6cOZOcN+rvD+rFdXYgOMIOBEHYgSAIOxAEYQeCIOxAEIQdCIhmwtw/Pjx
 ZH316tXJ+h133JGsv/7668l6rT+pnHLglwIFkfenSpcn66dOnk/Wrr766am3VqIXJeVestuxAEIQdCIKwA0EQdiAlwg4EQdi
 BIAg7EAT3s7eASy65JfMvNbxwb29v1dqKFSuS8953333J+pYtW5J1tJ7c97Ob2VNmdszMDo6Z1mZmL5rZu9nzzCKbB
 VC8iezG/1rSrV+Z9qCkXe5+paRd2XsAlaxm2N19r6Sv/h50maRN2etNku4quC8ABcv72/h2dz83WNaHktqrfdDMuiV1
 51wOgILUfSOMu3vqxJu790nqkzhBB5Qp76W3o2Y2S5Ky52PftQSGfKfZuk5dnr5ZK2FtMOgEapuRtvZlskFV/SZWZ2
 RNlVJK2T9DsZWyHpA0k/bGSTk92JEyfqmv+TTz7JPe8DDzyQRD/zzDPJeq0x1tE6aobd3buqlG4uuBcAdcTPZYEgCDsQB
 GEHgiDsQBCEHqiCW1wngWnTPlWtVfDCC8l5b7zxxmT9tttuS9Z37tyZrKP5GLIZCI6wA0EQdiAlwg4EQdiBIAg7EARhB
 4LgOvskN3/+GR9//79yfrw8HCyvmfPnmS9v7+/au2Jj55lztvMf5uTCdfZgeAlOxAEYQeCIOxAEIQdCIKwA0EQdiAlrrM
 H19nZmaw//fTtyfr06dNzL3vNmjXJ+ubNm5P1oaGhZD0qrrMDWRf2IAjCDgRB2IEgCDsQBGEHgiDsQBBC20fSNddck6
 xv2LAhWb/55vyD/fb29ibra9euTdYHBwdzL/tClvs6u5k9ZWbHzOzgmGkPmdmgmR3IHrcX2SyA4k1kN/7Xkm4dZ/q/u
 ntH9vh9sW0BKFrNsLv7XknHm9ALgAaq5wRdj5m9me3mz6z2ITPrNrN+M6v+x8gANFzesP9S0nxJHZKGJK2v9kF373P
 3Re6+KOeyABQgV9jd/ai7n3X3UUm/krS42LYAFC1X2M1s1pi3nZIOVvssgNZQ8zq7mW2R9H1Jl0k6KukX2fsOSS7psK
 SfunvNm4u5zj75zJgx1m/8847q9Zq3StvNu7l4i/t3r07WV+6dGmyPIlVu85+OQRm7Bpn8pN1dwSgqfi5LBAEYQeCIOx
 AEIQdCIKwA0FwiytK88UXxyTrF12Uvlg0MjKsRn9yyy1Vay+99Fjy3gsZf0oaCl6wA0EQdiAlwg4EQdiBIAg7EARhB4Koe
 dcbYrv22muT9XvvvTdZv+6666rWal1Hr2VgYCBZ37t3b13fP9mwZQeCIOxAEIQdCIKwA0EQdiAlwg4EQdiBILjOPsktW
 LAgWe/p6UnW77777mT98ssvP++eJurs2bPJ+tbQ+q+Xj46OfnOBY8tOxAEYQeCIOxAEIQdCIKwA0EQdiAlwg4EwXX
 2C0Cta9ldXeMntFtR6zr63LlZ87RUIp7+/mR97dq1yfQ2bduKbGfSq7lIN7M5ZrbHzAbM7G0zW5VNbzOzF83s3ex5ZuP
 bBZDXRHbjRyT9o7svlPR3klaa2UJJD0ra5e5XStqVvQfQomqG3d2H3H1/9vqkpHckzZa0TNKm7G0bJN3VqCYB1O+8jt
 nNbK6k70naJ6nd3c/9OPIDSe1V5umW1J2/RQBfMPDZeDP7pqRnJf3M3U+MrXlIdMhxB2109z53X+Tui+rqFEBdJhR2
 M/uGkKH/jbs/l00+amazsvosScca0yKAltTcjTczk/SkpHfcfcOY0jZjyyWty563NqTDSaC9fdwJnC8tXLgWwX/88ceT9auu
 uuq8eyrKvn37kvVHH320am3r1vQ/GW5RLdZEjtn/XtKPJb1lZgeyaWtUCfnvzGyFpA8k/bAxLQlOqs2wu/t/Sxp3cHdJN
 xfbDoBG4eeyQBCEHqiCsANBEHYgCMIOBMETrhPU1tZWtdbb25uct6Ojl1mfN29erp6K8MorryTr69evT9Z37NiRrH/2
 2Wfn3RMagy07EARhB4lg7EAQhB0lgrADQRB2IAjCDgQR5jr79ddfn6yVxR06WV+8eHHV2uzZs3P1VJRPP/20am3jxo3
 JeR955JfK/fTp07l6QuthyW4EQdiBIAg7EARhB4lg7EAQhB0lgrADQYS5zt7Z2VIXvR4DAwPJ+vbt25P1kZGRZD11z/nw8
 HByXsTBlh0lgrADQRB2IAjCDgRB2IEgCDsQBGEHgiB3T3/Abi6kzZLaJbmkPnf/NzN7SNIDkv6cfXSnu/++xnefWagbu4+
 7qjLEwn7LEmz3H2/mU2X9Jqu1QZj/2Uuz820SYIO9B41cl+kfHZhyQNZA9Pmtk7ksr90ywAzt5HbOb2VxJ35O0L5vU
 Y2ZvmtlTzjazyjzdZtZvZv11dQqgJlV347/8oNk3Jb0saa27P2dm7ZI+UuU4/p9V2dW/v8Z3sBsPNFjuY3ZJMrNvSNouaY
 e7bxinPlfSdne/psb3EHagwaqFveZuvJmZpCclvTM26NmJu3M6JR2st0kAjTORs/FLJP2XpLckjWaT10jqktShym78YUk/
 zU7mpb6LLTvQYHXtxheFsAONI3s3HsDkQNiBIAg7EARhB4lg7EAQhB0lgrADQRB2IAjCDgRB2IEgCDsQBGEHgiDsQBC
 EHQii2UM2fyTpgzHvL8umtaJW7a1V+5LoLa8ie/ubaoWm3s/+tYWb9bv7otlaSGjV3lq1L4ne8mpWb+zGA0EQdiClssP
 eV/LyU1q1t1btS6K3vJrSW6nH7ACap+wtO4AmlexAEKWE3cxuNbNDZvaemT1YRg/VmNlhM3vLzA6UPT5dNobeMT
 M7OGZam5m9aGbvZs/jjrFXUm8Pmdlgtu4OmNntJfU2x8z2mNmAmb1tZquy6aWuuORfTVlvTT9mN7Mpkv4oaamkl
 5JeldTl7gNNbaQKMzssaZG7l/4DDDP7B0mnJG0+N7SWmf2LpOPuvi77j3Kmu/+8RXp7SOc5jHeDeqs2zPhPVOK6K3L
 48zzK2LlVlvSeu7/v7mck/VbSshL6aHnuvlfS8a9MXiZpU/Z6kyr/WJquSm8twd2H3H1/9vqkpHPDje67hJ9NUUZYZ8t6
 U9j3h9Ra4337pj2mtlrZtZddjPjaB8zzNaHktrLbGYcNYfbqavDDPeMusuz/Dn9eIE3dctcfe/IXSbpJXZ7mpL8soxWctdO
 /2lpPmqiAE4JlG9mc1kw4w/K+ln7n5ibK3MdTdoX01Zb2WefVDSnDHvv51NawnuPpg9H5P0vCqHHa3k6LkRdLpNyY

X38yV3P+ruZ919VnKvVOK6y4YZf1bSb9z9uWxy6etuvL6atd7KCPurkq40s++Y2VRJP5K0rYQ+vsbMpmUnTmRm0yT9
QK03FPU2Scuz18slbS2xl7/QKsN4VxtmXCWvu9KHP3f3pj8k3a7KGfn/lfrPZfRQpa95kt7IHm+X3ZukLars1v2fKuc2Vkj
6a0m7JL0r6T8ltbVQb/+uytDeb6oSrFkl9bZEIV30NyUdyB63l73uEn01Zb3xc1kgCE7QAUEQdiAlwg4EQdiBIAg7EARhB
4lg7EAQ/w8ie3GmjcGk5QAAAABJRu5ErkJggg==\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": "Us6HotvxPPco",
    "outputId": "df6fdfe3-e423-4c9a-9354-bad2e9ff1a7e"
  },
  "execution_count": 5,
  "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": "code",
  "source": [
    "# Reshaping so as to convert images for our model\n",
    "X_train = X_train.reshape(60000, 28, 28, 1)\n",
    "X_test = X_test.reshape(10000, 28, 28, 1)"
  ],
  "metadata": {
    "id": "n962FFkFPUzH"
  },
  "execution_count": 6,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "print (\nShape of X_train: {}".format(X_train.shape))\n",
    "print (\nShape of y_train: {}".format(y_train.shape))\n",
    "print (\nShape of X_test: {}".format(X_test.shape))\n",
    "print (\nShape of y_test: {}".format(y_test.shape))"
  ],
  "metadata": {

```

```
"colab": {
  "base_uri": "https://localhost:8080/"
},
"id": "KZEPs_m5PZac",
"outputId": "97de7d4c-aedf-4c45-f3e9-5a7942778088"
},
"execution_count": 7,
"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"
    ]
  }
]
```