```
{
  "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": "iMDzDhZb0z-_"
      },
      "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 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": "RbzxozmQ1SLE",
        "outputId": "1bea9411-5b4f-4600-ac87-fe14d7c666cf"
      },
      "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": "-20tJf-d1VJT",
        "outputId": "3af54e4e-e232-40c1-df16-0d073dc6d0b3"
      },
      "execution_count": 3,
      "outputs": [
        {
          "output_type": "display_data",
          "data": {
            "text/plain": [
              "<Figure size 432x288 with 1 Axes>"
            ],
            "image/png":
"iVBORw0KGgoAAAANSUhEUgAAAPsAAAD4CAYAAAAq5pAIAAAABHNCSVQICAgIfAhkiAAAAAlw
SFlzAAALEgAACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9uMy4yL
jIsIGh0dHA6Ly9tYXRwbG90bGliLm9yZy+WH4yJAAAN9klEQVR4nO3df4xV9ZnH8c+zWP6Qoj
BrOhKKSyEGg8ZON4gbl6w1hvojGhw1TSexoZE4/YNJaLIhNewf1WwwZBU2SzTNTKMWNl1qEzU
gaQouoOzGhDgiKo5LdQ2mTEaowZEf/mCHefaPezBTnfu9w7nn3nOZ5/1Kbu6957nnnnicnfDi/
7pmvubbsATH5/VXYDAJqDsANBEHYgCMIOBEHYgSAuaubCzIxT/0CDubNN72uLbuZ3WpmR83sQzN
TN7sJ7vAtBYlvc6u5lNkfRHSUlHZH0qV0xaAouXejXf3ETPrkbkRD0hRJT7n724z+95VoYx+xA
wzXkRzUALhyEHQiCsANBEHYgCMIOBEHYgSAuaoCIOxAEYQeCIOxAEIQdCIKwA0EQdiBI
Ag7EARhB4Ig7EAQhB0IgrADQRB2IAjCDgRB2IEgCDsQBGEHgiDsQBCEHQiCsANBEHYgCMIOBJ
F7yGZcGKZMmZKsX3rppQ1dfk9PT9XaxRdfnJx3wYIFyfrKlSuT9ccee6xq7euurKznv559/nqy
vW7cuWX/44YeT9TLUFXYzOyppkppKSzkbbcfVERb9pvhbcfVERTQEoXhFb9pvc/aMCvgdAA3HMDgRRb9hd0k4w4z
e83Musf7gJl1m1m/mfXXuXAdah3N36Juw+a2bckvPkN5ncsDkFNdW
3Z3H8yej0l6fj/5D3f/QyFdTTJXXHFsj5l6tRk/YYbbkjWlyxZUrU2Y8aM5Lz33HNPsl6mgYGBZH3
JXXHFFsj516tRk/YYbbkjWlyxZUrU2Y8aM5Lz33HNPsl6mI0eOJOsbN25M1M25M1js7O6vWTp48mZz
3jTfeSNZffvnlZL0V5Q67u39zR//7WG32baasaHR1N1u+///5k/dSpU7ud91m7PO3QUvWhU3dfjXK
QiCsANBEHYgCMIOBEHYgSAIOxAEYQeCIOxAEIQdCIKwA0EQdiBIAg7EAT3s7eASy65Jfm9vbJ953333K+5JFmvNbtJ7c97Ob2y4VpjLw9j

```

VNmdszMDo6Z1mZmL5rZu9nzzCKbBVC8iezG/1rSrV+Z9qCkXe5+paRd2XsALaxm2N19r6Sv/h
50maRN2etNku4quC8ABcv72/h2dz83WNaHktqrfdDMuiV151wOgILUfSOMu3vqxJu790nqkzh
BB5Qp76W3o2Y2S5Ky52PFtQSgEfKGfZuk5dnr5ZK2FtMOgEapuRtvZlskfV/SZWZ2RNIvJK2T
9DszWyHpA0k/bGSTk92JEyfqmv+TTz7JPe8DDzyQrD/zzDPJeq0x1tE6aobd3buqlG4uuBcAD
cTPZYEgCDsQBGEHgiDsQBCEHQiCW1wngWnTplWtvfDCC8l5b7zxxxmT9tttuS9Z37tyZrKP5GL
IZCI6wA0EQdiAIwg4EQdiBIAg7EARhB4LgOvskN3/+/GR9//79yfrw8HCyvmfPnmS9v7+/au2
JJ55IztvMf5uTCdfZgeAIOxAEYQeCIOxAEIQdCIKwA0EQdiAIrrMH19nZmaw//9nzz/fTTyfr06
L3vNmjXJ+ubNm5P1oaGhZD0qrrMDWRF2IAjCDgRB2IEgCDsQBGEHgiDsQBBBcZ0fSNddck6xv2
LAhWb/55vyD/fb29ibra9euTdYHBwdzL/tClvs6u5k9ZWbhzOzgmGkPmdmR3IHrcX2SyA4k
1kN/7Xkm4dZ/q/untH9vh9sW0BKFrNsLv7XknHm9ALgAaq5wRdj5m9me3mz6z2ITPrNrN+M6v
+x8gANFzesP9S0nxJHZKGJK2v9kF373P3Re6+KOeyABQgV9jd/ai7n3X3UUm/krS42LYAFC1X
2M1s1pi3nZIOVvssgNS7psKSfunvNm4u5zj75zJgxI1m/8
847q9Zq3StvNu7l4i/t3r07WV+6dGmyPllVu85+0QRm7Bpn8pN1dwSgqfi5LBAEYQeCIOxAEI
QdCIKwA0FwiytK88UXTyTrF12Uvlg0MjKSrN9yyy1Vay+//HLy3gsZ97MDwRF2IAjCDgRB2IEgCDsQBGEHgiDsQBCEHQiCW1wngWnTplWtvfDCC8l5b7zxxmT9tttuS9Z37tyZrKP5GL
IOxAEIQdCIKwA0EQdiAIwg4EQdiBILjOPsktWLAgWe/p6UnW77777mT98ssvTJurs2bPJ+
tBQ+q+Xj46OFtnOBY8tOxAEYQeCIOxAEIQdCIKwA0EQdiAIwg4EwXX2C0C96eBlc8+cWOavH9M5Gf
r63Llz87RUiP7+/mR97dq1Wb/qmJydWzWnf+b+f8adXstqVvQfQomqG3d2H3H1/9vqkpHckzZa0TNKm
7GObJN3VqCYB1O+8jtnNbnb/O+8jtnNbK6k70naJ6nd3c/9OPlDSe1V5lkpaWV+LQJq15HbfNpsWrXz
U+MrXlldMhxB2109z53X+Tui+rqFEBdJhR2M/uGKkH/jbs/l00+amazsvosScca0yKAItTcjT
czk/SkpHfcfcOY0jZJJyWty563NqTDSaC9fdwjnC8tXLgwWX/88ceT9auuuuq8eyrKn37T2Ne6xkvV
HH320am3r1vQ/GW5RLdZEjtn/XtKPJb1lZgeyaWtUCfnvzGyFpA8k/bAxLQIoQs2wu/urkqr9+uNbxbYDoFH4BR0QBGEHgiDsQBCEHQiCW1wngWnTplWtvfDCC8l5b7zxxmT9tttuS9Z37tyZrKP5GL
cHdJNxfbDoBG4eeyQBCEHQiCsANBEHYgCMIOBMEtrhPU1tZWtdbb25uct6Oj67erp6K8
MorryTr69evT9Z37NiRrH/22Wfn3RMagy07EARhB4Ig7EAQhB0IgrADQRB2IAjCDgQR5jr79d
dfn6yvXr06WV+8eHHV2uzzs3P1VJRPPP/4/fTp07l7uamhN9u4EQdiBIAg7EAQhB0IgrADQYS5zt7Z2
7EARhB4Ig7EAQhB0IgrADQYS5zt7Z2VlXvR7vvfdesr558+Zkv729v+ulrt8+3d+n/+x
grADQRB2IAjCDgRB2IEgCDsQBGEHgjB3T3/AbYnZrQaSQbVkLaJbmkPnf/NzN7SSe+cx
nelFwagbu4+7qjLEwn7LEmz3H2/mQZ9Jqk7e+kfHZhyQNQZV+Pmt
k7ksr90ywAztt5HbOb2VxJ35OOL5vUY2ZvmtlZ3dZtZvZvZv11dQqgLjV347/8oNk3Jb0
saa27P2dm7ZI+uU4/p9V2dW/v8Z3sBsPNFjuY3ZJMrNvSvSNouaYe7bxinPsbfp3zHag
waqFveZuvJmZpCclvTM26SN0kAjTORs/FLJP2XpLckTjqtkSqTkuqjy/
U7mpb6LLTvQYHXtxheFsAONl3s3HsDkQNiBIAg7EARhB4IgrADgRB2IAjCDgRB2I
EgCDsQBGEHgiDsQBCEHQii2UM2fyTpL8V+L8ubBNp+/Qf+6wcHZr+2tTm/Rcg
7otIaSGjV3lq1L4ne8mpWb+zGA0EQdiCIssPeV/PyU9Pv0Xg3/t0m3jbbvFDh/UTzb+pI9Ba6nRR
IexAEKWE3cxSV+ffW0WLqauJtvGSyAvaGWnGiKrNuZlgOqjKPeA4JMdqdT+OrtUf/RL5+fHVUR
m8PmdlgtuOm0Znfnt5+F2z2mNmAmbkv4Ooz5eldTHF17+mAPkJ5JeldS17g
NNbaQKMzssaZGl/4DDDDP7B0mnJG0+N7SWmf2Puvi77j3Kmu/+8RXp7SOc5jHeDeqs2zPh
PVOK6K3L48zzK2LIvlvSeu7v7mck/VbSbjVhVbhL/UbSP/z2BqLIJ7IqscmwwSgmGpHaDrBpjeGBWJgmrrH3mJjkH9kaRq3pm2dltAeYa
3H1/9vqkpHPDjJe67hJ9NUUZYZ8t6U9hJmtlrZtZdpqtLr6aB8zzNaHktrLbGYcN
YfxbqavDDPeMusuz/Dn9eIE3dctcfe/lXSbpJXZ7ZpmLPmqkw4JG19mc1kw4
w/K+ln7n5ibK3MdTdOX01Zb2WEfVDSnDHvuuwaVu+mkRZrG0dP2gFRxPv0ZIeZvZ1YM8
3P+ruZ919VNKvOK6y4YZf1bSb9z9uWxyy7wdvQt
pmUnTmRm0yT9QK03FPUkScxl7/QKsN4VxtmXXCWvu9KHP3f3pj8k3a7KGfn/lfR
fRQpa95kt7IHm+X3ukLars1v2fKuC2Vkj6a0m0L6r6T8l1bZEl
30NyUdyB63l73uEn01Zb3xc1kgCE7QAUHEGkhgg4EQdiBIAg7EARhB4Ig7EAQ/w8ie3GmjcG
k5QAAAABJRU5ErkJggg==\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": "XZyMxpPz1Uvy",
    "outputId": "fc139da6-2679-4f94-e2c7-644f9fb0605f"
  },
  "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": "code",
  "source": [
    "X_train = X_train.reshape(60000, 28, 28, 1)\n",
    "X_test = X_test.reshape(10000, 28, 28, 1)"
  ],
  "metadata": {
    "id": "_CbvAuhz1Ubf"
  },
  "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))\n"
    ],
    "metadata": {
      "colab": {
        "base_uri": "https://localhost:8080/"
      },
      "id": "xrdaDDgP1UP_",
      "outputId": "f739c37d-f0cd-47c2-883c-ed6052ec2f05"
    },
    "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"
        ]
      }
    ]
  },
  {
    "cell_type": "code",
    "source": [
      "y_train = to_categorical(y_train)\n",
      "y_test = to_categorical(y_test)"
    ],
    "metadata": {
      "id": "QXusnXLx1UBE"
    },
    "execution_count": 7,
    "outputs": []
  },
  {
    "cell_type": "code",
    "source": [
      "model = Sequential()\n"
    ],
    "metadata": {
      "id": "_RWlfNVd1Txd"
    },
    "execution_count": 8,
    "outputs": []
  },
  {
    "cell_type": "code",
    "source": [
      "layer_1 = Conv2D(64, kernel_size=3, activation='relu',
input_shape=(28, 28, \n",
      "1))\n",
      "layer_2 = MaxPooling2D(pool_size=2)\n",
      "layer_3 = Conv2D(32, kernel_size=3, activation='relu')\n",
```

```
        "layer_4 = MaxPooling2D(pool_size=2)\n",
        "layer_5 = Dropout(0.5)\n",
        "layer_6 = Flatten()\n",
        "layer_7 = Dense(128, activation=\"relu\")\n",
        "layer_8 = Dropout(0.5)\n",
        "layer_9 = Dense(10, activation='softmax')"
      ],
      "metadata": {
        "id": "oJ5w_x9j1Tef"
      },
      "execution_count": 9,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "model.add(layer_1)\n",
        "model.add(layer_2)\n",
        "model.add(layer_3)\n",
        "model.add(layer_4)"
      ],
      "metadata": {
        "id": "znk9x6232R8c"
      },
      "execution_count": 10,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "model.compile(optimizer='adam',
loss='categorical_crossentropy',\n",
        "metrics=['accuracy'])\n"
      ],
      "metadata": {
        "id": "oiMAuNz2288l"
      },
      "execution_count": 11,
      "outputs": []
    }
  ]
}
```