

```

{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "provenance": [],
      "collapsed_sections": []
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    },
    "language_info": {
      "name": "python"
    }
  },
  "cells": [
    {
      "cell_type": "code",
      "source": [
        "import numpy as np\n",
        "import pandas as pd\n",
        "import matplotlib.pyplot as plt\n",
        "from keras.utils import np_utils\n",
        "from tensorflow.keras.datasets import mnist\n",
        "from tensorflow.keras.models import Sequential\n",
        "from tensorflow.keras.layers import Conv2D, Dense, Flatten\n",
        "from tensorflow.keras.optimizers import Adam\n",
        "from tensorflow.keras.models import load_model\n",
        "from PIL import Image, ImageOps\n",
        "import numpy"
      ],
      "metadata": {
        "id": "CCxED1HeMzVd"
      },
      "execution_count": 21,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "(X_train, y_train), (X_test, y_test) = mnist.load_data()"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "zLIGzPMHNPJAP",
        "outputId": "ecbd4710-5ef7-4f6b-8d08-d91d2f17f690"
      },
      "execution_count": 6,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",

```

```

"text": [
  "Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz\n",
  "11490434/11490434 [=====] - 1s 0us/step\n"
]
},
{
  "cell_type": "code",
  "source": [
    "print(X_train.shape)\n",
    "print(X_test.shape)"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "Y_tnEG7dNNNoZ",
    "outputId": "dc2e8101-fa06-4706-8325-5ea2b3659341"
  },
  "execution_count": 7,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "(60000, 28, 28)\n",
        "(10000, 28, 28)\n"
      ]
    }
  ]
},
{
  "cell_type": "code",
  "source": [
    "X_train[0]"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "779xmQUiNZeC",
    "outputId": "430c49c5-1110-43e9-e49b-b128b658d887"
  },
  "execution_count": 8,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "array([[ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,\n",
            "         0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,\n",
            "         0,  0],\n",
            "[ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,\n",
            "         0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,\n"

```

```

"    0, 0],\n",
"    [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
"    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
"    0, 0],\n",
"    [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
"    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
"    0, 0],\n",
"    [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
"    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
"    0, 0],\n",
"    [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3,\n",
"    18, 18, 18, 126, 136, 175, 26, 166, 255, 247, 127, 0, 0,\n",
"    0, 0],\n",
"    [ 0, 0, 0, 0, 0, 0, 0, 0, 30, 36, 94, 154, 170,\n",
"    253, 253, 253, 253, 253, 225, 172, 253, 242, 195, 64, 0, 0,\n",
"    0, 0],\n",
"    [ 0, 0, 0, 0, 0, 0, 0, 0, 49, 238, 253, 253, 253, 253,\n",
"    253, 253, 253, 253, 251, 93, 82, 82, 56, 39, 0, 0, 0,\n",
"    0, 0],\n",
"    [ 0, 0, 0, 0, 0, 0, 0, 18, 219, 253, 253, 253, 253,\n",
"    253, 198, 182, 247, 241, 0, 0, 0, 0, 0, 0, 0, 0,\n",
"    0, 0],\n",
"    [ 0, 0, 0, 0, 0, 0, 0, 0, 80, 156, 107, 253, 253,\n",
"    205, 11, 0, 43, 154, 0, 0, 0, 0, 0, 0, 0, 0,\n",
"    0, 0],\n",
"    [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 14, 1, 154, 253,\n",
"    90, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
"    0, 0],\n",
"    [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 139, 253,\n",
"    190, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
"    0, 0],\n",
"    [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 11, 190,\n",
"    253, 70, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
"    0, 0],\n",
"    [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 35,\n",
"    241, 225, 160, 108, 1, 0, 0, 0, 0, 0, 0, 0, 0,\n",
"    0, 0],\n",
"    [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
"    81, 240, 253, 253, 119, 25, 0, 0, 0, 0, 0, 0, 0,\n",
"    0, 0],\n",
"    [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
"    0, 45, 186, 253, 253, 150, 27, 0, 0, 0, 0, 0, 0,\n",
"    0, 0],\n",
"    [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
"    0, 0, 16, 93, 252, 253, 187, 0, 0, 0, 0, 0, 0,\n",
"    0, 0],\n",
"    [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
"    0, 0, 0, 0, 249, 253, 249, 64, 0, 0, 0, 0, 0,\n",
"    0, 0],\n",
"    [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
"    0, 46, 130, 183, 253, 253, 207, 2, 0, 0, 0, 0, 0,\n",
"    0, 0],\n",
"    [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 39,\n",
"    148, 229, 253, 253, 253, 250, 182, 0, 0, 0, 0, 0,\n",
"    0, 0],\n",

```

```

"    [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 24, 114, 221,\n",
"    253, 253, 253, 253, 201, 78, 0, 0, 0, 0, 0, 0, 0, 0,\n",
"    0, 0],\n",
"    [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 23, 66, 213, 253, 253,\n",
"    253, 253, 198, 81, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
"    0, 0],\n",
"    [ 0, 0, 0, 0, 0, 0, 18, 171, 219, 253, 253, 253, 253,\n",
"    195, 80, 9, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
"    0, 0],\n",
"    [ 0, 0, 0, 0, 55, 172, 226, 253, 253, 253, 253, 244, 133,\n",
"    11, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
"    0, 0],\n",
"    [ 0, 0, 0, 0, 136, 253, 253, 253, 212, 135, 132, 16, 0,\n",
"    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
"    0, 0],\n",
"    [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
"    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
"    0, 0],\n",
"    [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
"    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
"    0, 0],\n",
"    [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
"    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",
"    0, 0]], dtype=uint8)"
]
},
"metadata": {},
"execution_count": 8
}
]
},
{
"cell_type": "code",
"source": [
"y_train[0]"
],
"metadata": {
"colab": {
"base_uri": "https://localhost:8080/"
},
"id": "U8sjKbDoNbhm",
"outputId": "3b535bee-928b-4da4-d562-65019b2e4fa9"
},
"execution_count": 9,
"outputs": [
{
"output_type": "execute_result",
"data": {
"text/plain": [
"5"
]
},
},
"metadata": {},
"execution_count": 9
}
}

```

```
]
},
{
  "cell_type": "code",
  "source": [
    "plt.imshow(X_train[1])"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 282
    },
    "id": "X3f_OaRgNg3H",
    "outputId": "0284fc04-a668-4d23-cfc3-c77f9ec815bb"
  },
  "execution_count": 11,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "<matplotlib.image.AxesImage at 0x7f681b8d7f90>"
        ]
      },
      "metadata": {},
      "execution_count": 11
    },
    {
      "output_type": "display_data",
      "data": {
        "text/plain": [
          "<Figure size 432x288 with 1 Axes>"
        ],
        "image/png": "iVBORw0KGgoAAAANSUhEUgAAAPsAAAD4CAYAAAAAq5pAIAAAABHNCSVQICA  
gIfAhkIAAAAAIwSFlzAAALEgAACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAAbWF0cGxvdGxpYiB2ZXJzaW9u  
uMy4yLjIsIGh0dHA6Ly9tYXRwbG90bGliLm9yZy+WH4yJAAAOx0IEQVR4nO3df5DU9X3H8deb6wml4EAMh  
BBSonKhxDQQLxgbE0ycOGBnis40JkzHEGLnMpNoMdo2ju1MnHSmQzMmNmKwKYIEzA+czKiR6VAjXplaE  
0M4kAiCBkOggidUsAV/4R337h/3NXPqfT+77H53v3v3fj5mbnb3+97vft+z+uK73+9nv/sxdxeA0W9M2Q0AaA7C  
DgRB2IEgCDsQBGEHgiDZm7sNBvr4zShmZsEQnlFL+pVP2HD1eoKu5ktkvQNSW2SvufuK1PPH6cJusAuqWe  
TABi2e3dureaP8WbWJmmVpMWS5kpaamZza309AI1VzzH7AkIPufted39V0l2SlhTTFoCi1RP2GZKeHvL4QLbs  
dcysy8x6zKynTyfq2ByAejT8bLy7r3b3TnfvbNfYRm8OQI56wn5Q0swhj9+RLQPQguoJ+xZJs83sXWZ2mqRPSV  
pfTFsAilbz0Ju795vZNZJ+psGhtzXu/nhhnQEoVF3j7O6+QdKGgnoB0EB8XRYIgrADQRB2IAjCDgRB2IEgCDsQ  
BGEHgiDsQBCEHQiCsANBEHYgCMIOBEHYgSAIOxAEYQeCIOxAEIQdCIKwA0EQdiAIwg4EQdiBIJo6ZTN  
Gn/6PnZ+s934+f8qvX1+4Nrnu+x5Zlqy/fdVpyXrbpm3JejTs2YEGCDsQBGEHgiDsQBCEHQiCsANBEHYgCMbZ  
kTSwcH6y/s0130rWz23P/19soMK2H73w+8n6k50nk/W/mfXBCluIpa6wm9k+ScclnZTU7+6dRTQFoHhF7Nk/6u7  
PFfA6ABqIY3YgiHrD7pIeMLOtZtY13BPMrMvMesysp0/535MG0Fj1foy/yN0PmtlUSRvN7Al3f2joE9x9taTVkjT  
Jpnid2wNQo7r27O5+MLs9LOleSQuKaApA8WoOu5lNML0Jr92XdKmknUU1BqBY9XyMnybpXjN77XV+7O73  
F9IVmqbv0vRo6d/e9oNkvaM9fU35QGI0fW9fX3Ld/xsYm6zPT5d1YvEHcmvjN+1IrjvwyivpFx+Bag67u++V9L4  
CewHQQAY9AUEQdiAIwg4EQdiBIAG7EASXuI4CbZMm5dZe/Mic5LpfvPXHyfpHx79QYeu17y/ueP5PkvXu2y5  
M1n9+8zeT9Y3f+05ube4Pr0muc/aXHknWRyL27EAQhB0IgrADQRB2IAjCDgRB2IEgCDsQBOPso8CBO2fk1rZ  
8YFUTOzk1X5m6JVM//4z0OPzyfZcm62tnPZhbzmT3SHLd0Yg9OxAEYQeCIOxAEIQdCIKwA0EQdiAIwg4Ew  
Tj7CND/sfOT9XXz8qdNHqP0Tz1Xsnz/Jcl6z4N/lKzvuDq/t00vj0uuO7Xn5WT9qefT1+q3/+Om3NoYS646KrFnB4  
Ig7EAQhB0IgrADQRB2IAjCDgRB2IEgzN2btrFJNsUvsPS4bUQDC+cn6/+89rZk/dz22r8u8WdPXJGst/35i8n60T9  
9d7J+5Lz8Ae2OVU8n1+1/+kCyXsm/HdyaW+s9mR7D/+yyv0rW2zZtq6mnRtvs3TrmR4d90yvu2c1sjZkdNrOdQ5  
ZNMbONZrYnu51cZMMailfNx/g7JC16w7IbJXW7+2xJ3dljAC2sYtjd/SFJR9+weImktdn9tZiUL7gvAAWr9WBv
```

mrv3ZveflTQt74lm1iWpS5LG6fQaNwegXnWfjffBM3y5Z/ncfbW7d7p7Z7vG1rs5ADWqNeyHzGy6JGW3h4trCU  
Aj1Br29ZKWZfeXSbqvmHYANErFY3YzWyfpYklnmdkBSV+WtFLST8zsakn7JV3ZyCZHOjv/Pcn6c9enx3w72tP  
XpG89kV/7jxfmJtc9ctfMZP0tz6fnKT/zh79M1xO1/uSajTWtLX1IeeS6l5L1qfmXyresimF396U5Jb4dA4wgfF0WCI  
KwA0EQdiAIwg4EQdiBIPgp6QKMOT39NeD+rx5L1n85555k/Xf9rybr1990Q25t8n/9d3LdqRPS34c6mayOXgum7  
0/W9zWnjUKxZweCIOxAEIQdCIKwA0EQdiAIwg4EQdiBIBhnL8DLC9OXsP5sTvqnoCv5yxVfTNYn/jT/MtMy  
LyNFa2HPDgRB2IEgCDsQBGEHgiDsQBCEHQiCsANBMM5egD/+h+3J+pgK/6Yu35/+od7xP/3VKfcEqd3acmt9  
FWYqb7PmTWXeLOzZgSAIOxAEYQeCIOxAEIQdCIKwA0EQdiAIxtmr9L9XXZhb+/tptyTXHVCfKZcfSE+r/E  
79IlnH8Po8/1fvBzSQXPf+3en/JrO1raaeylRxx25ma8zssJntHLLsZjM7aGbb57/LGtsmgHpV8zH+DkmLhl+q7vPy/4  
2FNsWgKJVDLu7PyTpaBN6AdBA9Zygu8bMHss+5k/Oe5KZdZlZj5n19OIEHZsDUI9aw/5tSedImiepV9LX8p7o7  
qvvdvdPdO9s1tsbNAahXTWF390PuffLdByR9V9KCYtsCULSawm5m04c8vELSzrznaMgNFcfZzWydpIslNwVmb  
yR9WdLFZjZPkmtwqurPNbDHltA/Pr925pj0OPojr6QPX86+85n0tpPV0avSvPdP3HJehVfYmlv5i72Lk2vOWfG7Z  
H0kzltfMezuvnSYxabc3oBcADcTXZYEgCDsQBGEHgiDsQBCEHQiCS1yb4MjJM5L1/r37mtNli6k0tPbkycvm608  
s+Vay/u8vnZlbe2bVucl1Jz6fPw32SMWeHQiCsANBEHYgCMIOBEHYgSAIOxAEYQeCYJy9Cf76559I1jsSl2KO  
dAML5+fWDI//cnLd3Z3pcfRLdnwyWZ+waG9ubaJG3zh6JezZgSAIOxAEYQeCIOxAEIQdCIKwA0EQdiAIxtmr  
ZfmlMRX+zfzGREuS9VXqqKWjlrD/K/ITWUvS3Z/+em6toz39E9zv/9WyZP3tV+xK1vF67NmBIAg7EARhB4Ig7  
EAQhB0IgrADQRB2IAjG2avl+aUBDSRXXTj+SLJ+3R3nJ+vnfD/9+u3PHs+thVr41uS6Uz55IFm/9p3dyfri09PX4  
q9/cVpu7dM7FiXXPetfJyTrODUV9+xmNtPMNpnZLjN73MxWZMunmNlGM9uT3U5uflsAalXNx/h+STe4+1xJ  
H5T0BTObK+1GSd3uPltSd/YYQIuqGHZ373X3bdn945J2S5ohaYmktDnT1kq6vFFNAqjfKR2zm9ksSfMlbZY0zd1  
7s9KzkoY9ODOzLklkdjRO6bm9ADRO1WfjzewMSXdLus7djw2tubsr5xSWu692905372zX2LqaBVC7qsJuZu0aD  
PqP3P2ebPEhM5ue1adLOtyYFgEUoeLHeDMzSbdL2u3uQ69XXC9pmaSV2e19DelwFBhn6bd598e/k6w//OFxyfq  
eE2/LrS0/c19y3XqteObDyfr9v5iXW5u9It7POZepmmP2D0m6StlOM9ueLbtJgyH/iZldLWm/pCsb0yKAlIQMu7s/r  
Pyfbrik2HYANApflwWCIOxAEIQdCIKwA0EQdiAIG/zyW3NMsil+gY3ME/htHefk1jrW7U+u+09ve6SubVf6qep  
Kl9imPHoi/dpL/7MrWe9YPnqnm6JNnu3jvnRYUfP2LMDQRB2IAjCDgRB2IEgCDsQBGEHgiDsQBD8lHSVTv  
7mt7m1PZ+Y1Vx37rXXJuu7rvyXWlqqypwNn0/W333bS8l6x6OMo48W7NmBIAg7EARhB4Ig7EAQhB0IgrADQ  
RB2IAiuZwdGEa5nB0DYgSgIOxAEYQeCIOxAEIQdCIKwA0FUDLuZzTSzTWa2y8weN7MV2fKbzeygmW3P/i  
5rflsAalXNj1f0S7rB3beZ2URJW81sY1a71d1vaVx7A1pSzfzsvZJ6s/vHzWy3pBmNbgxAsU7pmN3MZkmaL2lztu  
gaM3vMzNaY2eScdbrMrMfMevp0oq5mAdSu6rCb2RmS7pZ0nbsfk/RtSedImqfBPf/XhlvP3Ve7e6e7d7ZrbAEtA6  
hFVWE3s3YNBv1H7n6PJLn7IXc/6e4Dkr4raUHj2gRQr2rOxpuk2yXtdvevD1k+fcjTrpC0s/j2ABSImrPxH5J0laQd  
ZrY9W3aTpKVmNk+SS9on6XMN6RBAIao5G/+wpOGuj91QfDsAGoVv0AFBEHYgCMIOBEHYgSAIOxAEYQ  
eCIOxAEIQdCIKwA0EQdiAIwg4EQdiBIAg7EARhB4Jo6pTNZvY/kvYPWXSwpOealsCpadXeWrUvid5qVWRv  
f+jubx2u0NSwv2njZj3u3llaAwmt2lur9iXRW62a1Rsf44EgCDsQRNlhX13y9lNatbdW7Uuit1o1pbdSj9kBNE/Ze3Y  
ATULYgSBKCbuzLTkZJ83sKTO7sYwe8pjZPjPbkU1D3VNyL2vM7LCZ7RyybIqZbTSzPdntsHPsldRbS0zjnZh  
mvNT3ruzpz5t+zG5mbZJ+I+njkg5I2iJpqbvnamojOcxsn6ROdy/9Cxhm9hFJL0i6093Py5Z9VdJRd1+Z/UM52d2/1C  
K93SzphbKn8c5mK5o+dJpxSZdL+oxKfO8SfV2pJrxvZezZF0h6yt33uvurku6StKSEPlqez8k6egbFi+RtDa7v1aD/  
7M0XU5vLcHde919W3b/uKTXphkv9b1L9NUUZyR9hqSnhzw+oNaa790IPWBmW82sq+xmhjHN3Xuz+89KmlZ  
mM8OoOI13M71hmvGWee9qmf68Xpyge7OL3P39khZL+kL2cbUI+eAxWCuNnVY1jXezDDPN+O+V+d7VOv1  
5vcoI+0FJM4c8fke2rCW4+8Hs9rCke9V6U1Efem0G3ez2cMn9/F4rTeM93DTjaoH3rszpz8si+xZJs83sXWZ2mqRP  
SVpfQh9vYmYTshMnMrMjki5V601FvV7Ssuz+Mkn3ldjL67TKNN5504yr5Peu9OnP3b3pf5Iu0+AZ+d9K+rsyesj  
p62xJv87+Hi+7N0nrNPixrk+D5zaulvQWSd2S9kh6UNKUFurtB5J2SHpMg8GaXlJvF2nwI/pjkrZnf5eV/d4l+mrK+  
8bXZYEgOEEHBEHYgSAIOxAEYQeCIOxAEIQdCIKwA0H8Px6GUTt0IpTWAAAAAEIFTkSuQmCC\n"

```
},
"metadata": {
  "needs_background": "light"
}
```

```
]
},
{
  "cell_type": "code",
  "source": [
    "X_train = X_train.reshape(60000, 28, 28, 1).astype('float32')\n",
    "X_test = X_test.reshape(10000, 28, 28, 1).astype('float32')\n",
  ],
  "metadata": {
    "id": "L-p01C12NnoR"
  }
}
```

```

    },
    "execution_count": 12,
    "outputs": []
  },
  {
    "cell_type": "code",
    "source": [
      "number_of_classes = 10\n",
      "Y_train = np_utils.to_categorical(y_train, number_of_classes)\n",
      "Y_test = np_utils.to_categorical(y_test, number_of_classes)"
    ],
    "metadata": {
      "id": "kp9OObLHNwsF"
    },
    "execution_count": 13,
    "outputs": []
  },
  {
    "cell_type": "code",
    "source": [
      "Y_train[0]"
    ],
    "metadata": {
      "colab": {
        "base_uri": "https://localhost:8080/"
      },
      "id": "uBDq03_ZN5_H",
      "outputId": "b4a5def4-7bce-4086-9458-61f45ec85373"
    },
    "execution_count": 14,
    "outputs": [
      {
        "output_type": "execute_result",
        "data": {
          "text/plain": [
            "array([0., 0., 0., 0., 0., 1., 0., 0., 0., 0.], dtype=float32)"
          ]
        },
        "metadata": {},
        "execution_count": 14
      }
    ]
  },
  {
    "cell_type": "code",
    "source": [
      "model = Sequential()\n",
      "model.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1), activation='relu'))\n",
      "model.add(Conv2D(32, (3, 3), activation='relu'))\n",
      "model.add(Flatten())\n",
      "model.add(Dense(number_of_classes, activation='softmax'))"
    ],
    "metadata": {
      "id": "gXqnKch1N_U5"
    },
  },

```

```

    "execution_count": 15,
    "outputs": []
  },
  {
    "cell_type": "code",
    "source": [
      "model.compile(loss='categorical_crossentropy', optimizer='Adam', metrics=['accuracy'])"
    ],
    "metadata": {
      "id": "nbH5ITuROIU5"
    },
    "execution_count": 16,
    "outputs": []
  },
  {
    "cell_type": "code",
    "source": [
      "model.fit(X_train, Y_train, batch_size=32, epochs=5, validation_data=(X_test, Y_test))"
    ],
    "metadata": {
      "colab": {
        "base_uri": "https://localhost:8080/"
      },
      "id": "akSsr8AoONGE",
      "outputId": "72599b43-452d-411e-e812-fcb238776531"
    },
    "execution_count": 17,
    "outputs": [
      {
        "output_type": "stream",
        "name": "stdout",
        "text": [
          "Epoch 1/5\n",
          "1875/1875 [=====] - 194s 103ms/step - loss: 0.2567 - accuracy: 0.9506\n",
          "- val_loss: 0.0980 - val_accuracy: 0.9693\n",
          "Epoch 2/5\n",
          "1875/1875 [=====] - 196s 105ms/step - loss: 0.0695 - accuracy: 0.9791\n",
          "- val_loss: 0.0983 - val_accuracy: 0.9735\n",
          "Epoch 3/5\n",
          "1875/1875 [=====] - 196s 105ms/step - loss: 0.0494 - accuracy: 0.9842\n",
          "- val_loss: 0.0906 - val_accuracy: 0.9755\n",
          "Epoch 4/5\n",
          "1875/1875 [=====] - 192s 102ms/step - loss: 0.0375 - accuracy: 0.9882\n",
          "- val_loss: 0.0913 - val_accuracy: 0.9787\n",
          "Epoch 5/5\n",
          "1875/1875 [=====] - 196s 104ms/step - loss: 0.0306 - accuracy: 0.9903\n",
          "- val_loss: 0.1032 - val_accuracy: 0.9743\n"
        ]
      },
      {
        "output_type": "execute_result",
        "data": {
          "text/plain": [
            "<keras.callbacks.History at 0x7f6817548650>"
          ]
        }
      }
    ]
  }
]

```



```

    },
    "metadata": {},
    "execution_count": 17
  }
]
},
{
  "cell_type": "code",
  "source": [
    "metrics = model.evaluate(X_test, Y_test, verbose=0)\n",
    "print(\"Metrics (Test Loss & Test Accuracy): \")\n",
    "print(metrics)"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "IwniW6MWSCt-",
    "outputId": "3acd23ba-b83c-4625-bd56-be9eabb5023e"
  },
  "execution_count": 18,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "Metrics (Test Loss & Test Accuracy): \n",
        "[0.10322817414999008, 0.9743000268936157]\n"
      ]
    }
  ]
},
{
  "cell_type": "code",
  "source": [
    "prediction = model.predict(X_test[:4])\n",
    "print(prediction)"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "GiYiBAYqSJr1",
    "outputId": "13963cd5-c4d2-4dea-d645-19d11808c5c4"
  },
  "execution_count": 29,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "1/1 [=====] - 0s 64ms/step\n",
        "[[4.77358049e-11 1.26020884e-14 2.23637656e-07 2.59297366e-07]\n",
        " 1.53105145e-18 1.41474479e-13 2.73819453e-19 9.99999523e-01]\n",
        " 5.75746352e-12 1.40723442e-08]\n"
      ]
    }
  ]
}

```

```

" [3.92702641e-05 3.63764530e-09 9.99928832e-01 1.10518204e-06\n",
" 3.28396650e-11 1.87219923e-13 3.02575540e-06 4.75269130e-12\n",
" 2.79003762e-05 1.17118581e-09]\n",
" [3.37602168e-11 9.99982953e-01 7.10459869e-09 3.63090309e-13\n",
" 1.67968246e-05 6.36366426e-09 4.59948364e-11 2.65287614e-09\n",
" 2.72516672e-07 1.53049936e-12]\n",
" [9.99999762e-01 1.02759820e-17 6.89465485e-10 4.13503087e-14\n",
" 3.53135576e-12 2.56500203e-11 6.89072754e-09 4.50628203e-14\n",
" 8.74276596e-10 1.82247064e-07]]\n"
]
}
]
},
{
"cell_type": "code",
"source": [
"print(numpy.argmax(prediction, axis=1))\n",
"print(Y_test[:4])"
],
"metadata": {
"colab": {
"base_uri": "https://localhost:8080/"
},
"id": "ZnSCNyNXSjvG",
"outputId": "16c52b18-1d32-41df-baf3-739d1f476c53"
},
"execution_count": 22,
"outputs": [
{
"output_type": "stream",
"name": "stdout",
"text": [
"[7 2 1 0]\n",
"[[0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]\n",
" [0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]\n",
" [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]\n",
" [1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]\n"
]
}
]
},
{
"cell_type": "code",
"source": [
"model.save(\"model.h5\")"
],
"metadata": {
"id": "vFBZ8us4SoBU"
},
"execution_count": 23,
"outputs": []
},
{
"cell_type": "code",
"source": [

```

```
    "model=load_model(\"model.h5\")"  
  ],  
  "metadata": {  
    "id": "IDH-txZvSs-3"  
  },  
  "execution_count": 24,  
  "outputs": []  
}  
]  
}
```