```
{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "provenance": [],
      "toc_visible": true
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    },
    "language_info": {
      "name": "python"
    }
  },
  "cells": [
    {
      "cell_type": "markdown",
      "source": [
        "Sprint 2"
      ],
      "metadata": {
        "id": "bAPYJfv1ge3U"
      }
    },
    {
      "cell_type": "markdown",
      "source": [
        "Team id: PNT2022TMID25747"
      ],
      "metadata": {
        "id": "25vwDtOkgjUT"
      }
    },
    {
      "cell_type": "markdown",
      "source": [
        "Libraries"
      ],
      "metadata": {
        "id": "3014sV0Pg7gW"
      }
    },
    {
      "cell_type": "code",
      "execution_count": 1,
      "metadata": {
        "id": "GUAqTqHWTJve"
      },
      "outputs": [],
      "source": [
        "import numpy as np\n",
        "import tensorflow \n",
        "from tensorflow.keras.datasets import mnist \n",
        "from tensorflow.keras.models import Sequential \n",
        "from tensorflow.keras import layers \n",
        "from tensorflow.keras.layers import Dense, Flatten \n",
```

```
    "from tensorflow.keras.layers import Conv2D \n",
    "from keras. utils import np_utils \n",
    "import matplotlib.pyplot as plt   "
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "Load the Data"
  ],
  "metadata": {
    "id": "0zIRTLkug-CT"
  }
},
{
  "cell_type": "code",
  "source": [
    "(x_train, y_train), (x_test, y_test)=mnist.load_data ()\n",
    "x_train=x_train.reshape (60000, 28, 28, 1).astype('float32')\n",
    "x_test=x_test.reshape (10000, 28, 28, 1).astype ('float32')\n",
    "\n",
    "x_train.shape"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "vXYh1tD4c1Eq",
    "outputId": "6565e34a-8a8e-42e0-97c1-d6a8c2537c85"
  },
  "execution_count": 19,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "(60000, 28, 28, 1)"
        ]
      },
      "metadata": {},
      "execution_count": 19
    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "One Hot encoding"
  ],
  "metadata": {
    "id": "t15cuClDhT7f"
  }
},
{
  "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": "g80zefzQhRC_"
      },
      "execution_count": 20,
      "outputs": []
    },
    {
      "cell_type": "markdown",
      "source": [
        "Add Layers"
      ],
      "metadata": {
        "id": "OsutO52qhctz"
      }
    },
    {
      "cell_type": "code",
      "source": [
        "model=Sequential ()\n",
        "\n",
        "model.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1),
activation='relu'))\n",
        "model.add(Conv2D(32, (3, 3), activation = 'relu'))\n",
        "\n",
        "model.add(Flatten())\n",
        "model.add(Dense(number_of_classes,activation = 'softmax'))"
      ],
      "metadata": {
        "id": "MwvV0HK4c6Oc"
      },
      "execution_count": 23,
      "outputs": []
    },
    {
      "cell_type": "markdown",
      "source": [
        "Compile model"
      ],
      "metadata": {
        "id": "BVSVSKuVhrHa"
      }
    },
    {
      "cell_type": "code",
      "source": [
        "model.compile(loss= 'categorical_crossentropy',
optimizer=\"Adam\", metrics=['accuracy'])"
      ],
      "metadata": {
        "id": "xdk6eEbTdIMJ"
      },
      "execution_count": 24,
      "outputs": []
    },
```

```
    {
      "cell_type": "code",
      "source": [
        "x_train = np.asarray(x_train)\n",
        "y_train = np.asarray(y_train)"
      ],
      "metadata": {
        "id": "trnT3hvrdLPQ"
      },
      "execution_count": 8,
      "outputs": []
    },
    {
      "cell_type": "markdown",
      "source": [
        "Model training"
      ],
      "metadata": {
        "id": "ValD_GsrhyQq"
      }
    },
    {
      "cell_type": "code",
      "source": [
        "model.fit(x_train, y_train, validation_data=(x_test, y_test),
epochs=5, batch_size=32)"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "2gWsqfxndORJ",
        "outputId": "77d7dce4-838f-490c-bd6a-d3fb4efb0b2d"
      },
      "execution_count": 25,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "Epoch 1/5\n",
            "1875/1875 [==============================] - 124s 66ms/step
- loss: 0.2126 - accuracy: 0.9534 - val_loss: 0.0857 - val_accuracy:
0.9727\n",
            "Epoch 2/5\n",
            "1875/1875 [==============================] - 122s 65ms/step
- loss: 0.0633 - accuracy: 0.9806 - val_loss: 0.0812 - val_accuracy:
0.9751\n",
            "Epoch 3/5\n",
            "1875/1875 [==============================] - 121s 65ms/step
- loss: 0.0451 - accuracy: 0.9863 - val_loss: 0.0776 - val_accuracy:
0.9780\n",
            "Epoch 4/5\n",
            "1875/1875 [==============================] - 122s 65ms/step
- loss: 0.0313 - accuracy: 0.9905 - val_loss: 0.1089 - val_accuracy:
0.9736\n",
            "Epoch 5/5\n",
```

```
            "1875/1875 [==============================] - 121s 65ms/step
- loss: 0.0277 - accuracy: 0.9917 - val_loss: 0.0893 - val_accuracy:
0.9779\n"
          ]
        },
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "<keras.callbacks.History at 0x7fd48aca7a10>"
            ]
          },
          "metadata": {},
          "execution_count": 25
        }
      ]
    },
    {
      "cell_type": "markdown",
      "source": [
        "Evaluating the model for Accuracy"
      ],
      "metadata": {
        "id": "p6gQCBuQh4jY"
      }
    },
    {
      "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": "4FfWxiILf6pJ",
        "outputId": "30f49618-7fb5-4095-bf9d-bba4df94fd70"
      },
      "execution_count": 26,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "Metrics [Test loss, Test Accuracy] : \n",
            "[0.0893181562423706, 0.9779000282287598]\n"
          ]
        }
      ]
    },
    {
      "cell_type": "markdown",
      "source": [
        "Prediction"
      ],
      "metadata": {
```

```
      "id": "7E7b6lcHiJBj"
    }
  },
  {
    "cell_type": "code",
    "source": [
      "prediction=model.predict(x_test[6000:6001])\n",
      "print(prediction)"
    ],
    "metadata": {
      "colab": {
        "base_uri": "https://localhost:8080/"
      },
      "id": "zMM6L4syf_lv",
      "outputId": "ea4d0d08-f87e-4e96-e7f7-57b713439c25"
    },
    "execution_count": 13,
    "outputs": [
      {
        "output_type": "stream",
        "name": "stdout",
        "text": [
          "1/1 [==============================] - 0s 15ms/step\n",
          "[[3.3381129e-13 1.0344202e-18 2.4497482e-11 1.6618515e-09
1.0295421e-06\n",
          "  2.3502807e-11 6.7067222e-20 1.2130131e-07 2.3948130e-07
9.9999857e-01]]\n"
        ]
      }
    ]
  },
  {
    "cell_type": "code",
    "source": [
      "import numpy as np\n",
      "print(np.argmax(prediction, axis=1))"
    ],
    "metadata": {
      "colab": {
        "base_uri": "https://localhost:8080/"
      },
      "id": "2K0liO_AgQBP",
      "outputId": "c08bdee9-2e68-4353-9ece-87d4f1ba9d56"
    },
    "execution_count": 16,
    "outputs": [
      {
        "output_type": "stream",
        "name": "stdout",
        "text": [
          "[9]\n"
        ]
      }
    ]
  },
  {
    "cell_type": "code",
    "source": [
```

```
        "np.argmax(y_test[6000:6001])"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "58Es--w_gTqH",
        "outputId": "02fc04d1-9226-49df-f7d3-0ced3da371fb"
      },
      "execution_count": 17,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "9"
            ]
          },
          "metadata": {},
          "execution_count": 17
        }
      ]
    },
    {
      "cell_type": "markdown",
      "source": [
        "Save the model"
      ],
      "metadata": {
        "id": "d4OryNZ9iOQT"
      }
    },
    {
      "cell_type": "code",
      "source": [
        "model.save('models/mnistDatasetmodel.h5')"
      ],
      "metadata": {
        "id": "9kmgtteYgZJQ"
      },
      "execution_count": 28,
      "outputs": []
    }
  ]
}
```