{

"cells": [

{

"cell\_type": "markdown",

"id": "345fe7e6",

"metadata": {},

"source": [

"# IBM cloud deployment"

]

},

{

"cell\_type": "markdown",

"id": "a7dfa821",

"metadata": {},

"source": [

"Team ID - PNT2022TMID27424"

]

},

{

"cell\_type": "markdown",

"id": "b1cbb86f",

"metadata": {},

"source": [

"# Importing the required libraries"

]

},

{

"cell\_type": "code",

"execution\_count": 5,

"id": "cc225ad3",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Requirement already satisfied: tensorflow in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (2.7.2)\n",

"Collecting tensorflow\n",

" Downloading tensorflow-2.10.0-cp39-cp39-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl (578.1 MB)\n",

"\u001b[K |████████████████████████████████| 578.1 MB 40 kB/s /s eta 0:00:01B 15.2 MB/s eta 0:00:34 |████▊ | 84.6 MB 15.2 MB/s eta 0:00:33 |█████ | 90.0 MB 15.2 MB/s eta 0:00:33 |█████████▉ | 177.6 MB 103.6 MB/s eta 0:00:04| 280.1 MB 104.4 MB/s eta 0:00:03/s eta 0:00:03\n",

"\u001b[?25hRequirement already satisfied: termcolor>=1.1.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.1.0)\n",

"Requirement already satisfied: flatbuffers>=2.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (2.0)\n",

"Requirement already satisfied: gast<=0.4.0,>=0.2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (0.4.0)\n",

"Requirement already satisfied: keras-preprocessing>=1.1.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.1.2)\n",

"Requirement already satisfied: numpy>=1.20 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.20.3)\n",

"Requirement already satisfied: grpcio<2.0,>=1.24.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.42.0)\n",

"Collecting absl-py>=1.0.0\n",

" Downloading absl\_py-1.3.0-py3-none-any.whl (124 kB)\n",

"\u001b[K |████████████████████████████████| 124 kB 83.1 MB/s eta 0:00:01\n",

"\u001b[?25hRequirement already satisfied: astunparse>=1.6.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.6.3)\n",

"Requirement already satisfied: six>=1.12.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.15.0)\n",

"Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (0.23.1)\n",

"Collecting keras<2.11,>=2.10.0\n",

" Downloading keras-2.10.0-py2.py3-none-any.whl (1.7 MB)\n",

"\u001b[K |████████████████████████████████| 1.7 MB 88.2 MB/s eta 0:00:01\n",

"\u001b[?25hCollecting libclang>=13.0.0\n",

" Downloading libclang-14.0.6-py2.py3-none-manylinux2010\_x86\_64.whl (14.1 MB)\n",

"\u001b[K |████████████████████████████████| 14.1 MB 78.8 MB/s eta 0:00:01\n",

"\u001b[?25hCollecting tensorboard<2.11,>=2.10\n",

" Downloading tensorboard-2.10.1-py3-none-any.whl (5.9 MB)\n",

"\u001b[K |████████████████████████████████| 5.9 MB 69.7 MB/s eta 0:00:01\n",

"\u001b[?25hRequirement already satisfied: opt-einsum>=2.3.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (3.3.0)\n",

"Requirement already satisfied: setuptools in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (58.0.4)\n",

"Requirement already satisfied: packaging in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (21.3)\n",

"Requirement already satisfied: typing-extensions>=3.6.6 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (4.1.1)\n",

"Requirement already satisfied: google-pasta>=0.1.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (0.2.0)\n",

"Requirement already satisfied: protobuf<3.20,>=3.9.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (3.19.1)\n",

"Requirement already satisfied: h5py>=2.9.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (3.2.1)\n",

"Collecting tensorflow-estimator<2.11,>=2.10.0\n",

" Downloading tensorflow\_estimator-2.10.0-py2.py3-none-any.whl (438 kB)\n",

"\u001b[K |████████████████████████████████| 438 kB 83.2 MB/s eta 0:00:01\n",

"\u001b[?25hRequirement already satisfied: wrapt>=1.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow) (1.12.1)\n",

"Requirement already satisfied: wheel<1.0,>=0.23.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from astunparse>=1.6.0->tensorflow) (0.37.0)\n",

"Requirement already satisfied: google-auth<3,>=1.6.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard<2.11,>=2.10->tensorflow) (1.23.0)\n",

"Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard<2.11,>=2.10->tensorflow) (0.4.4)\n",

"Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard<2.11,>=2.10->tensorflow) (0.6.1)\n",

"Requirement already satisfied: requests<3,>=2.21.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard<2.11,>=2.10->tensorflow) (2.26.0)\n",

"Requirement already satisfied: markdown>=2.6.8 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard<2.11,>=2.10->tensorflow) (3.3.3)\n",

"Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard<2.11,>=2.10->tensorflow) (1.6.0)\n",

"Requirement already satisfied: werkzeug>=1.0.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard<2.11,>=2.10->tensorflow) (2.0.2)\n",

"Requirement already satisfied: cachetools<5.0,>=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from google-auth<3,>=1.6.3->tensorboard<2.11,>=2.10->tensorflow) (4.2.2)\n",

"Requirement already satisfied: rsa<5,>=3.1.4 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from google-auth<3,>=1.6.3->tensorboard<2.11,>=2.10->tensorflow) (4.7.2)\n",

"Requirement already satisfied: pyasn1-modules>=0.2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from google-auth<3,>=1.6.3->tensorboard<2.11,>=2.10->tensorflow) (0.2.8)\n",

"Requirement already satisfied: requests-oauthlib>=0.7.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from google-auth-oauthlib<0.5,>=0.4.1->tensorboard<2.11,>=2.10->tensorflow) (1.3.0)\n",

"Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pyasn1-modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard<2.11,>=2.10->tensorflow) (0.4.8)\n",

"Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard<2.11,>=2.10->tensorflow) (2022.9.24)\n",

"Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard<2.11,>=2.10->tensorflow) (2.0.4)\n",

"Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard<2.11,>=2.10->tensorflow) (3.3)\n",

"Requirement already satisfied: urllib3<1.27,>=1.21.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests<3,>=2.21.0->tensorboard<2.11,>=2.10->tensorflow) (1.26.7)\n",

"Requirement already satisfied: oauthlib>=3.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tensorboard<2.11,>=2.10->tensorflow) (3.2.1)\n",

"Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from packaging->tensorflow) (3.0.4)\n",

"Installing collected packages: absl-py, tensorflow-estimator, tensorboard, libclang, keras, tensorflow\n",

" Attempting uninstall: absl-py\n",

" Found existing installation: absl-py 0.12.0\n",

" Uninstalling absl-py-0.12.0:\n",

" Successfully uninstalled absl-py-0.12.0\n",

" Attempting uninstall: tensorflow-estimator\n",

" Found existing installation: tensorflow-estimator 2.7.0\n",

" Uninstalling tensorflow-estimator-2.7.0:\n",

" Successfully uninstalled tensorflow-estimator-2.7.0\n",

" Attempting uninstall: tensorboard\n",

" Found existing installation: tensorboard 2.7.0\n",

" Uninstalling tensorboard-2.7.0:\n",

" Successfully uninstalled tensorboard-2.7.0\n",

" Attempting uninstall: keras\n",

" Found existing installation: Keras 2.2.4\n",

" Uninstalling Keras-2.2.4:\n",

" Successfully uninstalled Keras-2.2.4\n",

" Attempting uninstall: tensorflow\n",

" Found existing installation: tensorflow 2.7.2\n",

" Uninstalling tensorflow-2.7.2:\n",

" Successfully uninstalled tensorflow-2.7.2\n",

"\u001b[31mERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.\n",

"tensorflow-text 2.7.3 requires tensorflow<2.8,>=2.7.0, but you have tensorflow 2.10.0 which is incompatible.\n",

"tensorflow-metadata 1.5.0 requires absl-py<0.13,>=0.9, but you have absl-py 1.3.0 which is incompatible.\n",

"autoai-ts-libs 1.1.9 requires tensorflow<2.8,>=2.7.0; python\_version >= \"3.9\", but you have tensorflow 2.10.0 which is incompatible.\u001b[0m\n",

"Successfully installed absl-py-1.3.0 keras-2.10.0 libclang-14.0.6 tensorboard-2.10.1 tensorflow-2.10.0 tensorflow-estimator-2.10.0\n"

]

}

],

"source": [

"!pip install tensorflow --upgrade"

]

},

{

"cell\_type": "code",

"execution\_count": 31,

"id": "53a54029",

"metadata": {},

"outputs": [],

"source": [

"import numpy as np\n",

"import tensorflow #open source used for both ML and DL for computation\n",

"from tensorflow.keras.datasets import mnist #mnist dataset\n",

"from tensorflow.keras.models import Sequential #it is a plain stack of layers\n",

"from tensorflow.keras import layers #A Layer consists of a tensor- in tensor-out computat ion funct ion\n",

"from tensorflow.keras.layers import Dense, Flatten #Dense-Dense Layer is the regular deeply connected r\n",

"#faltten -used fot flattening the input or change the dimension\n",

"from tensorflow.keras.layers import Conv2D #convolutional Layer\n",

"from keras.utils import np\_utils #used for one-hot encoding\n",

"import matplotlib.pyplot as plt #used for data visualization"

]

},

{

"cell\_type": "markdown",

"id": "41612196",

"metadata": {},

"source": [

"# Load data"

]

},

{

"cell\_type": "code",

"execution\_count": 32,

"id": "3d416c13",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

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

"11493376/11490434 [==============================] - 0s 0us/step\n",

"11501568/11490434 [==============================] - 0s 0us/step\n"

]

}

],

"source": [

"(x\_train, y\_train), (x\_test, y\_test)=mnist.load\_data () #splitting the mnist data into train and test"

]

},

{

"cell\_type": "code",

"execution\_count": 33,

"id": "087fbafd",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"(60000, 28, 28)\n",

"(10000, 28, 28)\n"

]

}

],

"source": [

"print (x\_train.shape) #shape is used for give the dimens ion values #60000-rows 28x28-pixels\n",

"print (x\_test.shape)"

]

},

{

"cell\_type": "code",

"execution\_count": 34,

"id": "4758cec8",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"array([[ 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,\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,\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,\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,\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, 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, 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, 139, 253,\n",

" 190, 2, 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, 11, 190,\n",

" 253, 70, 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, 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,\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,\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,\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,\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,\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, 39,\n",

" 148, 229, 253, 253, 253, 250, 182, 0, 0, 0, 0, 0, 0,\n",

" 0, 0],\n",

" [ 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,\n",

" 0, 0],\n",

" [ 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,\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,\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,\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,\n",

" 0, 0],\n",

" [ 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,\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,\n",

" 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,\n",

" 0, 0]], dtype=uint8)"

]

},

"execution\_count": 34,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"x\_train[0]"

]

},

{

"cell\_type": "code",

"execution\_count": 35,

"id": "ba3df521",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"<matplotlib.image.AxesImage at 0x7f0004821880>"

]

},

"execution\_count": 35,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"plt.imshow(x\_train[6000]) #ploting the index=image"

]

},

{

"cell\_type": "code",

"execution\_count": 36,

"id": "8f87b22d",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"0"

]

},

"execution\_count": 36,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"np.argmax(y\_train[6000])"

]

},

{

"cell\_type": "markdown",

"id": "a6bd217a",

"metadata": {},

"source": [

"# Reshaping Dataset"

]

},

{

"cell\_type": "code",

"execution\_count": 37,

"id": "d298102b",

"metadata": {},

"outputs": [],

"source": [

"#Reshaping to format which CNN expects (batch, height, width, channels)\n",

"x\_train=x\_train.reshape (60000, 28, 28, 1).astype('float32')\n",

"x\_test=x\_test.reshape (10000, 28, 28, 1).astype ('float32')"

]

},

{

"cell\_type": "markdown",

"id": "113e9a6a",

"metadata": {},

"source": [

"# Applying One Hot Encoding"

]

},

{

"cell\_type": "code",

"execution\_count": 38,

"id": "76ba7afe",

"metadata": {},

"outputs": [],

"source": [

"number\_of\_classes = 10 #storing the no of classes in a variable"

]

},

{

"cell\_type": "code",

"execution\_count": 39,

"id": "ff17626c",

"metadata": {},

"outputs": [],

"source": [

"y\_train = np\_utils.to\_categorical (y\_train, number\_of\_classes) #converts the output in binary format\n",

"y\_test = np\_utils.to\_categorical (y\_test, number\_of\_classes)"

]

},

{

"cell\_type": "markdown",

"id": "be53ea02",

"metadata": {},

"source": [

"# Add CNN Layers"

]

},

{

"cell\_type": "code",

"execution\_count": 40,

"id": "119a34ab",

"metadata": {},

"outputs": [],

"source": [

"#create model\n",

"model=Sequential ()"

]

},

{

"cell\_type": "code",

"execution\_count": 41,

"id": "3e49c070",

"metadata": {},

"outputs": [],

"source": [

"#adding modeL Layer\n",

"model.add(Conv2D(64, (3, 3), input\_shape=(28, 28, 1), activation='relu'))\n",

"model.add(Conv2D(32, (3, 3), activation = 'relu'))"

]

},

{

"cell\_type": "code",

"execution\_count": 42,

"id": "8f94369b",

"metadata": {},

"outputs": [],

"source": [

"#flatten the dimension of the image\n",

"model.add(Flatten())"

]

},

{

"cell\_type": "code",

"execution\_count": 43,

"id": "84765fae",

"metadata": {},

"outputs": [],

"source": [

"#output layer with 10 neurons\n",

"model.add(Dense(number\_of\_classes,activation = 'softmax'))"

]

},

{

"cell\_type": "markdown",

"id": "2048aef5",

"metadata": {},

"source": [

"# Compiling the model"

]

},

{

"cell\_type": "code",

"execution\_count": 44,

"id": "856304e2",

"metadata": {},

"outputs": [],

"source": [

"#Compile model\n",

"model.compile(loss= 'categorical\_crossentropy', optimizer=\"Adam\", metrics=['accuracy'])"

]

},

{

"cell\_type": "code",

"execution\_count": 45,

"id": "f0468d09",

"metadata": {},

"outputs": [],

"source": [

"x\_train = np.asarray(x\_train)\n",

"y\_train = np.asarray(y\_train)"

]

},

{

"cell\_type": "markdown",

"id": "974dd9c8",

"metadata": {},

"source": [

"# Train the model"

]

},

{

"cell\_type": "code",

"execution\_count": 46,

"id": "bcde8241",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Epoch 1/5\n",

"1875/1875 [==============================] - 126s 67ms/step - loss: 0.2690 - accuracy: 0.9514 - val\_loss: 0.0884 - val\_accuracy: 0.9728\n",

"Epoch 2/5\n",

"1875/1875 [==============================] - 125s 66ms/step - loss: 0.0676 - accuracy: 0.9789 - val\_loss: 0.0803 - val\_accuracy: 0.9788\n",

"Epoch 3/5\n",

"1875/1875 [==============================] - 125s 67ms/step - loss: 0.0458 - accuracy: 0.9852 - val\_loss: 0.0791 - val\_accuracy: 0.9788\n",

"Epoch 4/5\n",

"1875/1875 [==============================] - 125s 67ms/step - loss: 0.0387 - accuracy: 0.9883 - val\_loss: 0.1079 - val\_accuracy: 0.9759\n",

"Epoch 5/5\n",

"1875/1875 [==============================] - 125s 67ms/step - loss: 0.0280 - accuracy: 0.9909 - val\_loss: 0.0991 - val\_accuracy: 0.9774\n"

]

},

{

"data": {

"text/plain": [

"<keras.callbacks.History at 0x7f0004e4a640>"

]

},

"execution\_count": 46,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"#fit the model\n",

"model.fit(x\_train, y\_train, validation\_data=(x\_test, y\_test), epochs=5, batch\_size=32)"

]

},

{

"cell\_type": "markdown",

"id": "81738cb4",

"metadata": {},

"source": [

"# Observing the metrics"

]

},

{

"cell\_type": "code",

"execution\_count": 47,

"id": "00a1c2dc",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Metrics (Test loss &Test Accuracy) : \n",

"[0.09910603612661362, 0.977400004863739]\n"

]

}

],

"source": [

"# Final evaluation of the model\n",

"metrics = model.evaluate(x\_test, y\_test, verbose=0)\n",

"print(\"Metrics (Test loss &Test Accuracy) : \")\n",

"print(metrics)"

]

},

{

"cell\_type": "markdown",

"id": "b3add595",

"metadata": {},

"source": [

"# Test The Model"

]

},

{

"cell\_type": "code",

"execution\_count": 48,

"id": "0508c462",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"[[9.1516389e-13 8.1778777e-19 2.4542002e-14 1.7823329e-07 5.2257418e-04\n",

" 5.8763407e-09 6.2800168e-17 3.1880148e-07 6.3142506e-03 9.9316275e-01]]\n"

]

}

],

"source": [

"prediction=model.predict(x\_test[6000:6001])\n",

"print(prediction)"

]

},

{

"cell\_type": "code",

"execution\_count": 49,

"id": "45a591fc",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"<matplotlib.image.AxesImage at 0x7f00044a3370>"

]

},

"execution\_count": 49,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"plt.imshow(x\_test[6000])"

]

},

{

"cell\_type": "code",

"execution\_count": 50,

"id": "4cf45ba6",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"[9]\n"

]

}

],

"source": [

"import numpy as np\n",

"print(np.argmax(prediction, axis=1)) #printing our Labels from first 4 images"

]

},

{

"cell\_type": "code",

"execution\_count": 51,

"id": "81f007d9",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"9"

]

},

"execution\_count": 51,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"np.argmax(y\_test[6000:6001]) #printing the actual labels"

]

},

{

"cell\_type": "markdown",

"id": "0e2f8106",

"metadata": {},

"source": [

"# Save The model"

]

},

{

"cell\_type": "code",

"execution\_count": 52,

"id": "8fc89e72",

"metadata": {},

"outputs": [],

"source": [

"# Save the model\n",

"model.save('models/mnistCNN.h5')"

]

},

{

"cell\_type": "code",

"execution\_count": 53,

"id": "d3fe4109",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"/home/wsuser/work/models\n"

]

}

],

"source": [

"cd models"

]

},

{

"cell\_type": "code",

"execution\_count": 54,

"id": "502acee5",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"mnistCNN.h5\r\n"

]

}

],

"source": [

"!tar -zcvf handwritten-digit-recognition-model\_new.tgz mnistCNN.h5"

]

},

{

"cell\_type": "code",

"execution\_count": 55,

"id": "86720654",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Collecting watson-machine-learning-client\n",

" Downloading watson\_machine\_learning\_client-1.0.391-py3-none-any.whl (538 kB)\n",

"\u001b[K |████████████████████████████████| 538 kB 21.5 MB/s eta 0:00:01\n",

"\u001b[?25hRequirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.3.3)\n",

"Requirement already satisfied: boto3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.18.21)\n",

"Requirement already satisfied: tqdm in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (4.62.3)\n",

"Requirement already satisfied: pandas in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.3.4)\n",

"Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.8.9)\n",

"Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2022.9.24)\n",

"Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.26.7)\n",

"Requirement already satisfied: ibm-cos-sdk in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2.11.0)\n",

"Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2.26.0)\n",

"Requirement already satisfied: botocore<1.22.0,>=1.21.21 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (1.21.41)\n",

"Requirement already satisfied: s3transfer<0.6.0,>=0.5.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (0.5.0)\n",

"Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (0.10.0)\n",

"Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from botocore<1.22.0,>=1.21.21->boto3->watson-machine-learning-client) (2.8.2)\n",

"Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->botocore<1.22.0,>=1.21.21->boto3->watson-machine-learning-client) (1.15.0)\n",

"Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk->watson-machine-learning-client) (2.11.0)\n",

"Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk->watson-machine-learning-client) (2.11.0)\n",

"Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->watson-machine-learning-client) (2.0.4)\n",

"Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->watson-machine-learning-client) (3.3)\n",

"Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client) (2021.3)\n",

"Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client) (1.20.3)\n",

"Installing collected packages: watson-machine-learning-client\n",

"Successfully installed watson-machine-learning-client-1.0.391\n"

]

}

],

"source": [

"!pip install watson-machine-learning-client --upgrade"

]

},

{

"cell\_type": "markdown",

"id": "bd05ddda",

"metadata": {},

"source": [

"# Cloud deploy"

]

},

{

"cell\_type": "code",

"execution\_count": 56,

"id": "20abb86b",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"<ibm\_watson\_machine\_learning.client.APIClient at 0x7f00044af820>"

]

},

"execution\_count": 56,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"from ibm\_watson\_machine\_learning import APIClient\n",

"credentials ={\n",

" \"url\":\"https://jp-tok.ml.cloud.ibm.com\",\n",

" \"apikey\":\"BHyalu2c7JN6n9cnvAVULvSKRYFVLMQ\_m51toZ9Yk0nS\"\n",

"}\n",

"client = APIClient(credentials)\n",

"client"

]

},

{

"cell\_type": "code",

"execution\_count": 57,

"id": "30ad9b4a",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"{'resources': [{'entity': {'compute': [{'crn': 'crn:v1:bluemix:public:pm-20:jp-tok:a/53f9f6400d0d44889534e8abcd2dfe39:0f4376b6-c944-4b27-b23e-48b54d8f4bbd::',\n",

" 'guid': '0f4376b6-c944-4b27-b23e-48b54d8f4bbd',\n",

" 'name': 'Watson Machine Learning-sp',\n",

" 'type': 'machine\_learning'}],\n",

" 'description': '',\n",

" 'name': 'digitrecognition',\n",

" 'scope': {'bss\_account\_id': '53f9f6400d0d44889534e8abcd2dfe39'},\n",

" 'stage': {'production': False},\n",

" 'status': {'state': 'active'},\n",

" 'storage': {'properties': {'bucket\_name': '63888f6f-d1ef-475c-a8d8-a2e4957bb673',\n",

" 'bucket\_region': 'jp-tok-standard',\n",

" 'credentials': {'admin': {'access\_key\_id': '834b3358ebb945fb9ebbb4020cd2bf0e',\n",

" 'api\_key': '2JONUuuPfYzZzPGzTp1J7dwwjNTpkOsyxdW5gx\_vml3m',\n",

" 'secret\_access\_key': '1ed5b29fdd6c65b48ca72963b6177133ce51a7b23acdcaa5',\n",

" 'service\_id': 'ServiceId-a2495f73-f36b-4fa1-9991-976f110c1a4f'},\n",

" 'editor': {'access\_key\_id': 'b56d445c54794369b2a4e0115e166605',\n",

" 'api\_key': 'wcwCBLp8z4xpgnsEDeUCOZquAovrWhXu2wcF9Kz5Vhpe',\n",

" 'resource\_key\_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/53f9f6400d0d44889534e8abcd2dfe39:d8fa8aee-cd61-4757-9543-a61f55971074::',\n",

" 'secret\_access\_key': '84b0b128f52e57c025e6517604a06212b8d19f0b349eeea3',\n",

" 'service\_id': 'ServiceId-4e1f87ab-27bc-4654-b6ea-667a8640c7e0'},\n",

" 'viewer': {'access\_key\_id': '558109e942fb4b1eb020c881f04d8588',\n",

" 'api\_key': 'zWS-VZ\_d9GfkDt1XnCmWoOA6liYXNnGtrPwJt2fI0UI5',\n",

" 'resource\_key\_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/53f9f6400d0d44889534e8abcd2dfe39:d8fa8aee-cd61-4757-9543-a61f55971074::',\n",

" 'secret\_access\_key': '3e2d27ab9d4041707cfa721daa638d1ad57f42ab8df94c09',\n",

" 'service\_id': 'ServiceId-93177c88-86e2-470d-b5bf-3aed99d093a8'}},\n",

" 'endpoint\_url': 'https://s3.jp-tok.cloud-object-storage.appdomain.cloud',\n",

" 'guid': 'd8fa8aee-cd61-4757-9543-a61f55971074',\n",

" 'resource\_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/53f9f6400d0d44889534e8abcd2dfe39:d8fa8aee-cd61-4757-9543-a61f55971074::'},\n",

" 'type': 'bmcos\_object\_storage'}},\n",

" 'metadata': {'created\_at': '2022-10-31T10:33:07.575Z',\n",

" 'creator\_id': 'IBMid-667000CZ2Y',\n",

" 'id': 'aa24227a-9f01-493f-90e6-1b6132057fc6',\n",

" 'updated\_at': '2022-10-31T10:33:25.148Z',\n",

" 'url': '/v2/spaces/aa24227a-9f01-493f-90e6-1b6132057fc6'}}]}"

]

},

"execution\_count": 57,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"client.spaces.get\_details()"

]

},

{

"cell\_type": "code",

"execution\_count": 58,

"id": "36192134",

"metadata": {},

"outputs": [],

"source": [

"def guid\_from\_space\_name(client,deploy):\n",

" space = client.spaces.get\_details()\n",

" return (next(item for item in space['resources'] if item['entity']['name']==deploy)['metadata']['id'])"

]

},

{

"cell\_type": "code",

"execution\_count": 59,

"id": "8db09ec8",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Space UID = aa24227a-9f01-493f-90e6-1b6132057fc6\n"

]

}

],

"source": [

"space\_uid = guid\_from\_space\_name(client,'digitrecognition')\n",

"print(\"Space UID = \" + space\_uid)"

]

},

{

"cell\_type": "code",

"execution\_count": 60,

"id": "dd499fbd",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"'SUCCESS'"

]

},

"execution\_count": 60,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"client.set.default\_space(space\_uid)"

]

},

{

"cell\_type": "code",

"execution\_count": 61,

"id": "b1037d4f",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"------------------------------- ------------------------------------ ----\n",

"NAME ASSET\_ID TYPE\n",

"default\_py3.6 0062b8c9-8b7d-44a0-a9b9-46c416adcbd9 base\n",

"kernel-spark3.2-scala2.12 020d69ce-7ac1-5e68-ac1a-31189867356a base\n",

"pytorch-onnx\_1.3-py3.7-edt 069ea134-3346-5748-b513-49120e15d288 base\n",

"scikit-learn\_0.20-py3.6 09c5a1d0-9c1e-4473-a344-eb7b665ff687 base\n",

"spark-mllib\_3.0-scala\_2.12 09f4cff0-90a7-5899-b9ed-1ef348aebdee base\n",

"pytorch-onnx\_rt22.1-py3.9 0b848dd4-e681-5599-be41-b5f6fccc6471 base\n",

"ai-function\_0.1-py3.6 0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda base\n",

"shiny-r3.6 0e6e79df-875e-4f24-8ae9-62dcc2148306 base\n",

"tensorflow\_2.4-py3.7-horovod 1092590a-307d-563d-9b62-4eb7d64b3f22 base\n",

"pytorch\_1.1-py3.6 10ac12d6-6b30-4ccd-8392-3e922c096a92 base\n",

"tensorflow\_1.15-py3.6-ddl 111e41b3-de2d-5422-a4d6-bf776828c4b7 base\n",

"runtime-22.1-py3.9 12b83a17-24d8-5082-900f-0ab31fbfd3cb base\n",

"scikit-learn\_0.22-py3.6 154010fa-5b3b-4ac1-82af-4d5ee5abbc85 base\n",

"default\_r3.6 1b70aec3-ab34-4b87-8aa0-a4a3c8296a36 base\n",

"pytorch-onnx\_1.3-py3.6 1bc6029a-cc97-56da-b8e0-39c3880dbbe7 base\n",

"kernel-spark3.3-r3.6 1c9e5454-f216-59dd-a20e-474a5cdf5988 base\n",

"pytorch-onnx\_rt22.1-py3.9-edt 1d362186-7ad5-5b59-8b6c-9d0880bde37f base\n",

"tensorflow\_2.1-py3.6 1eb25b84-d6ed-5dde-b6a5-3fbdf1665666 base\n",

"spark-mllib\_3.2 20047f72-0a98-58c7-9ff5-a77b012eb8f5 base\n",

"tensorflow\_2.4-py3.8-horovod 217c16f6-178f-56bf-824a-b19f20564c49 base\n",

"runtime-22.1-py3.9-cuda 26215f05-08c3-5a41-a1b0-da66306ce658 base\n",

"do\_py3.8 295addb5-9ef9-547e-9bf4-92ae3563e720 base\n",

"autoai-ts\_3.8-py3.8 2aa0c932-798f-5ae9-abd6-15e0c2402fb5 base\n",

"tensorflow\_1.15-py3.6 2b73a275-7cbf-420b-a912-eae7f436e0bc base\n",

"kernel-spark3.3-py3.9 2b7961e2-e3b1-5a8c-a491-482c8368839a base\n",

"pytorch\_1.2-py3.6 2c8ef57d-2687-4b7d-acce-01f94976dac1 base\n",

"spark-mllib\_2.3 2e51f700-bca0-4b0d-88dc-5c6791338875 base\n",

"pytorch-onnx\_1.1-py3.6-edt 32983cea-3f32-4400-8965-dde874a8d67e base\n",

"spark-mllib\_3.0-py37 36507ebe-8770-55ba-ab2a-eafe787600e9 base\n",

"spark-mllib\_2.4 390d21f8-e58b-4fac-9c55-d7ceda621326 base\n",

"xgboost\_0.82-py3.6 39e31acd-5f30-41dc-ae44-60233c80306e base\n",

"pytorch-onnx\_1.2-py3.6-edt 40589d0e-7019-4e28-8daa-fb03b6f4fe12 base\n",

"default\_r36py38 41c247d3-45f8-5a71-b065-8580229facf0 base\n",

"autoai-ts\_rt22.1-py3.9 4269d26e-07ba-5d40-8f66-2d495b0c71f7 base\n",

"autoai-obm\_3.0 42b92e18-d9ab-567f-988a-4240ba1ed5f7 base\n",

"pmml-3.0\_4.3 493bcb95-16f1-5bc5-bee8-81b8af80e9c7 base\n",

"spark-mllib\_2.4-r\_3.6 49403dff-92e9-4c87-a3d7-a42d0021c095 base\n",

"xgboost\_0.90-py3.6 4ff8d6c2-1343-4c18-85e1-689c965304d3 base\n",

"pytorch-onnx\_1.1-py3.6 50f95b2a-bc16-43bb-bc94-b0bed208c60b base\n",

"autoai-ts\_3.9-py3.8 52c57136-80fa-572e-8728-a5e7cbb42cde base\n",

"spark-mllib\_2.4-scala\_2.11 55a70f99-7320-4be5-9fb9-9edb5a443af5 base\n",

"spark-mllib\_3.0 5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9 base\n",

"autoai-obm\_2.0 5c2e37fa-80b8-5e77-840f-d912469614ee base\n",

"spss-modeler\_18.1 5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b base\n",

"cuda-py3.8 5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e base\n",

"autoai-kb\_3.1-py3.7 632d4b22-10aa-5180-88f0-f52dfb6444d7 base\n",

"pytorch-onnx\_1.7-py3.8 634d3cdc-b562-5bf9-a2d4-ea90a478456b base\n",

"spark-mllib\_2.3-r\_3.6 6586b9e3-ccd6-4f92-900f-0f8cb2bd6f0c base\n",

"tensorflow\_2.4-py3.7 65e171d7-72d1-55d9-8ebb-f813d620c9bb base\n",

"spss-modeler\_18.2 687eddc9-028a-4117-b9dd-e57b36f1efa5 base\n",

"pytorch-onnx\_1.2-py3.6 692a6a4d-2c4d-45ff-a1ed-b167ee55469a base\n",

"spark-mllib\_2.3-scala\_2.11 7963efe5-bbec-417e-92cf-0574e21b4e8d base\n",

"spark-mllib\_2.4-py37 7abc992b-b685-532b-a122-a396a3cdbaab base\n",

"caffe\_1.0-py3.6 7bb3dbe2-da6e-4145-918d-b6d84aa93b6b base\n",

"pytorch-onnx\_1.7-py3.7 812c6631-42b7-5613-982b-02098e6c909c base\n",

"cuda-py3.6 82c79ece-4d12-40e6-8787-a7b9e0f62770 base\n",

"tensorflow\_1.15-py3.6-horovod 8964680e-d5e4-5bb8-919b-8342c6c0dfd8 base\n",

"hybrid\_0.1 8c1a58c6-62b5-4dc4-987a-df751c2756b6 base\n",

"pytorch-onnx\_1.3-py3.7 8d5d8a87-a912-54cf-81ec-3914adaa988d base\n",

"caffe-ibm\_1.0-py3.6 8d863266-7927-4d1e-97d7-56a7f4c0a19b base\n",

"spss-modeler\_17.1 902d0051-84bd-4af6-ab6b-8f6aa6fdeabb base\n",

"do\_12.10 9100fd72-8159-4eb9-8a0b-a87e12eefa36 base\n",

"do\_py3.7 9447fa8b-2051-4d24-9eef-5acb0e3c59f8 base\n",

"spark-mllib\_3.0-r\_3.6 94bb6052-c837-589d-83f1-f4142f219e32 base\n",

"cuda-py3.7-opence 94e9652b-7f2d-59d5-ba5a-23a414ea488f base\n",

"nlp-py3.8 96e60351-99d4-5a1c-9cc0-473ac1b5a864 base\n",

"cuda-py3.7 9a44990c-1aa1-4c7d-baf8-c4099011741c base\n",

"hybrid\_0.2 9b3f9040-9cee-4ead-8d7a-780600f542f7 base\n",

"spark-mllib\_3.0-py38 9f7a8fc1-4d3c-5e65-ab90-41fa8de2d418 base\n",

"autoai-kb\_3.3-py3.7 a545cca3-02df-5c61-9e88-998b09dc79af base\n",

"spark-mllib\_3.0-py39 a6082a27-5acc-5163-b02c-6b96916eb5e0 base\n",

"runtime-22.1-py3.9-do a7e7dbf1-1d03-5544-994d-e5ec845ce99a base\n",

"default\_py3.8 ab9e1b80-f2ce-592c-a7d2-4f2344f77194 base\n",

"tensorflow\_rt22.1-py3.9 acd9c798-6974-5d2f-a657-ce06e986df4d base\n",

"kernel-spark3.2-py3.9 ad7033ee-794e-58cf-812e-a95f4b64b207 base\n",

"autoai-obm\_2.0 with Spark 3.0 af10f35f-69fa-5d66-9bf5-acb58434263a base\n",

"default\_py3.7\_opence c2057dd4-f42c-5f77-a02f-72bdbd3282c9 base\n",

"tensorflow\_2.1-py3.7 c4032338-2a40-500a-beef-b01ab2667e27 base\n",

"do\_py3.7\_opence cc8f8976-b74a-551a-bb66-6377f8d865b4 base\n",

"spark-mllib\_3.3 d11f2434-4fc7-58b7-8a62-755da64fdaf8 base\n",

"autoai-kb\_3.0-py3.6 d139f196-e04b-5d8b-9140-9a10ca1fa91a base\n",

"spark-mllib\_3.0-py36 d82546d5-dd78-5fbb-9131-2ec309bc56ed base\n",

"autoai-kb\_3.4-py3.8 da9b39c3-758c-5a4f-9cfd-457dd4d8c395 base\n",

"kernel-spark3.2-r3.6 db2fe4d6-d641-5d05-9972-73c654c60e0a base\n",

"autoai-kb\_rt22.1-py3.9 db6afe93-665f-5910-b117-d879897404d9 base\n",

"tensorflow\_rt22.1-py3.9-horovod dda170cc-ca67-5da7-9b7a-cf84c6987fae base\n",

"autoai-ts\_1.0-py3.7 deef04f0-0c42-5147-9711-89f9904299db base\n",

"tensorflow\_2.1-py3.7-horovod e384fce5-fdd1-53f8-bc71-11326c9c635f base\n",

"default\_py3.7 e4429883-c883-42b6-87a8-f419d64088cd base\n",

"do\_22.1 e51999ba-6452-5f1f-8287-17228b88b652 base\n",

"autoai-obm\_3.2 eae86aab-da30-5229-a6a6-1d0d4e368983 base\n",

"do\_20.1 f686cdd9-7904-5f9d-a732-01b0d6b10dc5 base\n",

"scikit-learn\_0.19-py3.6 f963fa9d-4bb7-5652-9c5d-8d9289ef6ad9 base\n",

"tensorflow\_2.4-py3.8 fe185c44-9a99-5425-986b-59bd1d2eda46 base\n",

"------------------------------- ------------------------------------ ----\n"

]

}

],

"source": [

"client.software\_specifications.list(limit=100)"

]

},

{

"cell\_type": "code",

"execution\_count": 62,

"id": "841ad2ab",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"'acd9c798-6974-5d2f-a657-ce06e986df4d'"

]

},

"execution\_count": 62,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"software\_space\_uid = client.software\_specifications.get\_uid\_by\_name('tensorflow\_rt22.1-py3.9')\n",

"software\_space\_uid"

]

},

{

"cell\_type": "code",

"execution\_count": 63,

"id": "82c978e1",

"metadata": {},

"outputs": [],

"source": [

"model\_details = client.repository.store\_model(model='handwritten-digit-recognition-model\_new.tgz',meta\_props={\n",

" client.repository.ModelMetaNames.NAME:\"CNN Digit recognition model\",\n",

" client.repository.ModelMetaNames.TYPE:\"tensorflow\_2.7\",\n",

" client.repository.ModelMetaNames.SOFTWARE\_SPEC\_UID:software\_space\_uid\n",

"})"

]

},

{

"cell\_type": "code",

"execution\_count": 64,

"id": "12700c5d",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"{'entity': {'hybrid\_pipeline\_software\_specs': [],\n",

" 'software\_spec': {'id': 'acd9c798-6974-5d2f-a657-ce06e986df4d',\n",

" 'name': 'tensorflow\_rt22.1-py3.9'},\n",

" 'type': 'tensorflow\_2.7'},\n",

" 'metadata': {'created\_at': '2022-11-01T10:15:40.847Z',\n",

" 'id': '97d463b1-45ee-47f7-b8af-aed338794ce1',\n",

" 'modified\_at': '2022-11-01T10:15:44.197Z',\n",

" 'name': 'CNN Digit recognition model',\n",

" 'owner': 'IBMid-667000CZ2Y',\n",

" 'resource\_key': '84636ddb-9fa8-47e4-8fa4-3c36731e2fe6',\n",

" 'space\_id': 'aa24227a-9f01-493f-90e6-1b6132057fc6'},\n",

" 'system': {'warnings': []}}"

]

},

"execution\_count": 64,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"model\_details"

]

},

{

"cell\_type": "code",

"execution\_count": 65,

"id": "edd79f04",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"'97d463b1-45ee-47f7-b8af-aed338794ce1'"

]

},

"execution\_count": 65,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"model\_id = client.repository.get\_model\_id(model\_details)\n",

"model\_id"

]

},

{

"cell\_type": "code",

"execution\_count": 66,

"id": "0a26ad43",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Successfully saved model content to file: 'DigitRecog\_IBM\_model.tar.gz'\n"

]

},

{

"data": {

"text/plain": [

"'/home/wsuser/work/models/DigitRecog\_IBM\_model.tar.gz'"

]

},

"execution\_count": 66,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"client.repository.download(model\_id,'DigitRecog\_IBM\_model.tar.gz')"

]

},

{

"cell\_type": "code",

"execution\_count": 67,

"id": "e3b67528",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"DigitRecog\_IBM\_model.tar.gz mnistCNN.h5\r\n",

"handwritten-digit-recognition-model\_new.tgz\r\n"

]

}

],

"source": [

"ls"

]

},

{

"cell\_type": "markdown",

"id": "62f82a21",

"metadata": {},

"source": [

"# TEST MODEL"

]

},

{

"cell\_type": "code",

"execution\_count": 286,

"id": "b3bc5a3d",

"metadata": {},

"outputs": [],

"source": [

"from tensorflow.keras.models import load\_model\n",

"from keras.preprocessing import image\n",

"from PIL import Image\n",

"import numpy as np"

]

},

{

"cell\_type": "code",

"execution\_count": 287,

"id": "0e416ac2",

"metadata": {},

"outputs": [],

"source": [

"model = load\_model(\"mnistCNN.h5\")"

]

},

{

"cell\_type": "code",

"execution\_count": 288,

"id": "d9d565ad",

"metadata": {},

"outputs": [],

"source": [

"\n",

"import os, types\n",

"import pandas as pd\n",

"from botocore.client import Config\n",

"import ibm\_boto3\n",

"\n",

"def \_\_iter\_\_(self): return 0\n",

"\n",

"# @hidden\_cell\n",

"# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.\n",

"# You might want to remove those credentials before you share the notebook.\n",

"cos\_client = ibm\_boto3.client(service\_name='s3',\n",

" ibm\_api\_key\_id='is\_QZGPyU8oxZr3W-td-LCHXS3QPMaWArILi18FdSyGT',\n",

" ibm\_auth\_endpoint=\"https://iam.cloud.ibm.com/oidc/token\",\n",

" config=Config(signature\_version='oauth'),\n",

" endpoint\_url='https://s3.private.ap.cloud-object-storage.appdomain.cloud')\n",

"\n",

"bucket = 'handwrittenimagerecognition-donotdelete-pr-8tlrnykut46vpi'\n",

"object\_key = 'mnist-dataset-1024x424 (2).png'\n",

"\n",

"streaming\_body\_1 = cos\_client.get\_object(Bucket=bucket, Key=object\_key)['Body']\n",

"\n",

"# Your data file was loaded into a botocore.response.StreamingBody object.\n",

"# Please read the documentation of ibm\_boto3 and pandas to learn more about the possibilities to load the data.\n",

"# ibm\_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/\n",

"# pandas documentation: http://pandas.pydata.org/\n"

]

},

{

"cell\_type": "code",

"execution\_count": 289,

"id": "0af4c026",

"metadata": {},

"outputs": [],

"source": [

"img = Image.open(streaming\_body\_1).convert(\"L\") # convert image to monochrome\n",

"img = img.resize( (28,28) ) # resizing of input image"

]

},

{

"cell\_type": "code",

"execution\_count": 290,

"id": "758a186b",

"metadata": {},

"outputs": [

{

"data": {

"image/png": "\n",

"text/plain": [

"<PIL.Image.Image image mode=L size=28x28 at 0x7EFFC445F430>"

]

},

"execution\_count": 290,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"img"

]

},

{

"cell\_type": "code",

"execution\_count": 291,

"id": "ad56c441",

"metadata": {},

"outputs": [],

"source": [

"im2arr = np.array(img) #converting to image\n",

"im2arr = im2arr.reshape(1, 28, 28, 1) #reshaping according to our requirement"

]

},

{

"cell\_type": "code",

"execution\_count": 292,

"id": "4ffcdfc3",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"[[1.0000000e+00 5.3912803e-17 3.9648812e-11 2.0051219e-16 5.1053910e-18\n",

" 2.9315760e-12 7.0849349e-13 2.0999634e-16 2.9204243e-09 7.4729778e-11]]\n"

]

}

],

"source": [

"pred = model.predict(im2arr)\n",

"print(pred)"

]

},

{

"cell\_type": "code",

"execution\_count": 293,

"id": "81258a25",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"[0]\n"

]

}

],

"source": [

"print(np.argmax(pred, axis=1)) #printing our Labels"

]

}

],

"metadata": {

"kernelspec": {

"display\_name": "Python 3.9",

"language": "python",

"name": "python3"

},

"language\_info": {

"codemirror\_mode": {

"name": "ipython",

"version": 3

},

"file\_extension": ".py",

"mimetype": "text/x-python",

"name": "python",

"nbconvert\_exporter": "python",

"pygments\_lexer": "ipython3",

"version": "3.9.13"

}

},

"nbformat": 4,

"nbformat\_minor": 5

}