**18CSC305J-Artificial Intelligence**

**LAB 14 – Applying deep learning methods to solve an application**

**Aim:** Applying deep learning methods to solve an application.

**Description:**

Deep Learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks. Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. Deep learning is a key technology behind driverless cars, enabling them to recognize a stop sign, or to distinguish a pedestrian from a lamppost.In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound. Deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance. Models are trained by using a large set of labeled data and neural network architectures that contain many layers.

**CODE :**

!pip install tensorflow

!pip install numpy

!pip install keras

!pip install pillow

import keras

from keras.datasets import mnist

from keras.models import Sequential

from keras.layers import Dense, Dropout, Flatten

from keras.layers import Conv2D, MaxPooling2D

from keras import backend as K

from keras.utils import np\_utils

(x\_train, y\_train), (x\_test, y\_test) = mnist.load\_data() print(x\_train.shape, y\_train.shape)

model = Sequential()

model.add(Conv2D(32, kernel\_size=(3, 3),activation='relu',input\_shape=input\_shape))

model.add(Conv2D(64, (3, 3), activation='relu'))

model.add(MaxPooling2D(pool\_size=(2, 2)))

model.add(Dropout(0.25))

model.add(Flatten())

model.add(Dense(256, activation='relu'))

model.add(Dropout(0.5))

model.add(Dense(num\_classes, activation='softmax')) y\_train = keras.utils.np\_utils.to\_categorical(y\_train, num\_classes) y\_test = keras.utils.np\_utils.to\_categorical(y\_test, num\_classes) #Normalize the data

x\_train = x\_train.astype('float32')

x\_test = x\_test.astype('float32')

x\_train /= 255

x\_test /= 255

print('x\_train shape:', x\_train.shape)

print(x\_train.shape[0], 'train samples')

print(x\_test.shape[0], 'test samples')

batch\_size = 128

epochs = 10

from tensorflow import keras

model.compile(loss=keras.losses.categorical\_crossentropy,optimizer = keras.optimizers.Adadelta(),metrics=['accuracy'])

hist =

model.fit(x\_train,y\_train,batch\_size=batch\_size,epochs=epochs,verbose=1,va lidation\_data=(x\_test, y\_test))

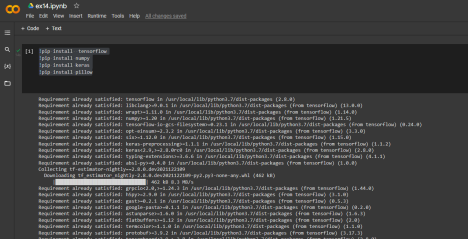
print("The model has successfully trained")

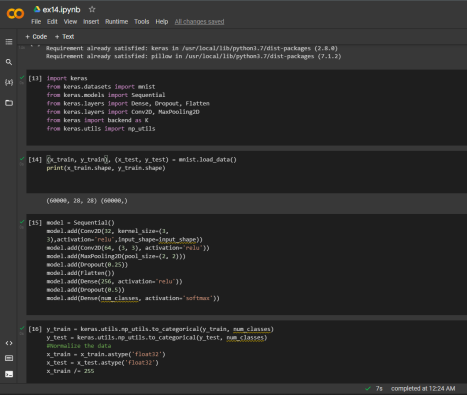
score = model.evaluate(x\_test, y\_test, verbose=0)

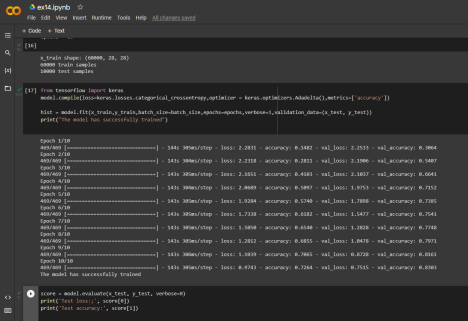
print('Test loss:;', score[0])

print('Test accuracy:', score[1])

**OUTPUT :**

****

****

****

****

**RESULT :**

Hence we applied deep learning methods to solve an application successfully.