**EXP 14: APPLYING DEEP LEARNING METHODS TO SOLVE REAL WORLD PROBLEMS**

**Name:** Praveen Kumar K

**Reg no:** RA1911030010069

**Subject:** Artificial Intelligence

**AIM**

Applying Deep Learning methods to solve real world problem.

**ALGORITHM**

1. Deep learning is a type of [machine learning](https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML) and artificial intelligence ([AI](https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence)) that imitates the way humans gain certain types of knowledge.
2. Deep learning is an important element of data science, which includes statistics and [predictive modelling](https://www.techtarget.com/searchenterpriseai/definition/predictive-modeling). It is extremely beneficial to data scientists who are tasked with collecting, analysing, and interpreting large amounts of data; deep learning makes this process faster and easier.
3. Deep Neural Networks (DNNs) are such types of networks where each layer can perform complex operations such as representation and abstraction that make sense of images, sound, and text.

**CODE**

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

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

print(x\_train.shape, y\_train.shape)

num\_classes = 10

x\_train = x\_train.reshape(x\_train.shape[0], 28, 28, 1)

x\_test = x\_test.reshape(x\_test.shape[0], 28, 28, 1)

input\_shape = (28, 28, 1)

y\_train = keras.utils.to\_categorical(y\_train, num\_classes)

y\_test = keras.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

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'))

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, validation\_data=(x\_test, y\_test))

print("The model has successfully trained")

model.save('mnist.h5')

print("Saving the model as mnist.h5")

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

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

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

**OUTPUT**

A screenshot of a computer

Description automatically generated

Chart

Description automatically generated with medium confidence

A picture containing text

Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

A picture containing text

Description automatically generated

**RESULT**

Thus, we successfully solved one real world life problem using Deep Learning methods.