

Name: 7ejas Redkar

PRN: 1032210937

Panel-C, Batch: C2

ROLL NO: PC-44

ATES Lab Assignment - 8

7itle - Implement a Neural network you a real-life application

Aim- Implement d' Understand working of Neural network for a vieal-life application: Face Recognition with Python.

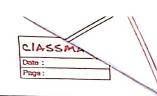
Objective - To vaturdy & implement face vacognition using Python & the open-source library

OpenCV

* Theovey

1) Difference between Machine learning & Deep learning

Machine learning is a broader concept that involves
the development of algorithms & models that
enable computers to learn & make predictions our
decisions based on data. It encompasses various
techniques, including traditional statistical methods
& newer approaches like neural networks.



Deep learning is a wilbrest of machine learning that upperight cally focuses on newcoll networks with multiple layers (deep newral networks). It is inspired by the structure & function of the human brain, allowing it to automatically learn hierarchical proposesontations of data. Deep learning has been particularly conceessful in tasks couch as image & speech vie cognition.

2) What is open CV? How does it work?

Open (V (Open Source (omputer Vision) is a liburary of progreamming functions your computer vision tasks. It provides tools for images & Video processing, enabling computer vision applications

3) What are Newwal Networks? How working of OpenCV? Write in boriet about them is different from Newral Network?

Neural Netwoodles are computational models inspired by the human birain, processing information through interconnected nodes. Open cris a computer vision liburary, while neural networks focus on learning patterns from data.

Input - Input an image with a human face is it.

```
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
# Sample input data (features)
X = np.array([[0, 0],
        [0, 1],
        [1, 0],
        [1, 1]])
# Sample output data (labels)
y = np.array([0, 1, 1, 0])
# Build a simple feedforward neural network
model = Sequential()
model.add(Dense(4, input_dim=2, activation='relu')) # Hidden layer with 4 neurons and
ReLU activation
model.add(Dense(1, activation='sigmoid'))
                                                # Output layer with 1 neuron and
Sigmoid activation
```

CODE:

Compile the model

```
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

# Train the model
model.fit(X, y, epochs=500, verbose=0)

# Test the model
sample_input = np.array([[0, 0]])
predicted_output = model.predict(sample_input)
print(f"Sample Input: {sample_input}")
print(f"Predicted Output: {predicted_output}")

Input:
[[0 0]]

Output:
[[0.03575368]]
```