EXPERIMENT – 16 Feedforward Neural Network (from scratch, one hidden layer)

AIM:

Tiny NN for binary classification (sigmoid activations).

CODE:

# simple\_nn.py

import numpy as np

def sigmoid(x): return 1/(1+np.exp(-x))

def train(X,y,epochs=1000,lr=0.1):

n\_in=X.shape[1]; n\_h=4; n\_out=1

W1=np.random.randn(n\_in,n\_h); b1=np.zeros(n\_h)

W2=np.random.randn(n\_h,n\_out); b2=np.zeros(n\_out)

for e in range(epochs):

Z1=sigmoid(X.dot(W1)+b1)

Z2=sigmoid(Z1.dot(W2)+b2)

err=Z2-y

dW2=Z1.T.dot(err\*Z2\*(1-Z2))

db2=(err\*Z2\*(1-Z2)).sum(axis=0)

dW1=X.T.dot((err\*Z2\*(1-Z2)).dot(W2.T)\*(Z1\*(1-Z1)))

db1=((err\*Z2\*(1-Z2)).dot(W2.T)\*(Z1\*(1-Z1))).sum(axis=0)

W1-=lr\*dW1; b1-=lr\*db1

W2-=lr\*dW2; b2-=lr\*db2

return (W1,b1,W2,b2)

if \_\_name\_\_=='\_\_main\_\_':

X=np.array([[0,0],[0,1],[1,0],[1,1]])

y=np.array([[0],[1],[1],[0]]) # XOR (hard for 1 hidden but example)

W1,b1,W2,b2=train(X,y,epochs=5000,lr=0.1)

out = sigmoid(sigmoid(X.dot(W1)+b1).dot(W2)+b2)

print(out.round(2))

OUTPUT:

