# 4. Implement and test MLP trained with back-propagation algorithm

clear all;

close all;

clc;

i1 = 0.05;

i2 = 0.1;

op1 = 0.01;

op2 = 0.99;

b1 = 0.35;

b2 = 0.6;

n = 0.5;

w = [0.15 0.2 0.05 0.13 0.4 0.45 0.5 0.55];

neth1 = w(1,1)\*i1+w(1,2)\*i2+b1\*1;

neth2 = w(1,3)\*i1+w(1,4)\*i2+b1\*1;

oh1 = 1/(1+exp(-neth1));

oh2 = 1/(1+exp(-neth2));

neto1 = w(1,5)\*oh1+w(1,6)\*oh2+ b2\*1;

outo1 = 1/(1+exp(-neto1));

neto2 = w(1,7)\*oh1+w(1,8)\*oh2+ b2\*1;

outo2 = 1/(1+exp(-neto2));

et1 = 0.5\*((op1-outo1)^2);

et2 = 0.5\*((op2-outo2)^2);

et = et1+et2;

det = outo1-op1;

douto1 = outo1\*(1-outo1);

dneto1 = oh1;

dett = det\*douto1\*dneto1;

do1 = det\*douto1;

wn(1,5) = w(1,5)-0.5\*do1\*oh1;

wn(1,6) = w(1,6)-0.5\*do1\*oh1;

wn(1,7) = w(1,7)-0.5\*do1\*oh1;

wn(1,8) = w(1,8)-0.5\*do1\*oh1;

deo1\_outo1 = -1\*(op1-outo1);

douto1\_neto1 = exp(-neto1)/((1+exp(-neto1))^2);

deo1\_neto1 = deo1\_outo1\*douto1\_neto1;

deo1\_oh1 = deo1\_neto1\*w(1,5);

deo2\_outo2 = -1\*(op2-outo2);

douto2\_neto2 = exp(-neto2)/((1+exp(-neto2))^2);

deo2\_neto2 = deo2\_outo2\*douto2\_neto2;

deo2\_oh1 = deo2\_neto2\*w(1,7);

detotal\_oh1 = deo1\_oh1+deo2\_oh1;

doh1\_neth1 = exp(-oh1)/((1+exp(-oh1))^2);

detotal\_w1 = detotal\_oh1\*doh1\_neth1\*i1;

wn(1,1) = w(1,1)-0.5\*detotal\_w1;

wn(1,2) = w(1,2)-0.5\*detotal\_w1;

wn(1,3) = w(1,3)-0.5\*detotal\_w1;

wn(1,4) = w(1,4)-0.5\*detotal\_w1;

w

wn

**Output:**

w =

0.1500 0.2000 0.0500 0.1300 0.4000 0.4500 0.5000 0.5500

wn =

0.1498 0.1998 0.0498 0.1298 0.3589 0.4089 0.4589 0.5089