

Anexa 3

Lucrarea de laborator nr 5

```
clear all
close all
clc

P=[0 1];
T=[0 2];
net=newlind(P,T);
y=sim(net,P);
figure;
plot(P,T,'*r');
hold on;
plot(P,y);
lr=input('Rata de invatare: ');
%lr=maxlinlr(P);
fprintf('Valoarea ratei de invatare: %g\n\n',lr);

lin=newlin(minmax(P),1,[0],lr); % neuronul liniar creat este pregatit pt a fi
antrenat cu subrutina "learnwh" % (se verifica valorile cerute prin apelarea
"help learnwh") % eroarea este data de functia "mse" (eroarea
medie patratica)

% parametrii specifici antrenarii
iteratii=input('Nr de iteratii= '); % se va antrena cite o iteratie (epoca)
pt a putea % reprezenta grafic evolutia parametrilor

lin.trainParam.epochs=1;
lin.trainParam.goal=input('Eroarea medie patratica dorita= ');
lin.trainParam.show=inf; % eroarea nu este afisata in timpul antrenarii,
deoarece % antrenarea este efectuata epoca cu epoca
% eroarea va fi memorata la fiecare pas si
afisarea va fi facuta la sfirsit

% suprafata erorii
w_range = -2:0.4:6;
b_range = -4:0.4:4;
figure;
ES = errsurf(P,T,w_range,b_range,'purelin')/length(T); % valorile erorii
medii patraticice
plotes(w_range,b_range,ES);

% alegerea cu mouse-ul a parametrilor initiali ai retelei
subplot(1,2,2);
h = text(mean(get(gca,'xlim')),mean(get(gca,'ylim')), '* Alege cu mouse-ul
*');
set(h,'horizontal','center', 'fontweight','bold');
```

Rețele Neurale și Logică Fuzzy

```
[lin.IW{1,1},lin.b{1}] = ginput(1); % valorile alese cu mouse-ul devin
parametrii initiali ai rețelei
fprintf('\nValoarea ponderii initiale: %g\nValoarea deplasarii initiale:
%g\n',lin.IW{1,1},lin.b{1});
delete(h);

% antrenarea rețelei si reprezentarea la fiecare pas (epoca) a parametrilor
pe suprafata de eroare
eroare(1)= mse(T-sim(lin,P)); % eroarea medie patratica inainte de
inceperea antrenarii;
% vectorul "eroare" va contine evolutia
erorii
pondere(1)= lin.IW{1,1}; % vectorul care va contine evolutia
ponderilor
deplasare(1)= lin.b{1}; % vectorul care va contine evolutia
deplasarilor
h= plotep(lin.IW{1,1},lin.b{1},eroare(1)); % parametrii initiali
reprezentati pe suprafata de eroare
% (cu eroarea corespunzatoare)

for i= 1:iteratii,
    [lin,tr]= train(lin,P,T);
    h= plotep(lin.IW{1,1},lin.b{1},tr.perf(1),h); % evolutia parametrilor pe
suprafata de eroare
    eroare(i)= tr.perf(1); % eroarea pt noile valori ale parametrilor
    pondere(i+1)= lin.IW{1,1};
    deplasare(i+1)= lin.b{1};
    if eroare(i) <= lin.trainParam.goal
        ind = i;
        eroare(ind)
        break; % iesire pe conditia de atingere a pragului de eroare impus
    end
end

iteratii=i; % numarul de epoci cit a durat antrenarea
fprintf('\n\nAntrenarea a durat %g epoci\n',iteratii);
fprintf('\n\nValorile parametrilor finali:\nPonderea: %g\nDeplasarea:
%g\n\n',lin.IW{1,1},lin.b{1});

% evolutia erorii
disp('Apasati o tasta pt a vedea evolutia erorii functie de nr de epoci...');
pause;
figure;
plot((1:iteratii),eroare);
xlabel('iteratii');
ylabel('eroarea');
title('Eroarea medie patratica functie de numarul de iteratii');

% dinamica aproximarii
disp('Apasati o tasta pt a vedea aproximarea initiala...');
pause;
figure;
plot(P,T,'r+'); % punctele prin care se doreste aproximarea
xlabel('vectori prototip P');
ylabel('vectori tinta T');
hold on;
```

```
h= plot(P,pondere(1)*P+deplasare(1));  
axis manual;  
  
disp('Apasati o tasta pt a vedea evolutia dinamica a aproximarii...');  
pause;  
for i=1:iteratii  
    pause(0.1);  
    delete(h);  
    h= plot(P,pondere(i+1)*P+deplasare(i+1));  
end  
  
axis auto;  
hold off;
```