Universitatea Tehnică “Gheorghe Asachi”, Iași
Facultatea de Automatică și Calculatoare
Specializarea Calculatoare și Tehnologia Informației
Grupa 1411B

Disciplina Managementul Proiectelor Software

Etapa 4
Implementare

Studenti,
Anițoaei Teodor
Chihalău Adrian
Mîrț Alexandru
Stanciu Ioan
Sîrghi Simona
Stoian Alin-Bogdan
Zbereanu Alexandru

Cuprins

I. Enunțarea problemei

Scopul acestei etape este implementarea propriu-zisă a unei rețele neuronale de tip RBF folosind Matlab 2021b.

Rețeaua neuronală va aproxima funcția neliniară

$$\varphi(u) = sin^2(u)$$

Iar setul de date de antrenare este format din

$$\boldsymbol{u(i) = \frac{2(i-1)\pi}{N}}$$

$$\boldsymbol{d(i) = sin^2\big(u(i)\big), i = 1, N}$$

În acest sens se vor considera următoarele situații:

SPREAD = 1, 10, 0.1
Număr exemple în setul de antrenare = 10, 100
GOAL = 0.0001, 0

Se vor utiliza funcțiile Matlab newrb si newrbe. Ambele metode vor fi prezentate în acest document.

II. Codul problemei

```
SPREAD = [1 10 0.1];
EXAMPLES = [10 100];
GOAL = [0.0001 0];

% algoritmul in cazul functiei newrb
for i = 0:length(SPREAD)-1
       for j = 0:length(EXAMPLES)-1
       for k = 0:length(GOAL)-1
              spread = SPREAD(i + 1);    % spread
              examples = EXAMPLES(j + 1);  % numarul de exemple in setul de
antrenare
              goal = GOAL(k + 1);       % eroarea maxima admisa (goal)

              fprintf("Cazul %g: SPREAD = %g; EXAMPLES = %g; GOAL = %g\n", i*4 + j*2
+ k + 1, spread, examples, goal);

              dataset = linspace(0, 2*pi, examples);   % setul de date de marime
'examples'
              values = sin(dataset) .* sin(dataset);   % valorile functiei pentru
setul de date

              net = newrb(dataset, values, goal, spread, 50, 2);   % crearea retelei
de tip RBF
              fprintf("Parametrii retelei: \n");
              display(net.IW);
              display(net.LW);
              display(net.b);

              test_set = 0:0.001:2*pi;         % setul de date de testare
              test_values = sin(test_set) .* sin(test_set); % valorile functiei
pentru setul de testare

              sim_result = sim(net, test_set);              % rezultatele simularii
              % afisarea rezultatelor
              figure(1);
              subplot(3, 4, i*4 + j*2 + k + 1);
              plot(test_set, test_values);
              hold on;
              plot(test_set, sim_result);
              title({['SPREAD = ', num2str(spread),' EX = ', num2str(examples)]; ['
GOAL = ', num2str(goal), ' ']});
              hold on;
       end
       end
end
```

În cazul utilizării funcției newrbe, linia

```
net = newrb(dataset, values, goal, spread, 50, 2);
```

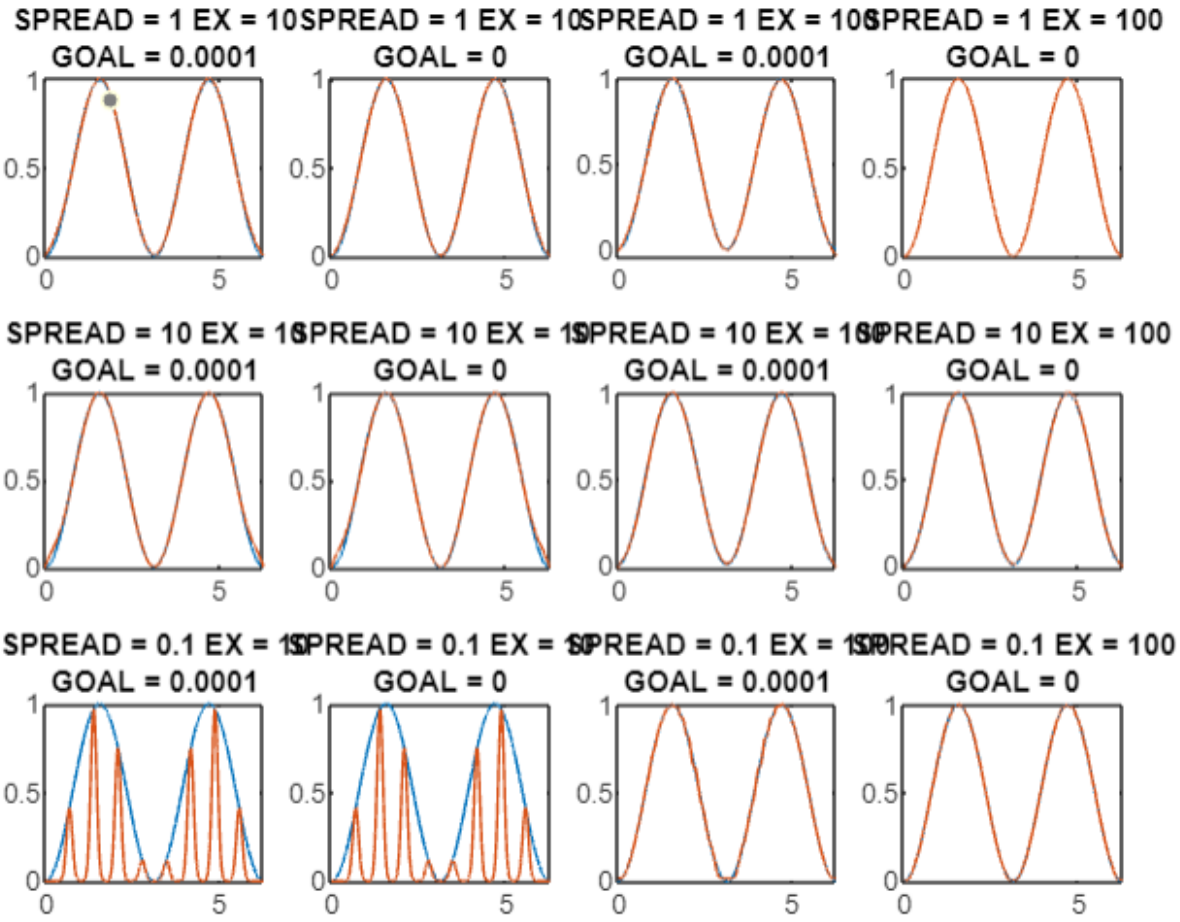
va fi înlocuită cu

```
 net = newrbe(dataset, values, spread);
```

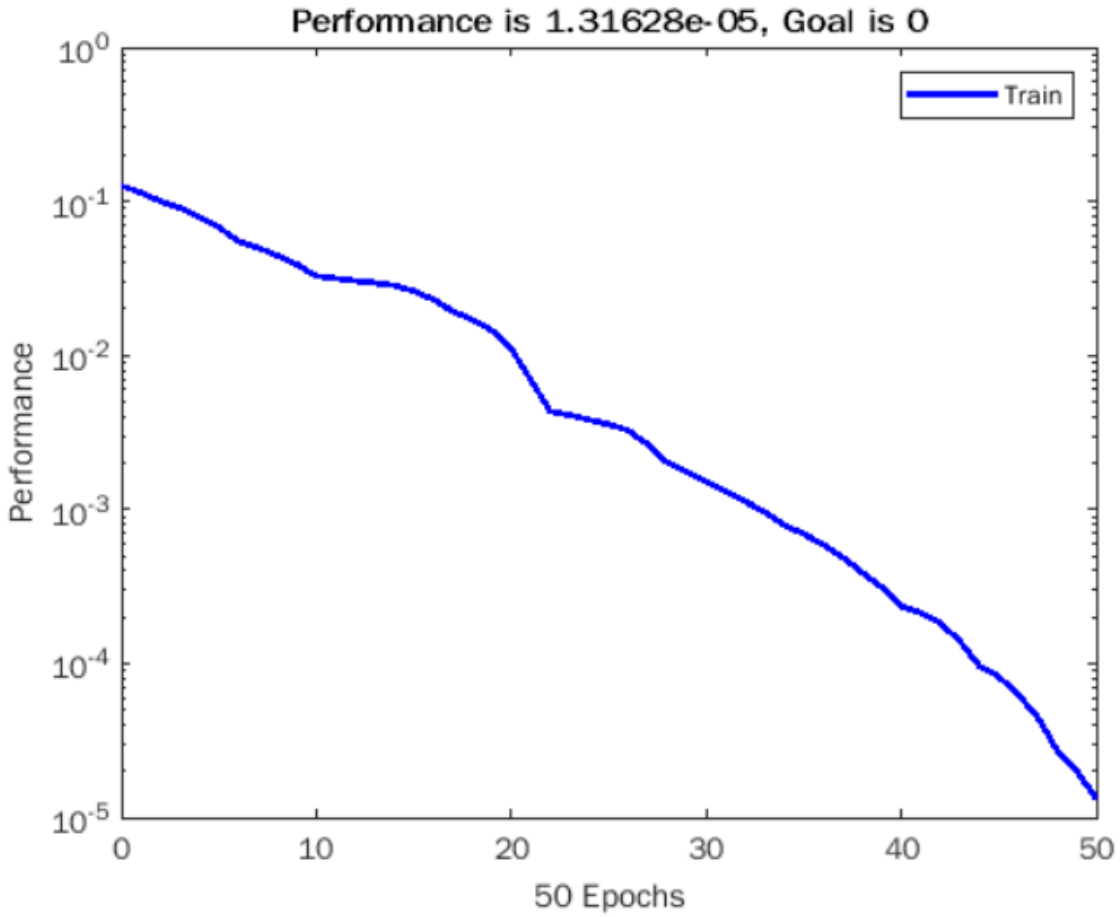
și, desigur, GOAL-ul devine irelevant; așadar, în cazul funcției newrb, vom avea 12 combinații, iar pentru newrbe, doar 6.

III. Capturi ale rezultatelor

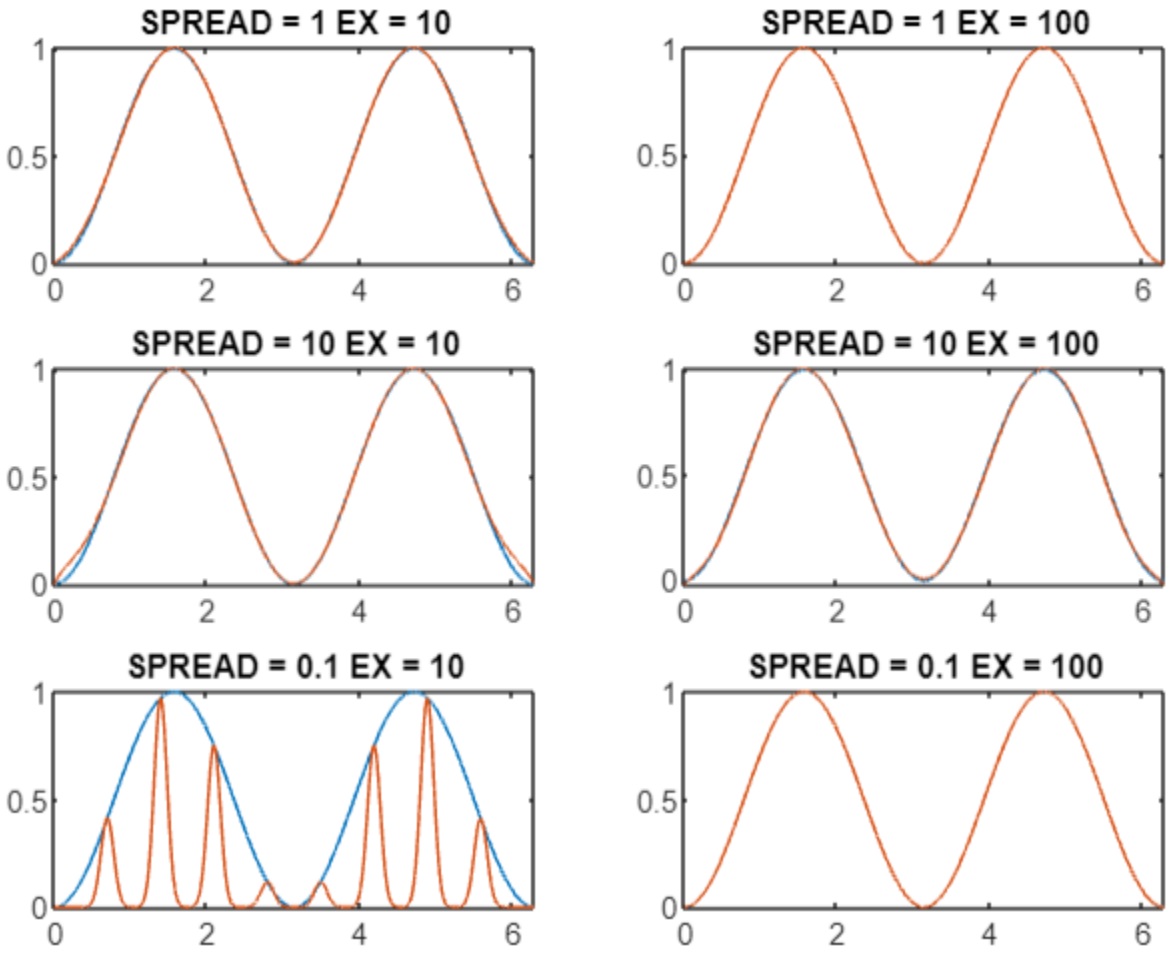
a. Folosind functia newrb()



Exemplu de rulare:



b. Folosind funcția newrbe()



IV. Concluzii

Se observă că o valoare mai mare pentru SPREAD ajută la acuratețea prezicerilor rețelei neurale atât în cazul rețelei newrb cât și pentru newrbe. Totuși, chiar și cu o valoare mai mică pentru SPREAD (0.1) putem obține rezultate bune dacă numărul de exemple în setul de antrenament este relativ mare (100). Putem observa în ambele cazuri că dacă numărul de exemple este mic (10), si SPREAD-ul mic (0.1), rețeaua nu poate prezice deloc cum arată funcția.

În cazul newrb, un număr adecvat de neuroni variază între 10 și 50, depinzând de SPREAD, GOAL și numărul de exemple.