



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

Disciplina Managementul Proiectelor Software

Etapă 6

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

Etapa 1. Centralizarea rezultatelor testelor

1.1 Rezultatele testelor (organizare a timpului și motivare personală)

	Test	
Name	Time management	Motivation to lead
Anițoei Teodor	42	40
Chihalău Adrian	37	49
Mîrț Alexandru	50	58
Stanciu Ioan	60	62
Sîrghi Maria-Simona	50	49
Stoian Alin-Bogdan	58	59
Zboreanu Alexandru	49	45

Interpretarea scorurilor

Motivation to lead

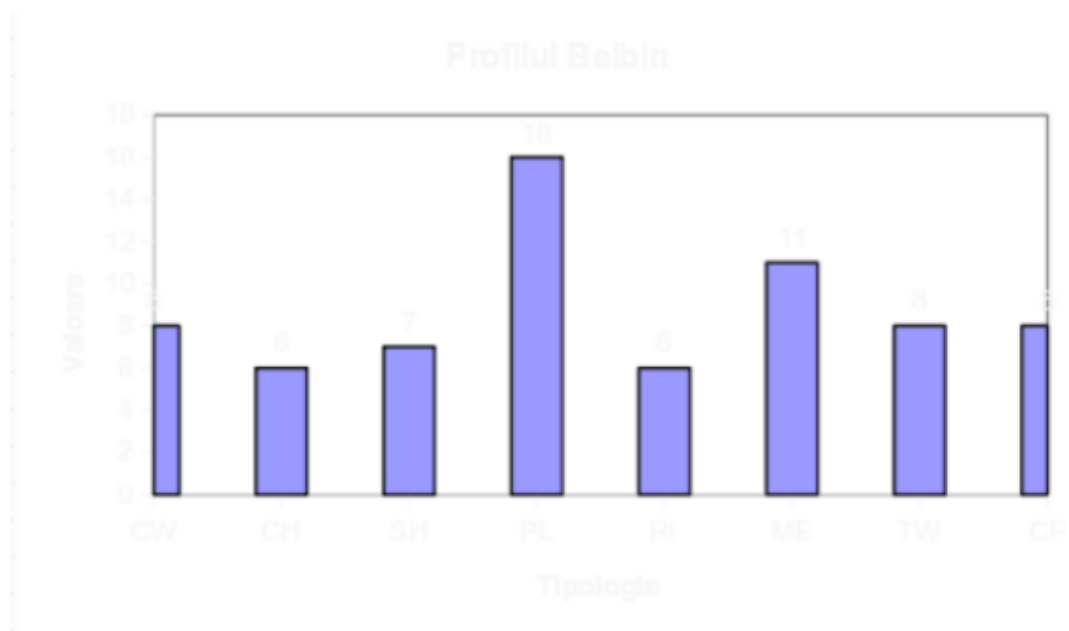
Score	Comment
14-27	This implies a low motivation to lead.
28-55	This implies some uncertainty about your motivation to lead.
56-70	This implies a strong motivation to lead.

Time management

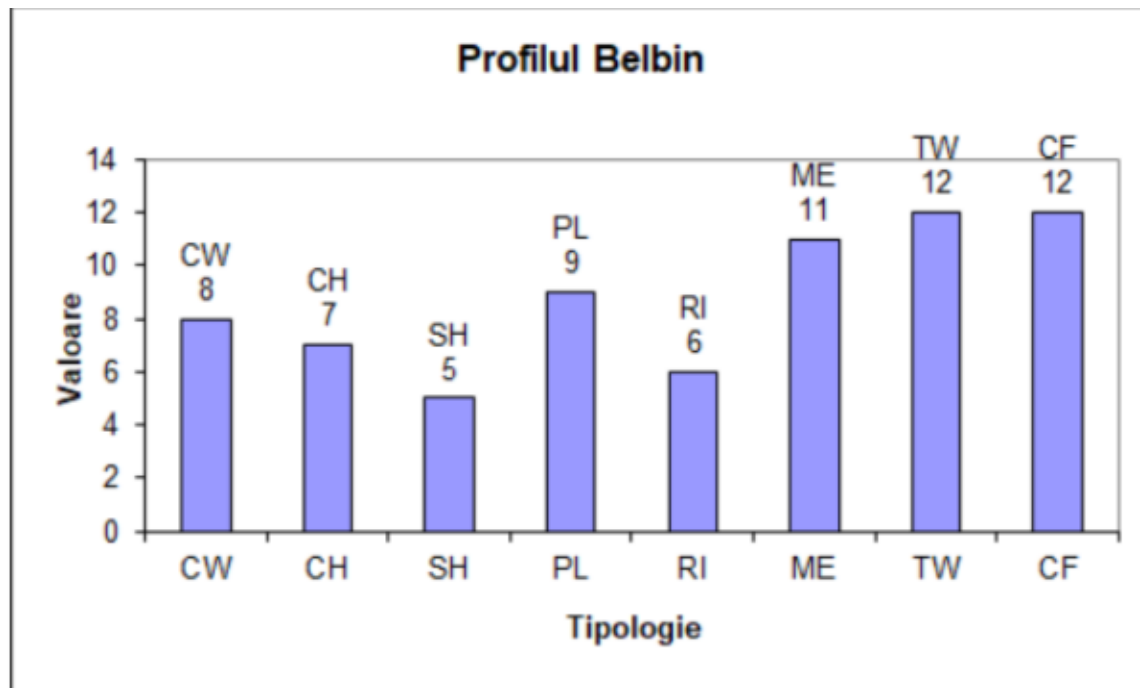
Score	Comment
15-30	Ouch. The good news is that you've got a great opportunity to improve your effectiveness at work, and your long term success! However, to realize this, you've got to fundamentally improve your time management skills. (Read below to start.)
31-45	You're good at some things, but there's room for improvement elsewhere. Focus on the serious issues below , and you'll most likely find that work becomes much less stressful.
46-75	You're managing your time very effectively! Still, check the sections below to see if there's anything you can tweak to make this even better.

1.2 Rezultate test Belbin

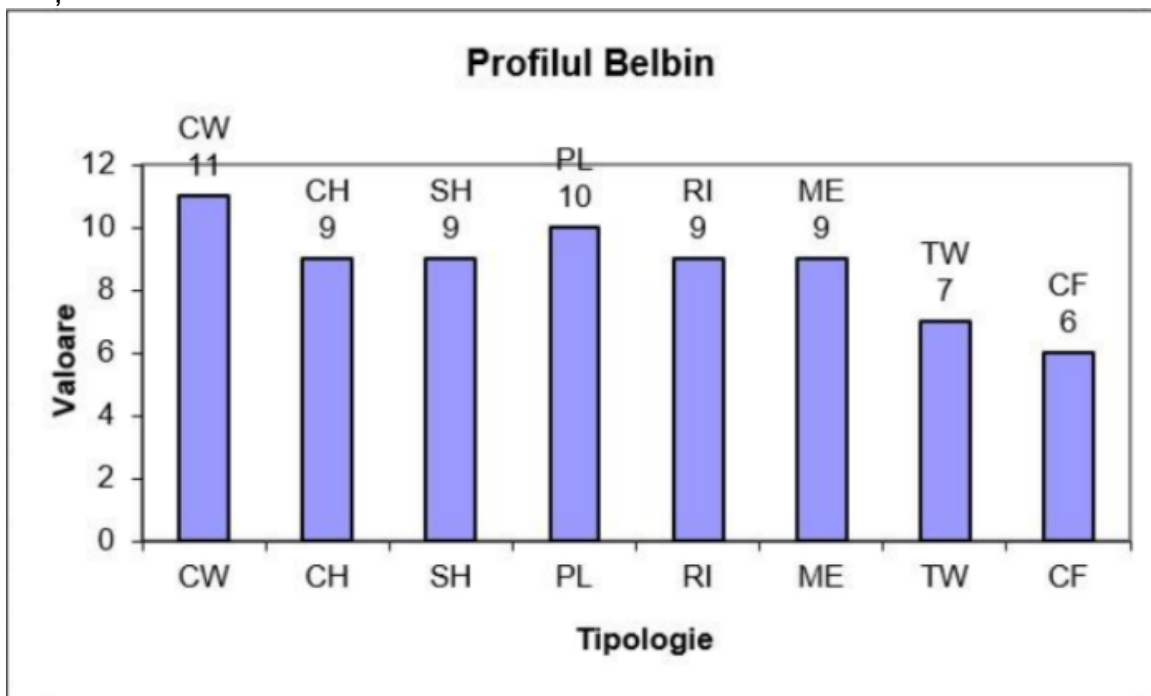
Anițoaei Teodor



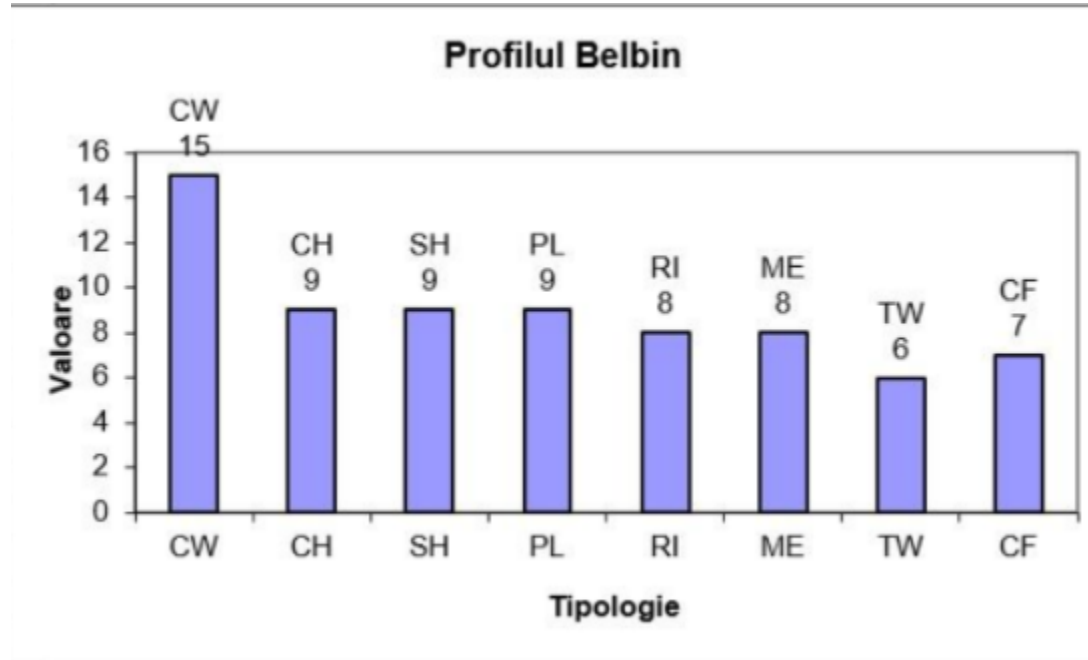
Chihalău Adrian



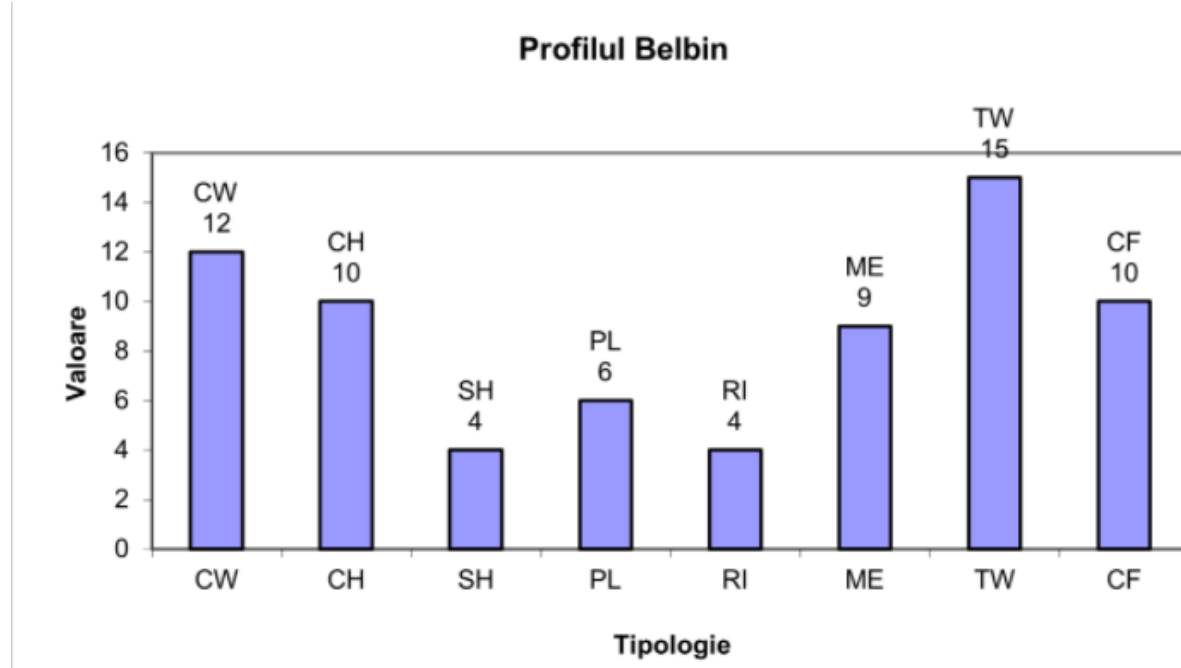
Mîrț Alexandru



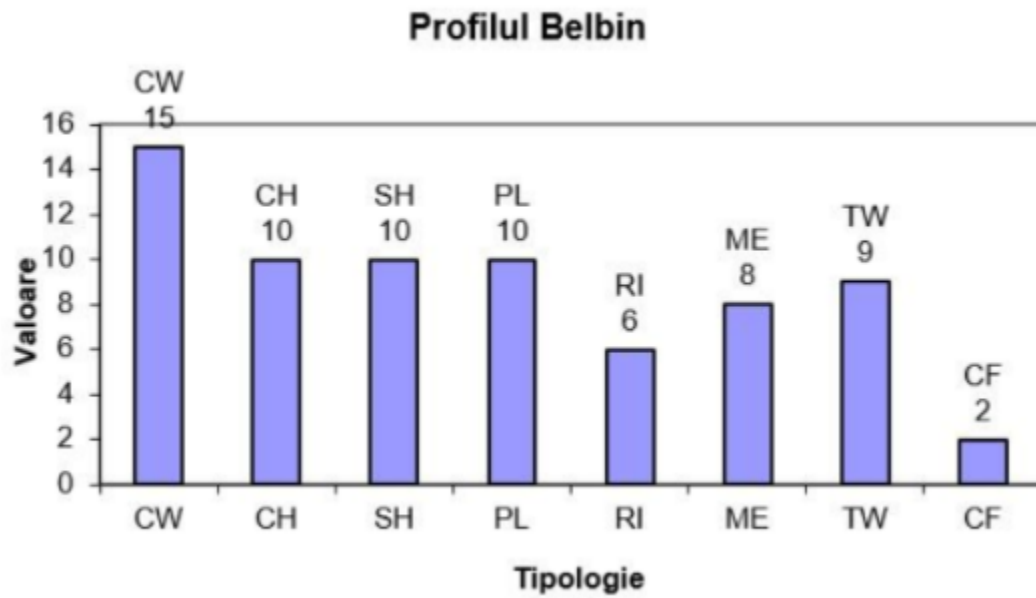
Stanciu Ioan



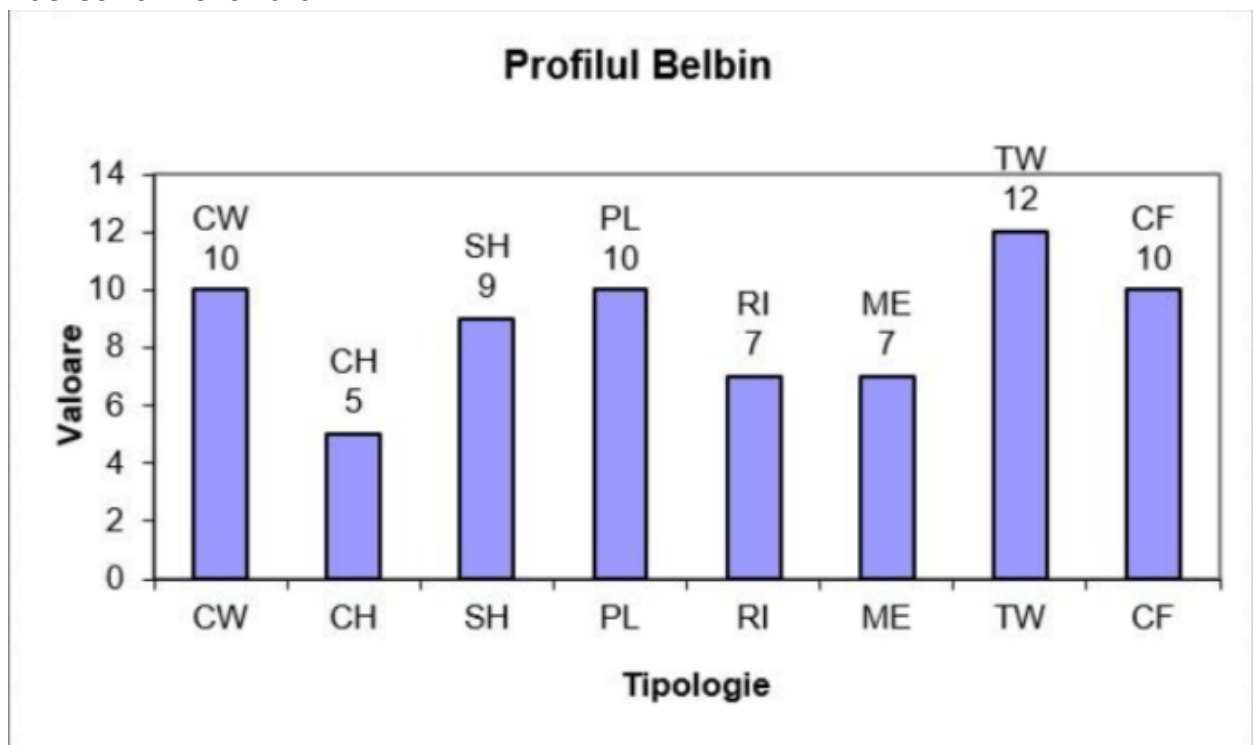
Sîrghi Maria-Simona



Stoian Alin-Bogdan



Zboreanu Alexandru



Etapa 2 – SRS

1. Introducere

1.1. Scopul documentului

Prin documentul de față ne propunem să realizăm o implementare și proiectare a unei RNA (Rețea Neuronală Artificială) de tip RBF (Radial Basis Function) prin intermediul unor metode eficiente de proiectare a acesteia.

Vor fi afișate diagramele folosite în scopul proiectării cât și cele rezultate în obținerea rezultatelor, cât și tot ce ține de soluțiile găsite.

1.2. Conținutul documentului

Documentul prezintă la modul general proiectul, elementele utilizate în proiectarea acestuia și metodele de implementare folosite, cât și detalii față de avantajele și dezavantajele implementării noastre, precum și scopul realizării aplicației.

Sunt descrise, așadar, cititorilor misiunea, contextul și cerințele funcționale ale acestui proiect.

2. Descrierea proiectului

2.1. Descrierea generală a produsului

Acest proiect urmărește implementarea unei **Rețele Neuronale Artificiale (RNA)** pe baza **Funcțiilor Radiale** (eng. **Radial Basis Function - RBF**), având stratul de intrare cu un singur neuron, stratul ascuns cu k neuroni (unde k reprezintă un număr variabil), și stratul de ieșire cu un singur neuron.

Această rețea va aproxima în mod optim funcția neliniară:

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

pentru care vom folosi următorul **set de date de antrenare**:

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

$$d(i) = \sin^2(u(i)), i = 1, N$$

În cadrul analizei, se vor lua în considerație situațiile:

- SPREAD = 1, 10, 0.1;
- Număr de exemple în setul de intrare: 10, 100;
- GOAL = 0.0001; 0.

Pentru fiecare combinație de parametri precizați mai sus, produsul va determina:

- Valorile inițiale ale parametrilor și valorile finale ale acestora (rezultate de antrenare);
- Număr neuroni ascunși k necesari;

- Reprezentarea grafică a aproximării realizate de rețea pe baza acestor valori pe setul de antrenare și pe setul de testare:

$$UTEST = 0 : 0.001 : 2 * \pi .$$

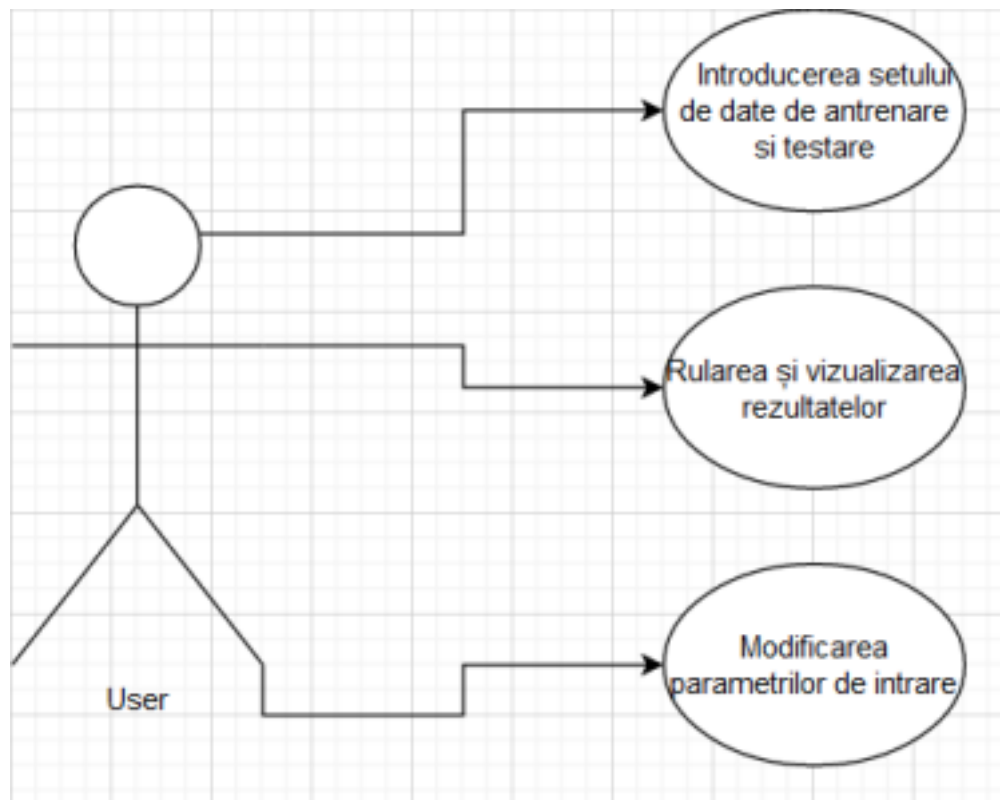
- Reprezentarea grafică a ieșirilor neuronilor ascunși (pentru tiparele de intrare din setul de antrenare).

2.2. Misiunea proiectului

Simularea în întregime a unui tur de dezvoltare a unui produs software, incluzând etapele de analiză, proiectare și implementare, cu scopul final de a livra și lansa un produs de calitate în domeniul software, utilizând algoritmi de inteligență artificială.

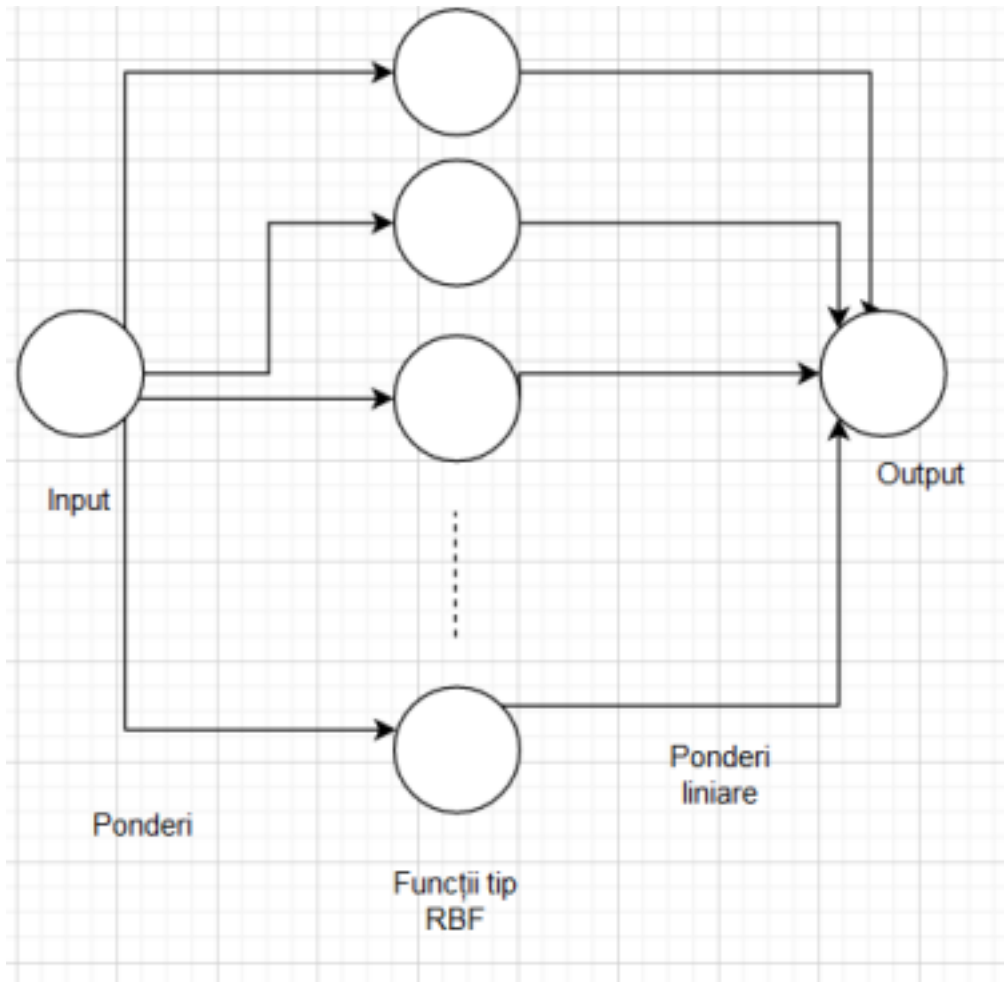
3. Cerințe funcționale

3.1. Diagrama de use-case



5

3.2. Diagrama de sistem



6

4. Cerințe nefuncționale

4.1. Cerințe de interfață

Aplicația oferă utilizatorului graful care aproximează funcția cu ajutorul careia s-a construit problema pe baza unui set de date de intrare care vor fi folosite pentru antrenare.

4.2 Cerințe de performanță

Programul trebuie să se execute în parametrii solicitați de utilizator. Putem cuantifica performanța sistemului prin măsurarea timpului în care se realizează procesarea datelor. Acest timp este variabil direct proporțional cu setul de date de

intrare folosit, modul în care se realizează antrenarea rețelei, precizia cu care se fac aproximări și de hardware-ul care executa codul.

4.3 Cerințe de fiabilitate

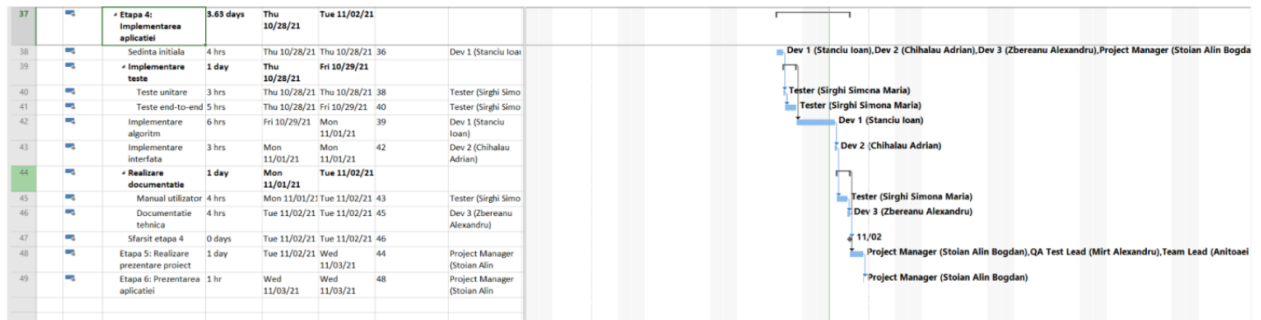
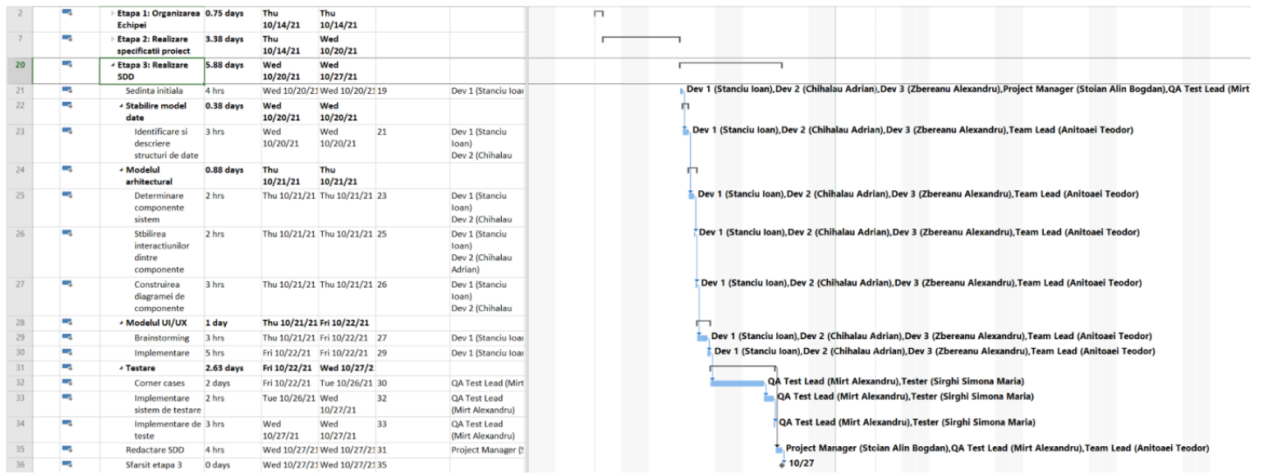
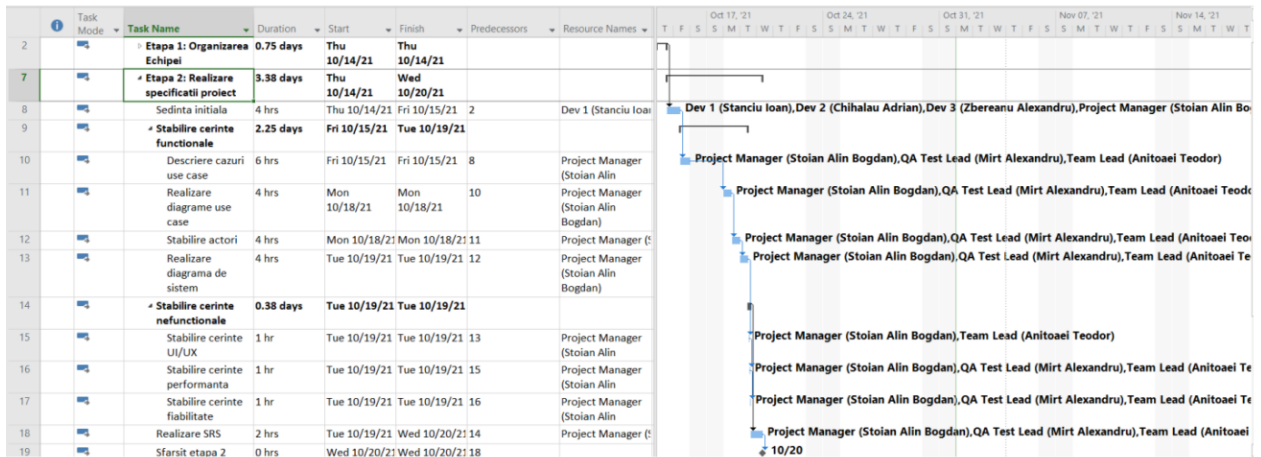
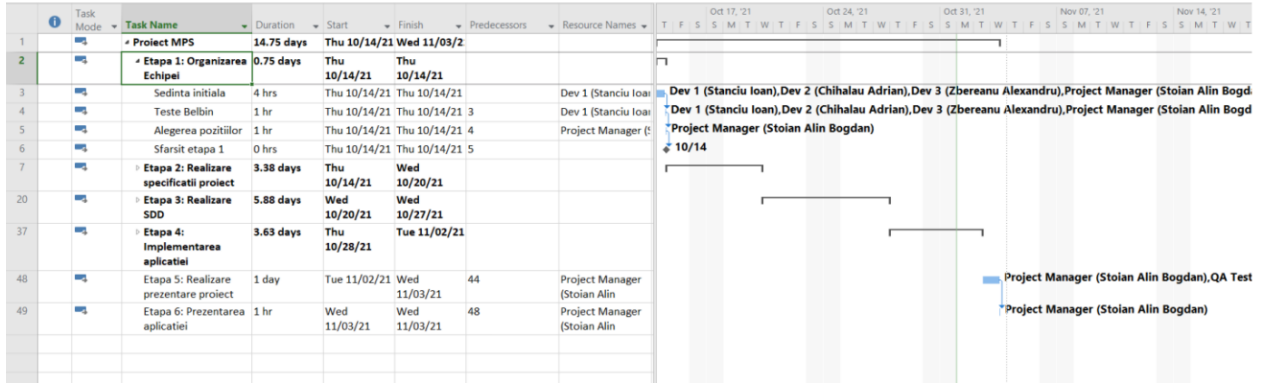
Aplicația trebuie sa aproximeze funcția și să ofere utilizatorului rezultate cât mai precise pentru a facilita lucrul cu acestea.

5. Planificarea proiectului

	Resource Name	Type	Material	Initials	Group	Max.	Std. Rate	Ovt. Rate	Cost/Use	Accrue	Base	Code
	Project Manager (Stoian Alin Bogdan)	Work		P		100%	\$40.00/hr	\$80.00/hr	\$0.00	Prorated	Standard	
	Team Lead (Anitoaei Teodor)	Work		T		100%	\$35.00/hr	\$70.00/hr	\$0.00	Prorated	Standard	
	QA Test Lead (Mirt Alexandru)	Work		Q		100%	\$30.00/hr	\$60.00/hr	\$0.00	Prorated	Standard	
	Dev 1 (Stanciu Ioan)	Work		D		100%	\$25.00/hr	\$50.00/hr	\$0.00	Prorated	Standard	
	Dev 2 (Chihalau Adrian)	Work		D		100%	\$25.00/hr	\$50.00/hr	\$0.00	Prorated	Standard	
	Dev 3 (Zbereanu Alexandru)	Work		D		100%	\$25.00/hr	\$50.00/hr	\$0.00	Prorated	Standard	
	Tester (Sirghi Simona Maria)	Work		T		100%	\$15.00/hr	\$30.00/hr	\$0.00	Prorated	Standard	

	Resource Name	Work	Add New Column	Details	M	T	W	T	F	S	Oct 31, '21	S	M	T	W	T
	Unassigned	0 hrs		Work												
1	Project Manager (Stoian Alin Bogdan)	54 hrs		Work			4h	4h						3h	6h	
2	Team Lead (Anitoaei Teodor)	70 hrs		Work			4h	4h						3h	5h	
3	QA Test Lead (Mirt Alexandru)	72 hrs		Work	8h	8h	8h	4h						3h	5h	
4	Dev 1 (Stanciu Ioan)	41 hrs		Work				4h	4h				2h			
5	Dev 2 (Chihalau Adrian)	38 hrs		Work				4h					3h			
6	Dev 3 (Zbereanu Alexandru)	39 hrs		Work				4h						4h		
7	Tester (Sirghi Simona Maria)	50 hrs		Work	8h	8h	4h	8h	4h				3h	1h		
				Work												
				Work												

Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names	31, '21	Nov 7, '21	Nov 14, '21	Nov 21, '21	Nov 28, '21	Dec 5
	Project MPS	14.75 days	Mon 11/1/21	Fri 11/19/21								
	Etapă 1: Organizarea Echipei	0.75 days	Mon 11/1/21	Mon 11/1/21								
	Etapă 2: Realizare specificatii proiect	3.38 days	Mon 11/1/21	Fri 11/5/21								
	Etapă 3: Realizare SDD	5.88 days	Fri 11/5/21	Fri 11/12/21								
	Etapă 4: Implementarea aplicatiei	3.63 days	Mon 11/15/21	Thu 11/18/21								
	Etapă 5: Realizare prezentare proiect	1 day	Thu 11/18/21	Fri 11/19/21	44	Project Manager (Stoian Alin Bogdan)						
	Etapă 6: Prezentarea aplicatiei	1 hr	Fri 11/19/21	Fri 11/19/21	48	Project Manager (Stoian Alin Bogdan)						



Etapa 3 – SDD

I. Introducere

A. Scopul documentului

Documentul de față are drept scop o descriere a proiectării și implementării unei rețele neurale (RNA) de tip RBF (Radial Basis Function); avem un strat de intrare cu un singur neuron, strat ascuns cu k neuroni (k fiind un număr variabil) și strat de ieșire cu un singur neuron.

Documentul are scopul de a ghida echipa de dezvoltare și implementare în construirea unei soluții cât mai eficiente.

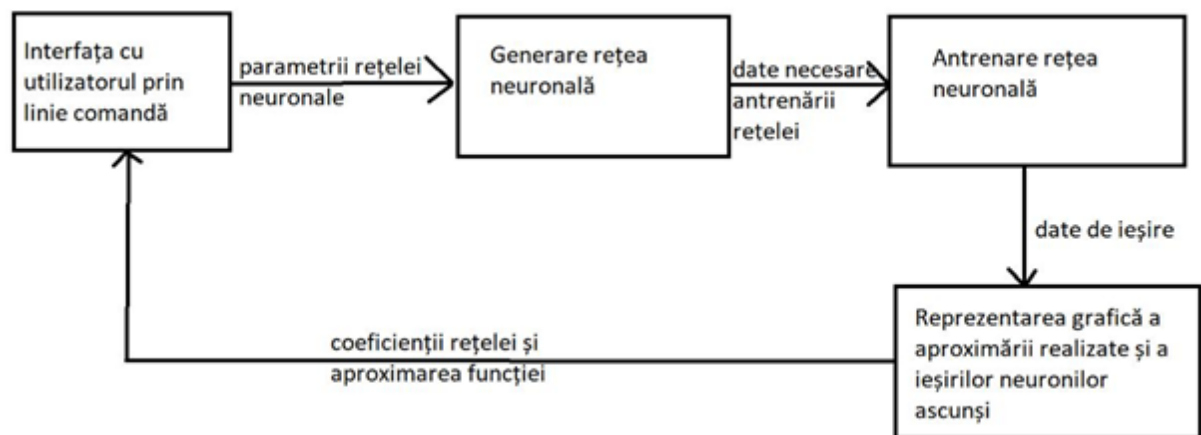
B. Conținutul documentului

1. Modelul arhitectural: acesta descrie arhitectura sistemului, precum și tot ce ține de componentele arhitecturii.

2. Documentarea planului de proiectare: descrie prețarea cunoștințelor teoretice asupra cerințelor din descrierea problemei.

II. Modelul arhitectural

A. Diagrama de arhitectură



B. Descrierea componentelor

1. Interfața cu utilizatorul

Această componentă va oferi posibilitatea utilizatorului de a selecta parametrii de antrenare: SPREAD între 10, 1 sau 0.1; numărul de exemple în setul de antrenare: 10

sau 100; GOAL: 0.00001 sau 0.

2. Componenta de generare a rețelei neuronale

Aceasta va prelua datele introduse de utilizator și pe baza acestora va genera rețeaua neuronală;

3.Componenta de antrenare a rețelei neuronale

Aceasta va antrena efectiv rețeaua neuronală din cadrul programului.

4.Componenta de reprezentare grafică

Aceasta va realiza reprezentarea grafică a aproximării realizate și a ieșirilor neuronilor ascunși.

III. Documentarea planului de proiectare

Presupunem problema de aproximare a funcției neliniare:

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

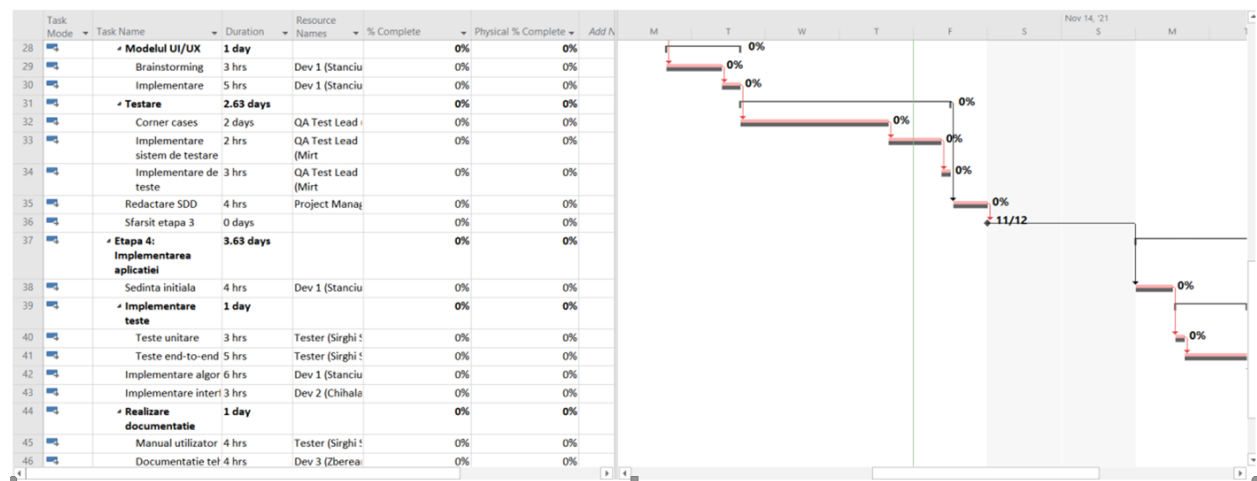
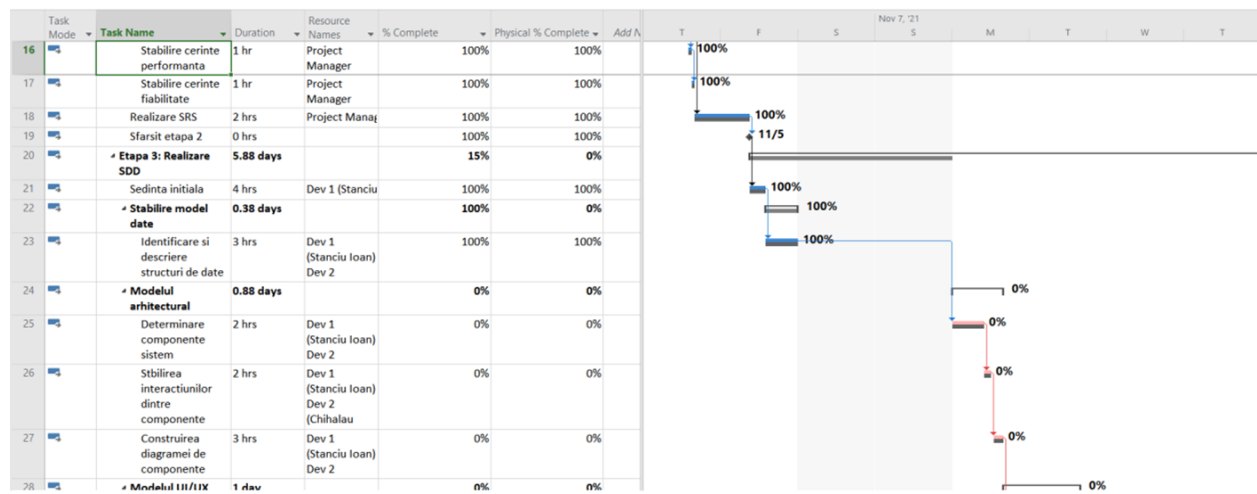
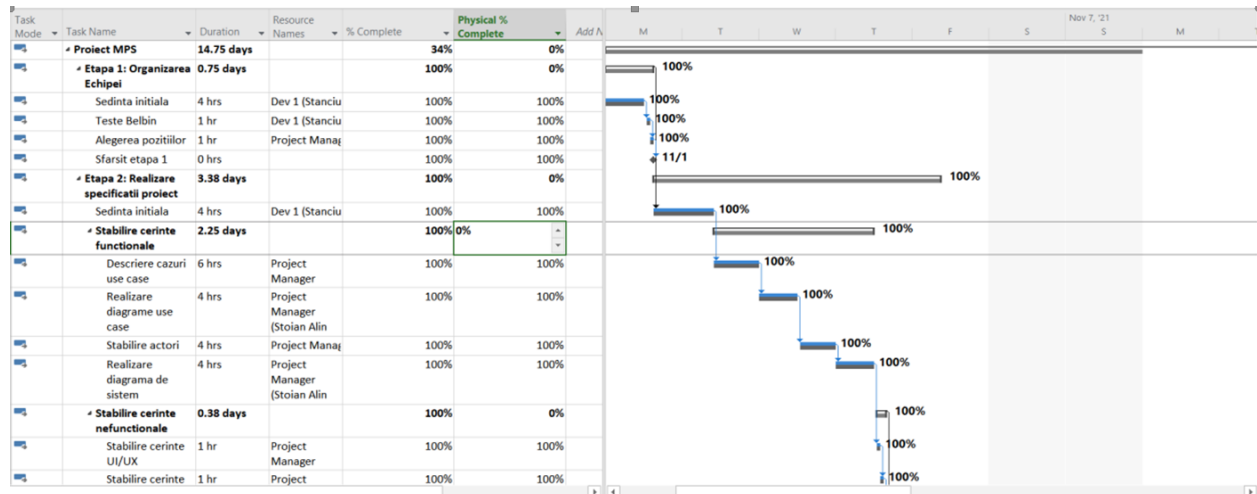
Pentru care vom folosi setul de date de antrenare:

$$u(i) = 2 * (i - 1) * \frac{\pi}{N};$$
$$d(i) = \sin^2(u(i))$$

Unde $i = 1, N$ și N poate avea valorile 10 sau 100, în funcție de datele de intrare.

Rețeaua neuronală va conține un strat de intrare cu un singur neuron, un strat de neuroni ascunși cu k neuroni și un strat de ieșire cu un singur neuron.

IV. Planificarea activităților și progresul față de etapa precedentă:



Etapa 4 – Implementare

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

$$u(i) = \frac{2(i-1)\pi}{N}$$
$$d(i) = \sin^2(u(i)), i = 1, N$$

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 newrb, linia

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

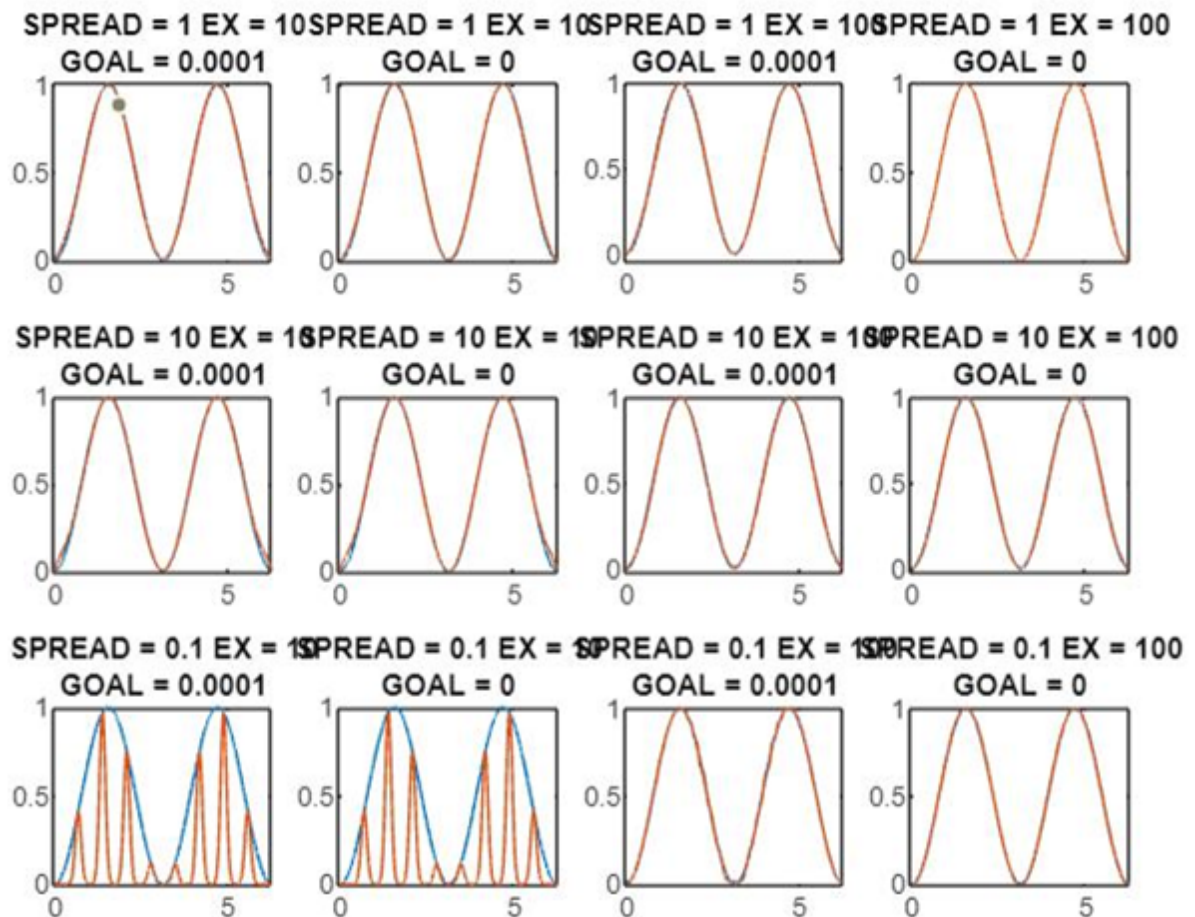
va fi înlocuită cu

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

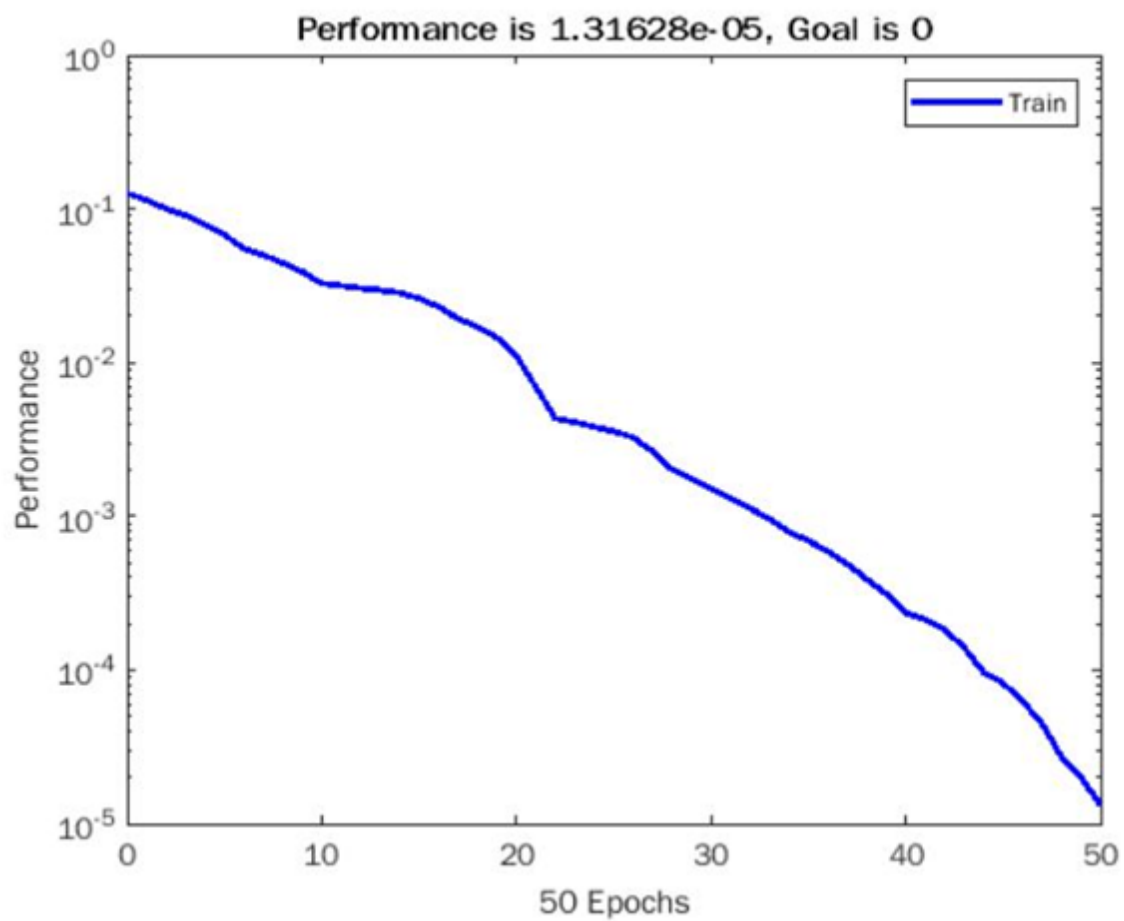
și, desigur, GOAL-ul devine irelevant; așadar, în cazul funcției newrb, vom avea 12 combinații, iar pentru newrbe, doar 6.

III. Capturi ale rezultatelor

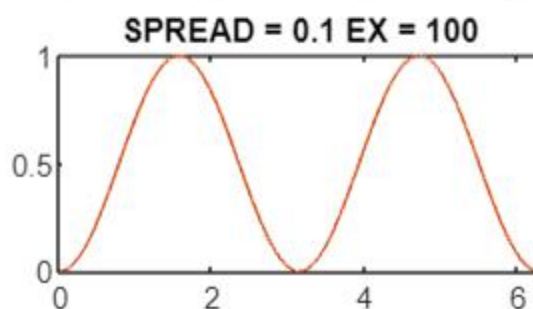
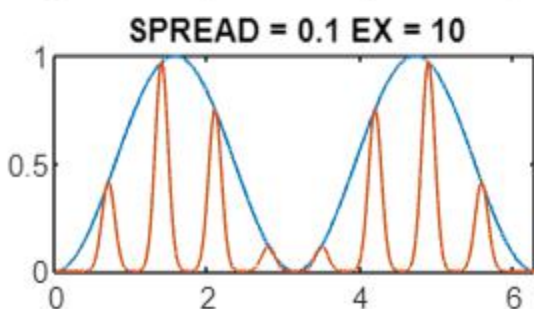
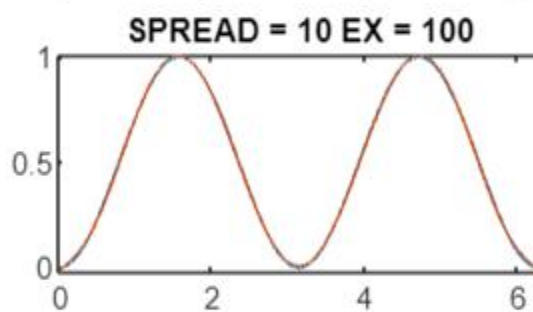
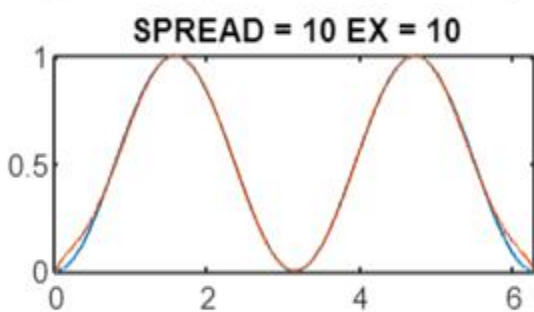
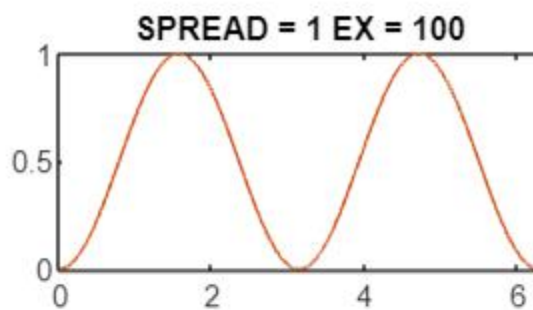
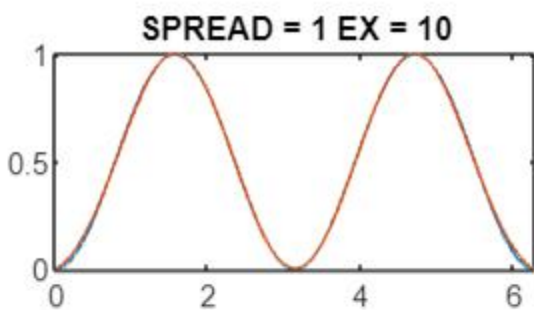
a. Folosind funcția newrb()



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), și 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.

Etapa 5 – Testare

Testarea aplicației

Aplicatia noastra aproximează cu ajutorul unei rețele neuronale de tip RBF funcția:folosind diferite valori pentru parametrii rețelei. Ne propunem sa testam efectele parametrilor asupra rezultatului prezis de rețea, cu alte cuvinte, dorim sa testam care sunt parametri optimi în cazul nostru pentru a obține o estimare cat mai apropiata de adevăr. În acest scop, vom folosi funcția internă matlab *immse*, care calculeaza eroarea medie pătratică între doua seturi de date. Un set va fi cel prezis de rețea, iar celalalt va conține valorile reale ale funcției. Prin aceste teste ne asigurăm de asemenea ca rețeaua poate aproxima cu un grad mare de încredere funcția.

Vom folosi un prag de $1e-4$ pentru a stabili dacă parametri rețelei aduc rezultate satisfăcătoare sau nu. Astfel, dacă eroarea dintre rezultatul rețelei și valorile reale este mai mare de $1e-4$, vom decide ca acei parametri utilizati pentru obținerea rezultatului nu sunt optimi pentru problema noastra.

Codul pentru calcularea erorii și afișarea rezultatelor:

```
error = immse(test_values, sim_result);  
fprintf("***** NEWRB %g %g %g %g %g\n", goal, examples, spread, error,  
                                             error < 1.e-4);
```

Tabelul cu parametri rețelei și rezultatele testelor:

METHOD	GOAL	EXAMPLES	SPREAD	ERROR	PASS
NEWRB	0.0001	10	1	0.000153379	FAIL
NEWRB	0	10	1	7.24E-05	PASS
NEWRB	0.0001	100	1	7.20E-05	PASS
NEWRB	0	100	1	1.89E-18	PASS
NEWRB	0.0001	10	10	0.000344117	FAIL
NEWRB	0	10	10	0.000364051	FAIL
NEWRB	0.0001	100	10	3.71E-05	PASS
NEWRB	0	100	10	3.74E-05	PASS

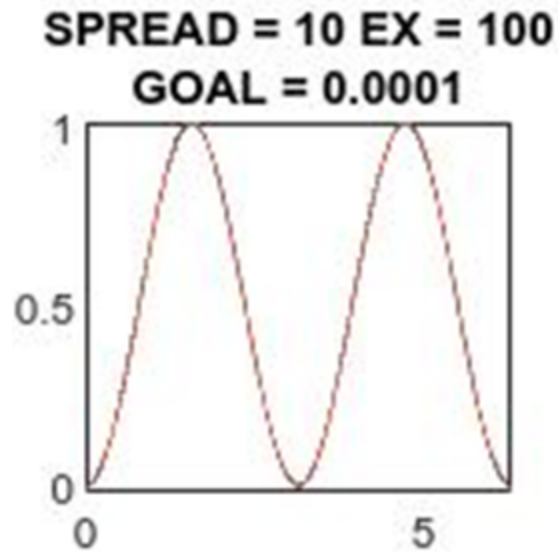
NEWRB	0.0001	10	0.1	0.228213	FAIL
NEWRB	0	10	0.1	0.228213	FAIL
NEWRB	0.0001	100	0.1	9.49E-05	PASS
NEWRB	0	100	0.1	1.29E-05	PASS
NEWRBE	N/A	10	1	7.24E-05	PASS
NEWRBE	N/A	100	1	2.11E-18	PASS
NEWRBE	N/A	10	10	0.000364058	FAIL
NEWRBE	N/A	100	10	3.72E-05	PASS
NEWRBE	N/A	10	0.1	0.228213	FAIL
NEWRBE	N/A	100	0.1	2.23E-11	PASS

Corelatia dintre parametri rețelei (EXAMPLES și SPREAD) și rata de trecere a testelor, de unde observăm că există o corelație puternică între numărul de exemple și rata de reușită:

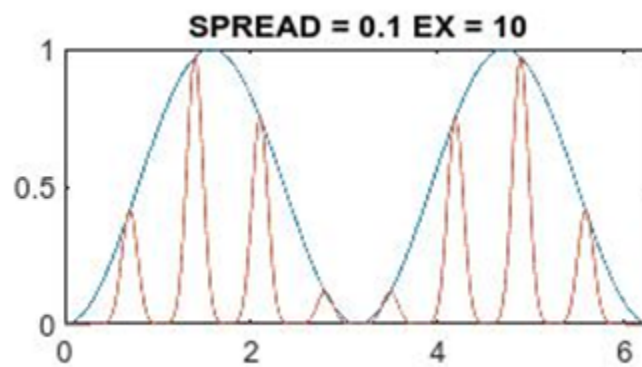
	PASS
EXAMPLES	0.8
SPREAD	-0.14

Din aceste tabel putem deduce că este esențial să alegem un set de date suficient de mare pentru a aproxima cu un grad mare de acuratețe funcția.

Exemplu de test reușit (PASS, în cazul rețelei NEWRB):



Exemplu de test eșuat (FAIL, în cazul rețelei NEWRBE):



Progresul activităților

Activitățile corespundente acestei etape:

37	■	• Etapa 4: Implementarea aplicației	3.63 days	Mon 11/15/21	Thu 11/18/21			
38	■	Sedinta initiala	4 hrs	Mon 11/15/21	Mon 11/15/21	36	Dev 1 (Stanciu Ioan),Dev 2 (Chihalau Adrian),Dev 3 (Zbereanu Alexu	Dev 1 (Stanciu Ioan),Dev 2 (Chihalau Adrian),Dev 3 (Zbereanu Alexu
39	■	• Implementare teste	1 day	Mon 11/15/21	Tue 11/16/21			Tester (Sirghi Simona Maria)
40	■	Teste unitare	3 hrs	Mon 11/15/21	Mon 11/15/21	38	Tester (Sirghi Simona Maria)	Tester (Sirghi Simona Maria)
41	■	Teste end-to-end	5 hrs	Mon 11/15/21	Tue 11/16/21	40	Tester (Sirghi Simona Maria)	Dev 1 (Stanciu Ioan)
42	■	Implementare algorithm	6 hrs	Tue 11/16/21	Wed 11/17/21	39	Dev 1 (Stanciu Ioan)	Dev 2 (Chihalau Adrian)
43	■	Implementare interfata	3 hrs	Wed 11/17/21	Wed 11/17/21	42	Dev 2 (Chihalau Adrian)	Tester (Sirghi Simona Maria)
44	■	• Realizare documentatie	1 day	Wed 11/17/21	Thu 11/18/21			Dev 3 (Zbereanu Alexandru)
45	■	Manual utilizator	4 hrs	Wed 11/17/21	Thu 11/18/21	43	Tester (Sirghi Simona Maria)	
46	■	Documentatie tehnica	4 hrs	Thu 11/18/21	Thu 11/18/21	45	Dev 3 (Zbereanu Alexandru)	