

ANTRENAREA SUPERVIZATĂ A REȚELOR FEEDFORWARD MULTISTRAT ÎN PROBLEME DE CLASIFICARE

1. Considerații generale, motivație și obiectiv

Rețelele neuronale feed-forward multistrat cu noduri de intrare sigmoide sunt utilizate în probleme de clasificare a datelor. Alegerea funcțiilor de activare are o influență majoră asupra performanțelor rețelei. Performanța rețelei MLP este calculată pe baza procentajului elementelor clasificate corect. Pașii urmăți pentru crearea modelului conexiunilor folosite în rezolvarea problemei sunt : definirea unui set de trăsături folosite ca date de intrare precum și definirea structurii rețelei, a ratei de învățare și a metodelor de validare. Reperul de bază în utilizarea rețelelor neurale în recunoașterea de șabloane îl constituie setul de trăsături care trebuie să reprezinte fără ambiguitate toate șabloanele permise ca date de intrare pentru categoriile respective.

2. Cunoștințe prealabile necesare

- Capitolul "Antrenarea supervizată a rețelelor neuronale", secțiunea "Rețele feed-forward multistrat" din cursul "Aplicații ale rețelelor neurale în automatică".
- Experiență de programare în MATLAB.

3. Breviar teoretic

Utilizarea rețelelor neuronale feed-forward multistrat în probleme de clasificare se bazează pe antrenarea rețelelor prin algoritmul Backpropagation. Denumirea algoritmului provine din faptul că informațiile sunt procesate în sens invers de la ieșirea rețelei către intrare.

3.1. Algoritmul de clasificare

Se consideră imaginea:

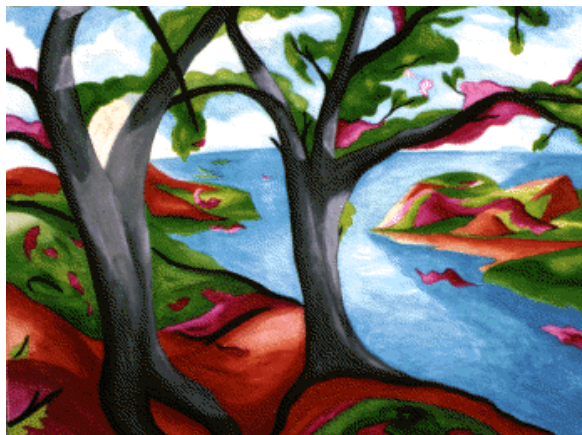


Figura 1. Imaginea inițială

Pentru clasificarea pixelilor imaginii considerate se ține cont de faptul că fiecare fixel $\mathbf{p}$ al imaginii color este caracterizat prin:

- un vector cu două componente, notat $\mathbf{g}(\mathbf{p})$ care conține coordonatele geometrice ale pixelului
- un vector cu trei componente, $\mathbf{c}(\mathbf{p})$ care exprimă coordonatele de culoare din cubul RGB.

Folosind coordonatele geometrice, să definim mulțimea pixelilor $\mathbf{G}_k$ care corespund regiunilor ocupate de culoarea $\mathbf{e}_k$, $k = 1, 2, 3, 4$, unde:

- $\mathbf{e}_1$ semnifică culoarea neagră
- $\mathbf{e}_2$ semnifică culoarea roșie
- $\mathbf{e}_3$ semnifică culoarea verde și
- $\mathbf{e}_4$ semnifică culoarea albastră.

Evident mulțimile $\mathbf{G}_k$ sunt disjuncte. În cubul RGB scalat la $[0, 1] \times [0, 1] \times [0, 1]$ definim mulțimea de culori asociată fiecărui element $\mathbf{e}_k$ cu $k = 1, 2, 3$ sau 4 .

$$C_k = \{ \cup_p c(p) \mid g(p) \in G_k \} \quad (1)$$

Ipoteză de lucru: Cel puțin din punct de vedere teoretic se poate considera că mulțimile C_1, C_2, C_3 și C_4 sunt distincte. Satisfacerea practică a acestei ipoteze depinde de calitatea imaginii color.

Algoritm

Pasul 1. Se selectează eșantioane de culoare reprezentative pentru cele trei tipuri de regiuni și se definesc mulțimile de eșantioane $\tilde{C}_1, \tilde{C}_2, \tilde{C}_3$ și $\tilde{C}_4$. Acestea constituie cea mai bună posibilitate de a obține informații parțiale despre mulțimile de culori C_1, C_2, C_3 și C_4 care sunt necunoscute ca valori concrete în cubul RGB. Se elimină vectorii identici din fiecare mulțime de culoare. În baza ipotezei de lucru considerate, $\tilde{C}_k \subseteq C_k$, $k = 1, 2, 3, 4$ și drept urmare mulțimile de eșantioane $\tilde{C}_1, \tilde{C}_2, \tilde{C}_3$ și $\tilde{C}_4$ sunt disjuncte.

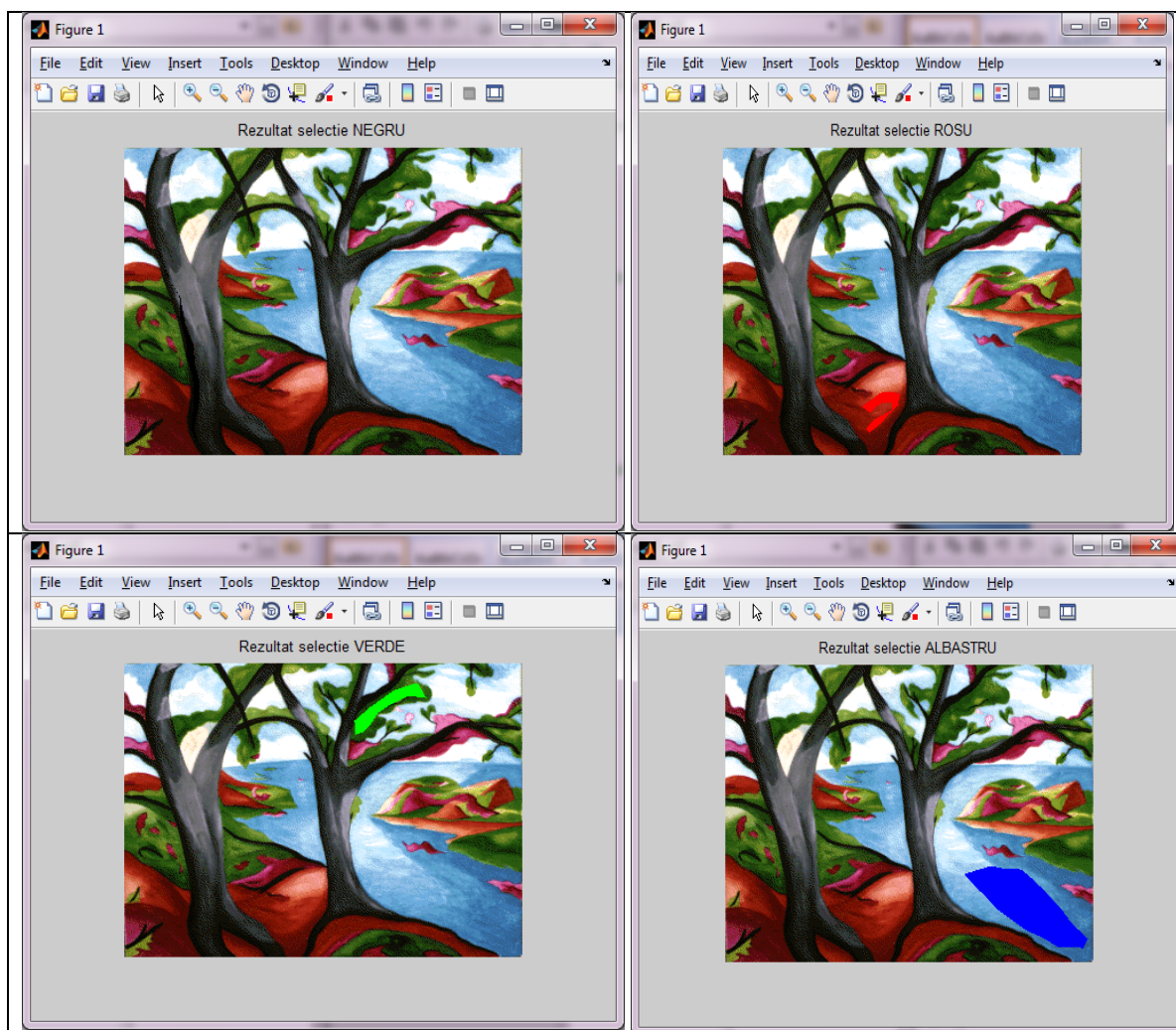


Figura 2. Alegerea eșantioanelor de culoare din regiuni reprezentative ale imaginii

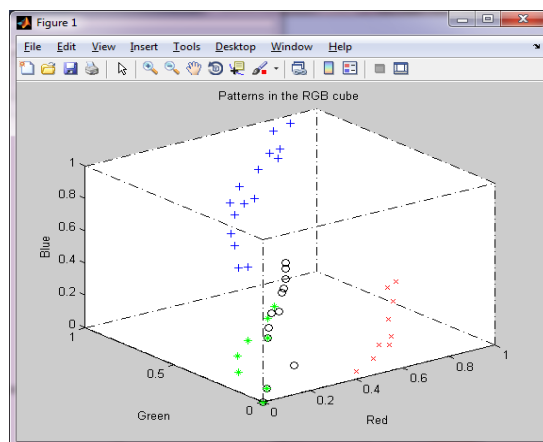


Figura 3. Plasarea culorilor din mulțimile de eșantioane $\tilde{C}_1$, $\tilde{C}_2$, $\tilde{C}_3$ și $\tilde{C}_4$ în cubul RGB scalat la $[0, 1] \times [0, 1] \times [0, 1]$.

Pasul 2. Se antrenează o rețea neurală cu arhitectură adecvată astfel încât să clasifice în trei clase culorile din mulțimile de eșantioane $\tilde{C}_1$, $\tilde{C}_2$, $\tilde{C}_3$ și $\tilde{C}_4$.

Pasul 3. Se utilizează rețeaua antrenată pentru a clasifica toți pixelii din imagine în cele trei clase, corespunzătoare lui $\tilde{C}_1$, $\tilde{C}_2$, $\tilde{C}_3$ și $\tilde{C}_4$.

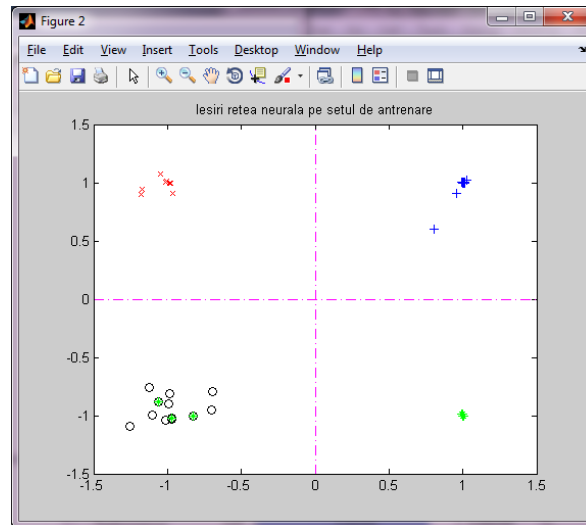


Figura 4. Ieșirile rețelei neurale pe setul de antrenare

Pasul 4. Se construiește o nouă imagine digitală în care se definesc următoarele patru regiuni care estimează regiunile ideale de interes G_k . Se utilizează trei culori distincte pentru a reprezenta regiunile estimate $\tilde{G}_1$, $\tilde{G}_2$, $\tilde{G}_3$ și $\tilde{G}_4$, pentru a asigura o mai bună vizualizare. Estimarea regiunilor ideale se va obține prin segmentare.

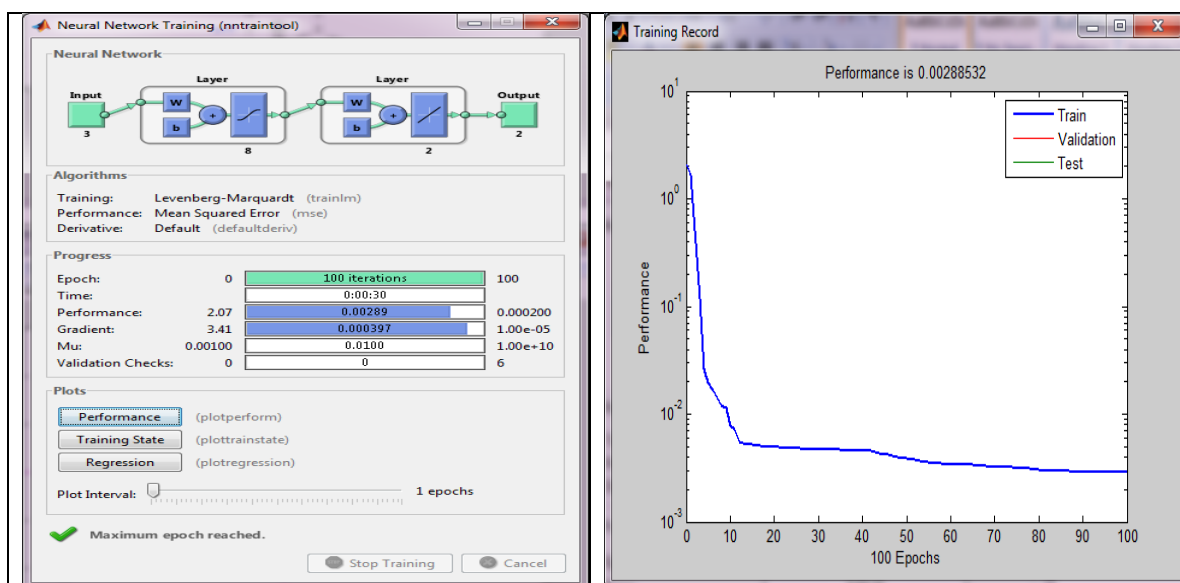


Figura 5. Performanța rețelei neurale antrenate prin algoritmul Levenberg-Marquardt

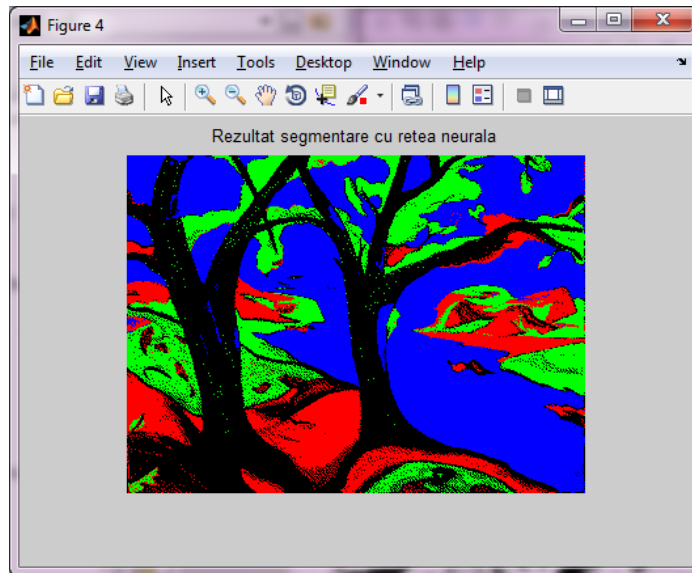


Figura 6. Separarea în clase – regiuni de interes realizată de o rețea neurală antrenată prin algoritmul Levenberg-Marquardt

Pasul 5. Se măsoară ariile regiunilor estimate $\tilde{G}_1$, $\tilde{G}_2$, $\tilde{G}_3$ și $\tilde{G}_4$ și se calculează procentele ocupate de fiecare.

3.2.Reducerea numărului de culori dintr-o imagine cu funcția imapprox

Funcția imapprox este folosită atunci când se dorește reducerea numărului de culori dintr-o imagine. Are la bază funcția rgb2ind și folosește aceleași metode de aproximare. De fapt, imapprox apelează ind2rgb pentru a converti imaginea în formatul RGB, după care apelează rgb2ind pentru a obține o imagine indexată cu mai puține culori decât imaginea inițială.

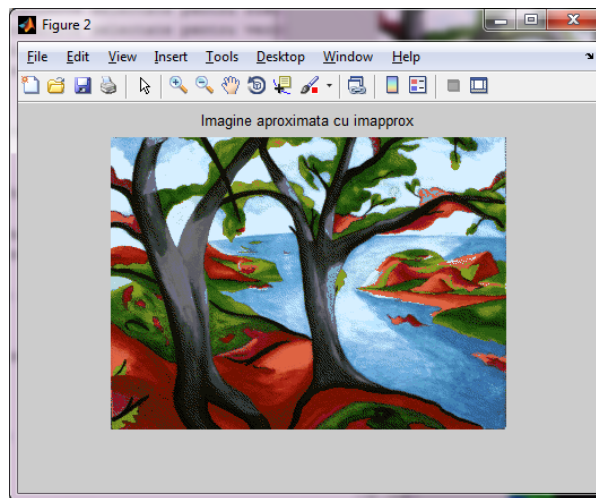


Figura 7. Aproximarea imaginii inițiale cu funcția imapprox

Calitatea imaginii rezultate depinde de metoda de aproximare utilizată, gama de culori a imaginii inițiale sau folosirea sau nu a oscilațiilor.

3.3. Identificarea regiunilor de interes cu funcția roicolor

Funcția **roicolor** poate fi folosită pentru a selecta regiuni de interes într-o imagine pe baza culorii și returnează o imagine binară.

```
BW = roicolor(A, low, high)
BW = roicolor(A, v)
```

Primul mod de apel returnează o regiune de interes ai cărei pixeli sunt în gama mapată de culori [low high]. BW este o imagine binară, semnificația unui pixel de 0 fiind că este în afara regiunii de interes, iar a unui pixel de 1 că aparține regiunii de interes.

```
BW = (A >= low) & (A <= high)
```

Cel de-al doilea tip de apel returnează o regiune de interes ai cărei pixeli corespund ca valoare elementelor din vectorul v. Semnificația elementelor de 1 ale imaginii binare BW este că elementele aferente din A corespund elementelor vectorului v. Imaginea A trebuie să fie numerică.

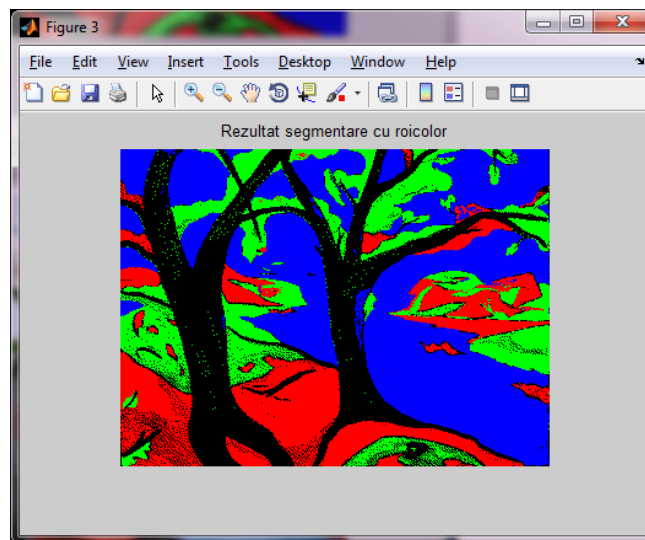


Figura 8. Aproximarea imaginii inițiale cu roicolor

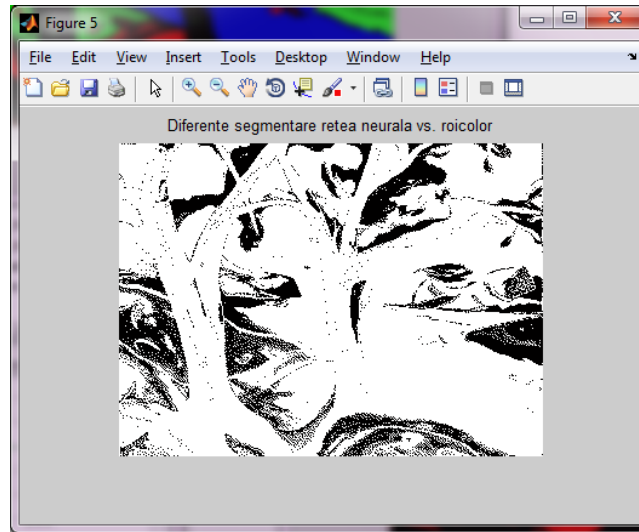


Figura 9. Diferențele între segmentarea utilizând o rețea neurală sau segmentarea cu funcția roicolor.

4. Problematika propusă pentru studiu

Problema 4.1.

Se consideră imaginea RGB din figura 1 reprezentând o un peisaj ce conține patru elemente diferite de culoare: negru, roșu, verde și albastru.

Să se precizeze componentele rețelei neurale:

- mulțimea vectorilor prototip;
- mulțimea vectorilor țintă;
- numărul de straturi și numărul de neuroni de pe fiecare strat;
- rutina de antrenare și rutina de învățare;
- numărul epocilor de antrenare;
- funcția de evaluare a performanțelor rețelei;
- denumirea și dimensiunile matricii care stochează imaginea segmentată.

Să se determine procentele din aria totală a imaginii pe care le ocupă fiecare din cele patru elemente.

Se va studia și se va lansa în execuție programul **nn_segment** elaborat în MATLAB.

NEURAL NETWORK TOOLBOX – FUNCȚIILE INTERFEȚEI GRAFICE

5. Interfața grafică a NNToolbox

Interfața grafică a NNToolbox oferă următoarele facilități pentru a lucra cu rețele neurale:
>>nnstart

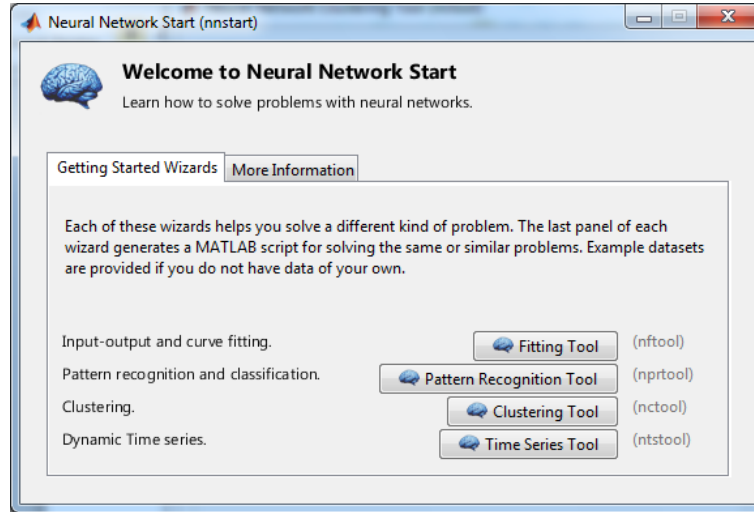


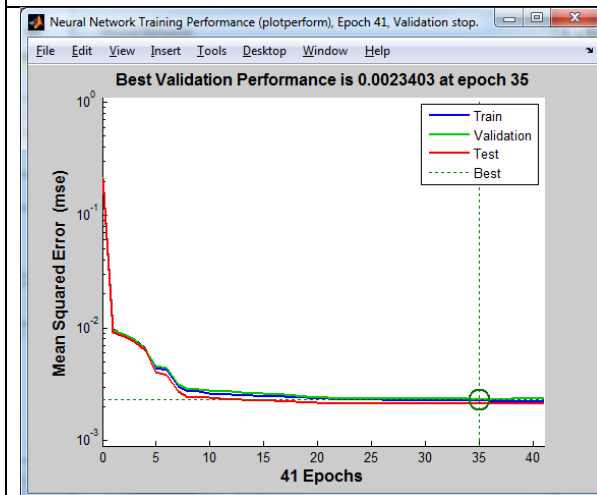
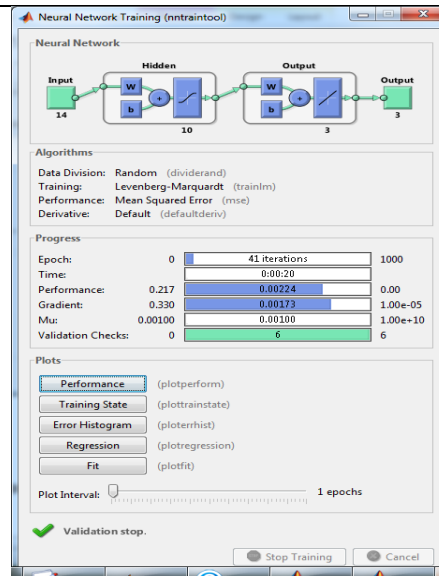
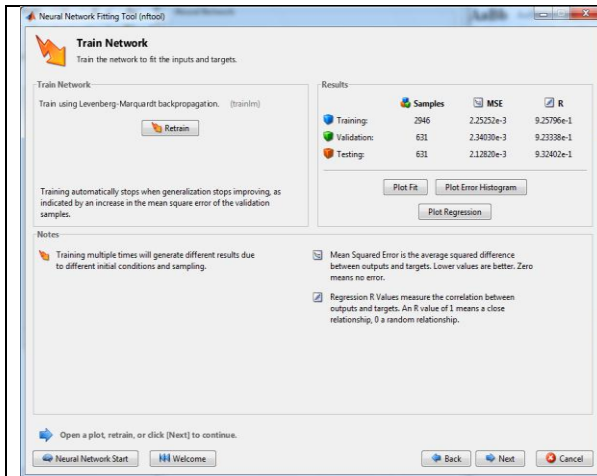
Figura xx. Neural Network Start

- **nctool** – toolbox pentru clasificare sau clusterizare folosind rețele neurale;
- **nftool** – toolbox pentru aproximare folosind rețele neurale;
- **nntraintool** – toolbox pentru antrenarea unei rețele neurale;
- **nprtool** – toolbox pentru recunoașterea caracteristicilor;
- **ntstool** – toolbox pentru rețele neurale cu serii și
- **view** - pentru vizualizarea arhitecturii unei rețele neurale.

5.1. Nftool

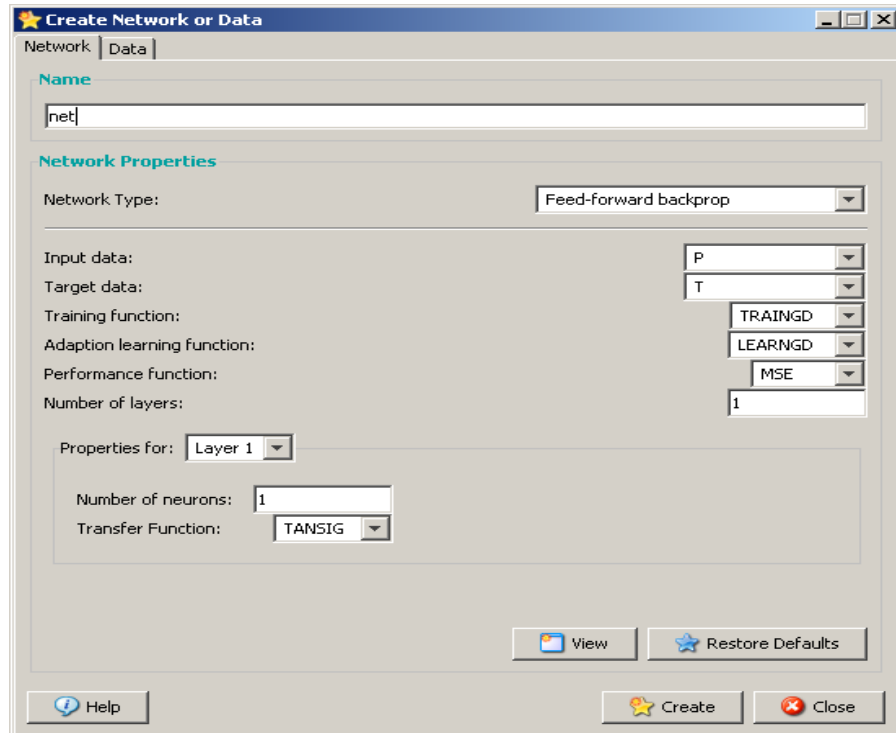
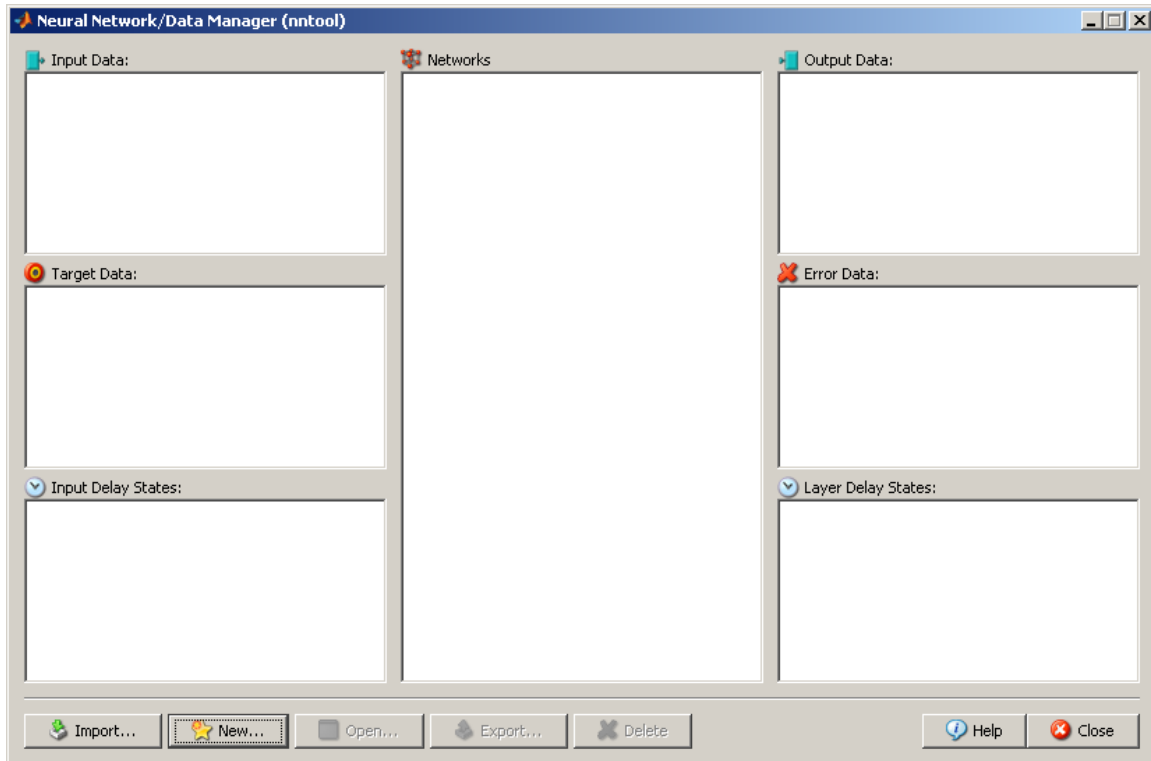
The screenshots illustrate the workflow of the Neural Network Fitting Tool (nftool):

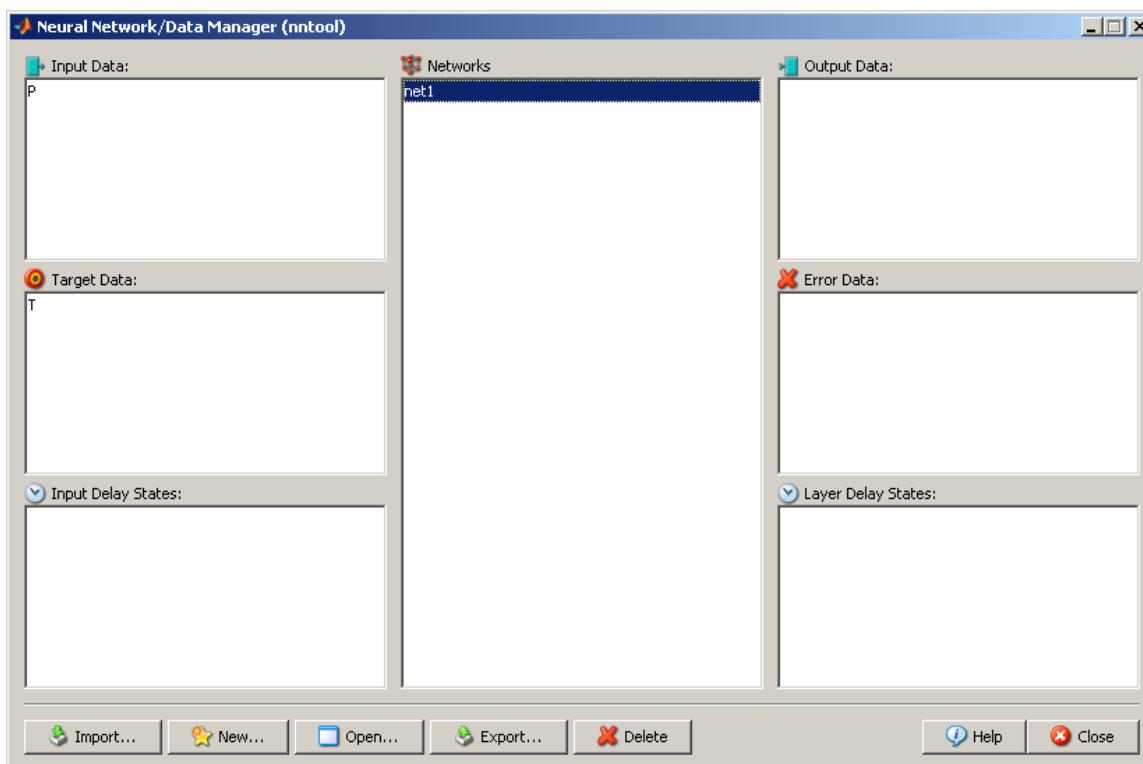
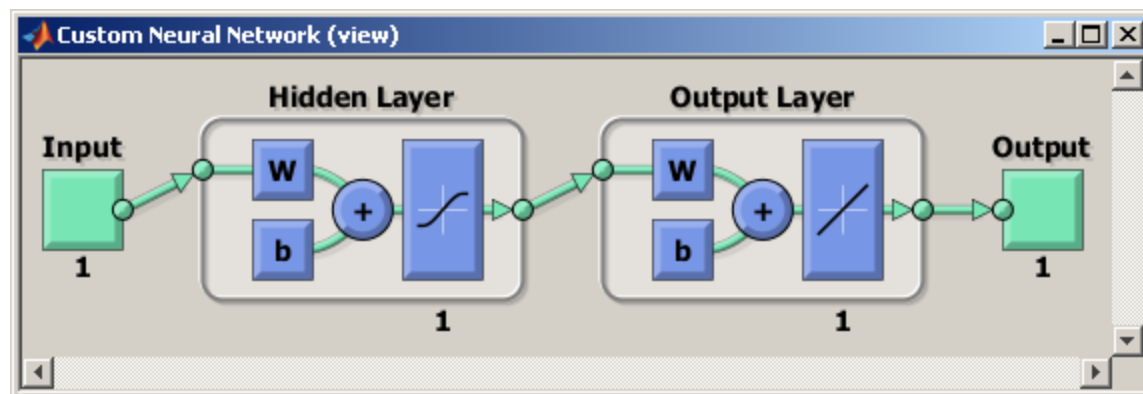
- Welcome to the Neural Network Fitting Tool:** Provides an introduction to the tool and its purpose. It includes a diagram of a two-layer feed-forward network with an input layer, a hidden layer, and an output layer.
- Select Data:** Allows users to select input and target data from the workspace. It includes a summary of the selected data and a button to load an example data set.
- Validation and Test Data:** Allows users to set aside some samples for validation and testing. It includes a summary of the validation and test data and a button to restore defaults.
- Network Architecture:** Allows users to set the number of neurons in the fitting network's hidden layer. It includes a recommendation section and a diagram of the network architecture.
- Train Network:** Allows users to train the network to fit the inputs and targets. It includes a table of results showing the number of samples, MSE, and R for training, validation, and testing. It also includes a section for notes and a button to plot the error histogram.



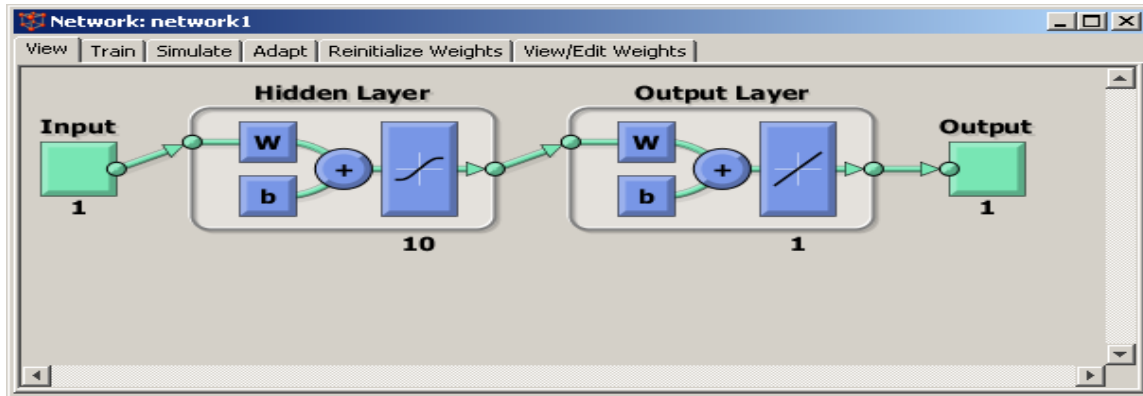
5.2. Nntool

>>nntool





5.3. Nntraintool



The window displays training parameters and results for "Network: network1".

Training Info

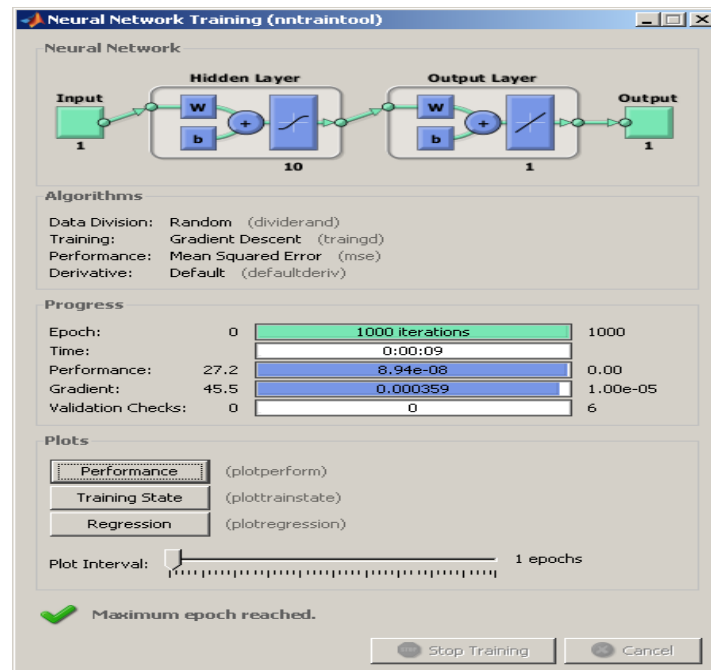
Training Data

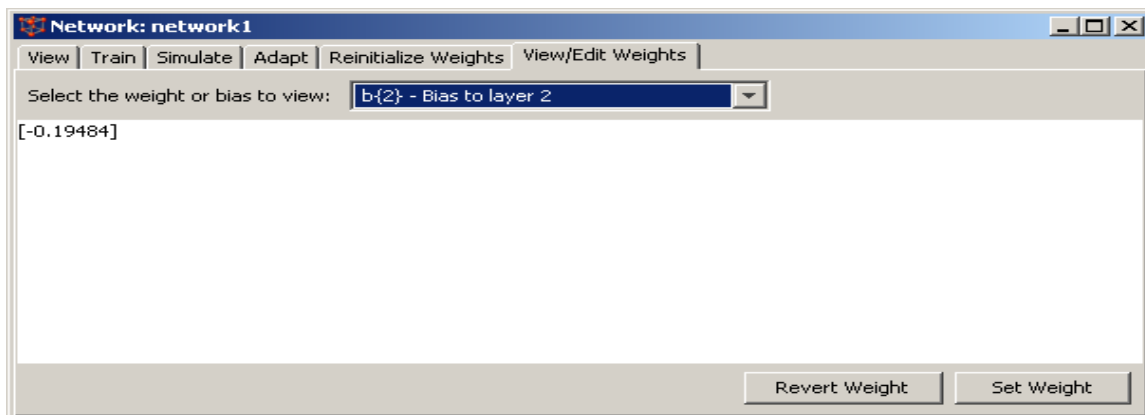
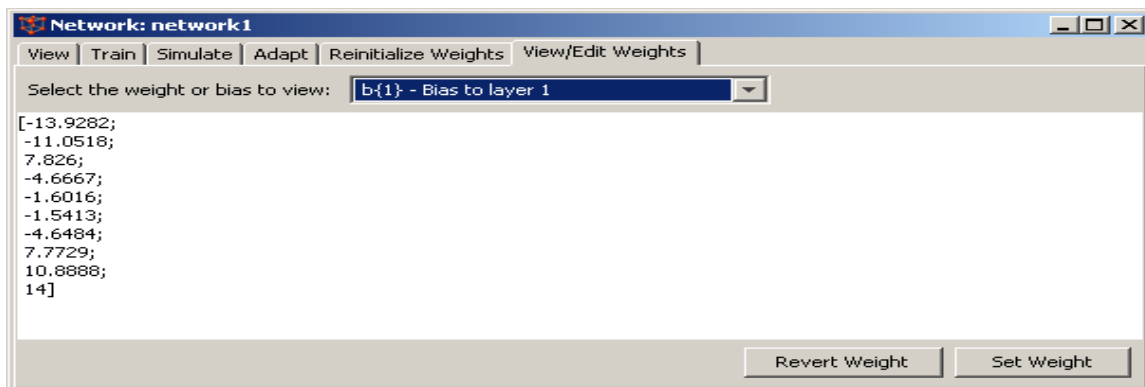
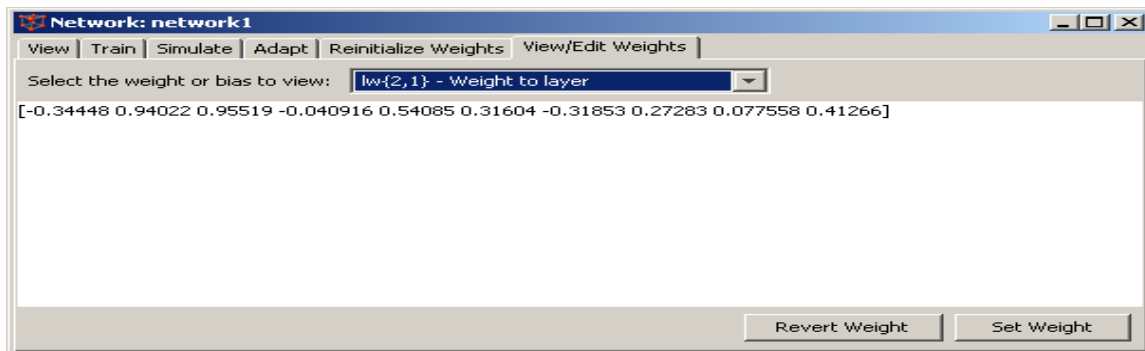
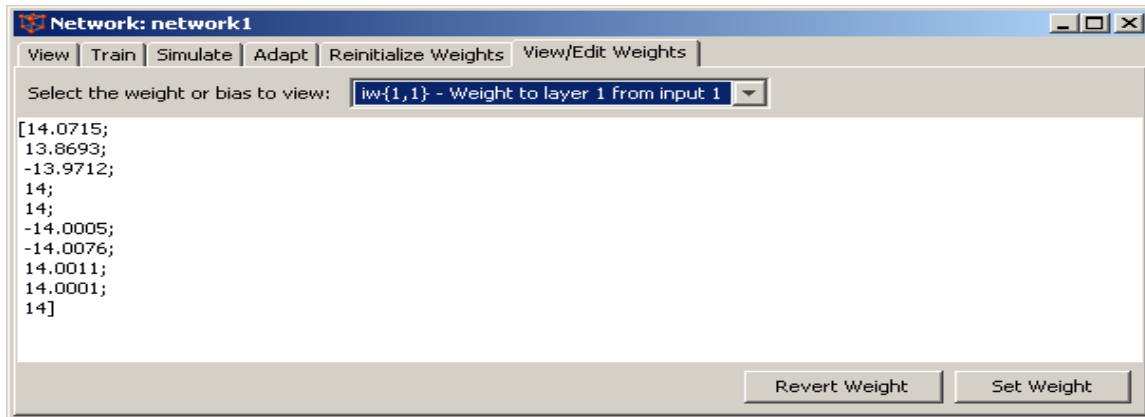
Inputs	P
Targets	T
Init Input Delay States	(zeros)
Init Layer Delay States	(zeros)

Training Results

Outputs	network1_outputs
Errors	network1_errors
Final Input Delay States	network1_inputStates
Final Layer Delay States	network1_layerStates

Train Network





6. Problematika propusă pentru studiu

Problema 6.1.

Să se construiască o rețea neurală cu două straturi, cu $s = 2, 4$ sau 8 neuroni, cu funcție de activare sigmoid bipolar pe stratul ascuns și 1 neuron cu funcție de activare liniară pe stratul de ieșire. Se consideră vectorul prototip P calculat după formula:

$$P = \frac{i - 1}{50} \text{ cu } i = 1, \dots, 51$$

și vectorul țintă calculat după formula:

$$T = 1 + \exp(P(i)) \cdot \sin\left(2 \cdot \pi \cdot P(i) - \frac{\pi}{2}\right) \text{ cu } i = 1, \dots, 51.$$

Pentru fiecare din cele trei cazuri să se deseneze

- arhitectura rețelei;
- graficul evoluției erorii medii pătratice pe parcursul antrenării;
- graficul funcției (P, T) și al aproximării cu rețeaua MLP pe aceeași figură.

și să se noteze:

- numărul de iterații cât a durat antrenarea;
- ponderile și deplasările aferente neuronilor de pe stratul ascuns;
- ponderea și deplasarea aferentă neuronului de pe stratul de ieșire;
- ponderile pentru conexiunile dintre stratul de ieșire și stratul ascuns.