UNIVERSITATEA TEHNICĂ „GHEORGHE ASACHI” IAŞI FACULTATEA AUTOMATICĂ ŞI CALCULATOARE

DOMENIUL: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI PROFIL DE STUDIU: TEHNOLOGIA INFORMAȚIEI

### Inteligență artificială - Proiect

## Antrenarea unei rețele neuronale de tip perceptron multistrat cu o structură predefinită cu ajutorul unui algoritm evolutiv

#### Studenți,

Radu Ștefan Vlăduț - 1405B

Donici Leonardo Mario - 1405B

#### Profesor Coordonator,

Prof.dr. ing. Mircea Hulea

#### An universitar 2024-2025

**Cuprins**

1. [**Descrierea problemei**](#_heading=h.gjdgxs) **3**
2. [**Aspecte teoretice privind algoritmul**](#_heading=h.30j0zll) **3**
3. [**Modalitatea de rezolvare**](#_heading=h.1fob9te) **5**
4. [**Listarea părților semnificative din codul sursă însoțite de explicații și comentarii**](#_heading=h.3znysh7) **6**
5. [**Rezultatele obținute prin rularea programului în diverse situații**](#_heading=h.2et92p0) **10**
6. [**Concluzii**](#_heading=h.tyjcwt) **11**
7. [**Bibliografie**](#_heading=h.3dy6vkm) **12**
8. [**Contribuțiile membrilor**](#_heading=h.1t3h5sf) **12**

# Descrierea problemei

Problema constă în utilizarea unei rețele neuronale de tip perceptron multistrat cu o structură predefinită (număr de straturi ascunse și neuroni) pentru a rezolva o anumită sarcină, folosind un algoritm evolutiv. Structura rețelei, precum și funcția de activare, sunt stabilite în prealabil. Se recomandă utilizarea a 1-2 straturi ascunse și selectarea unui număr adecvat de neuroni pentru fiecare strat, în funcție de complexitatea problemei alese. Algoritmul evolutiv va fi implementat pentru a determina ponderile conexiunilor și pragurile neuronilor din rețea. În final, se vor analiza rezultatele rețelei pe setul de antrenare aferent problemei.

Pentru acest scop, s-a utilizat un set de date din UCI Machine Learning Repository:  
[Setul de date Iris](https://archive.ics.uci.edu/ml/datasets/Iris), care tratează clasificarea speciilor de plantă iris. Acest set de date include 3 clase (Iris Setosa, Iris Versicolour și Iris Virginica), fiecare cu câte 50 de instanțe. Datele sunt structurate pe baza a 5 atribute: lungimea și lățimea sepalelor, lungimea și lățimea petalelor, precum și clasa din care face parte fiecare instanță.

# Aspecte teoretice privind algoritmul

Aplicația utilizează un algoritm evolutiv pentru a identifica cele mai bune soluții pentru problema dată. Algoritmii evolutivi sunt metode de optimizare inspirate din procesele evoluției biologice, care presupun utilizarea unui set de soluții posibile ce sunt modificate succesiv prin operațiuni aleatorii pentru a găsi soluția optimă. Membrii populației reprezintă variante diferite de soluționare a problemei.

Selecția părinților se face pe baza valorii funcției de adaptare, care măsoară cât de bună este fiecare soluție, iar selecția de tip competiție alege cel mai adaptat individ dintre părinți. Funcția de adaptare (fitness) indică cât de bine este un cromozom în raport cu obiectivele problemei.

Încrucișarea are loc cu o probabilitate specificată prin interfață și presupune combinarea informațiilor de la doi părinți (mamă și tată) pentru a genera un copil. Ulterior, se aplică mutația, care introduce variație genetică în populație, cu scopul de a explora mai multe soluții și de a evita stagnarea într-un optim local. Mutația ajută la menținerea diversității populației și contribuie la găsirea unor soluții mai bune.

În cea de a doua parte a parte a programului a fost folosită o rețea neuronala tip perceptron multistrat ce prezintă atât un număr de neuroni cât și un strat ascuns. Rețeaua neuronala este un procesor masiv paralel ce se distinge prin abilitatea sa de a stoca cunoștințe pentru a fi reutilizate. Perceptronul multi-strat este o rețea neuronala cu propagare înainte, iar calculele se realizează numai în neuronii din straturile ascunse și în stratul de ieșire. Semnale de intrare sunt propagate înainte succesiv prin straturile rețelei.

Un strat ascuns este o componentă a unei rețele neuronale artificiale situată între stratul de intrare și cel de ieșire. Acesta conține neuroni care procesează informațiile primite de la stratul de intrare și generează rezultatele ce sunt transmise mai departe la stratul de ieșire. Deoarece stratul ascuns nu are o conexiune directă cu mediul exterior, nu este clar care ar trebui să fie ieșirea specifică a unui neuron din acest strat. Astfel, acest strat este adesea denumit "cutia neagră" a rețelei, deoarece designul său intern nu este vizibil și nu este ușor de înțeles procesul prin care informațiile trec de la intrare la ieșire.

# Modalitatea de rezolvare

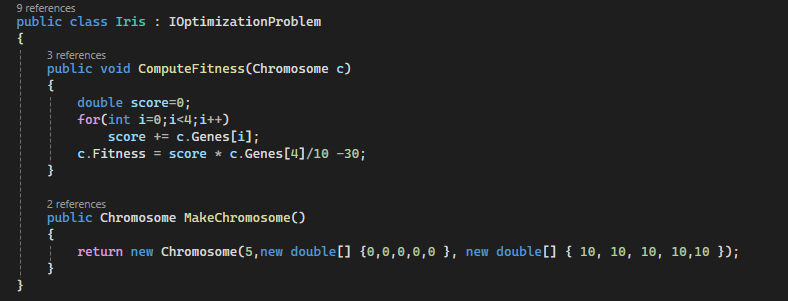
Procesul începe prin citirea numărului de indivizi din populație, numărului de generații și a valorilor operatorilor de mutație direct din interfața utilizatorului. Ulterior, populația este inițializată utilizând datele din setul „[iris.data](http://iris.data)”. Algoritmul evolutiv progresează prin selecția de tip turneu (Tournament), unde doi indivizi din populatie sunt aleși aleatoriu, iar cel mai bine adaptat dintre aceștia este selectat. Noua generație este generată prin aplicarea unui crossover aritmetic pentru a produce descendenți, urmată de o mutație care contribuie la diversificarea populației.

Perceptronul multistrat este antrenat prin apăsarea butonului „Antrenează”, utilizând un algoritm genetic pentru a stabili ponderile și pragurile. Fiecare pondere este calculată pe baza celei mai adaptate soluții din populație, cu ajutorul funcției de adaptare (fitness). Ulterior, perceptronul, având doi neuroni în stratul ascuns, calculează ieșirea pentru diverse intrări x și y. Pentru fiecare neuron, se determină suma ponderată a intrărilor, se scade pragul, iar rezultatul este procesat prin funcția de activare aleasă de utilizator (prag, semi liniară sau sigmoidală unipolară).

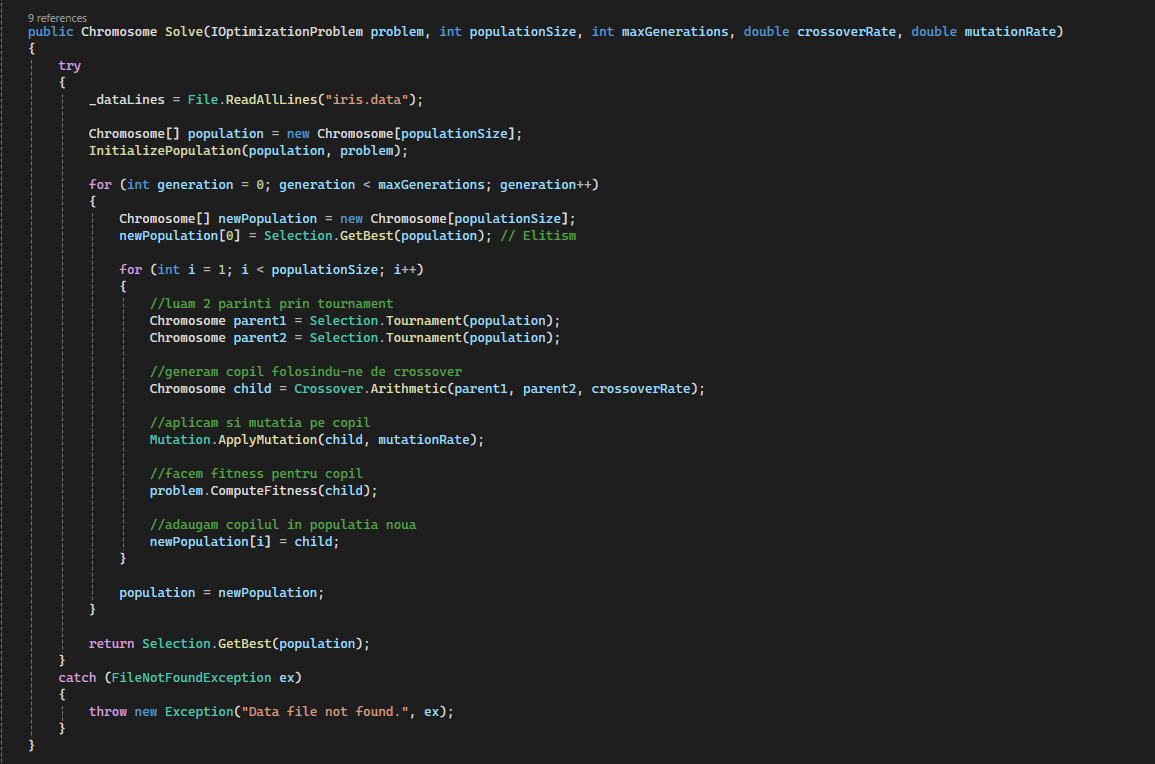
La final, regiunile de decizie ale perceptronului multistrat sunt afișate vizual în cadrul picturebox ului.

# Listarea părților semnificative din codul sursă însoțite de explicații și comentarii

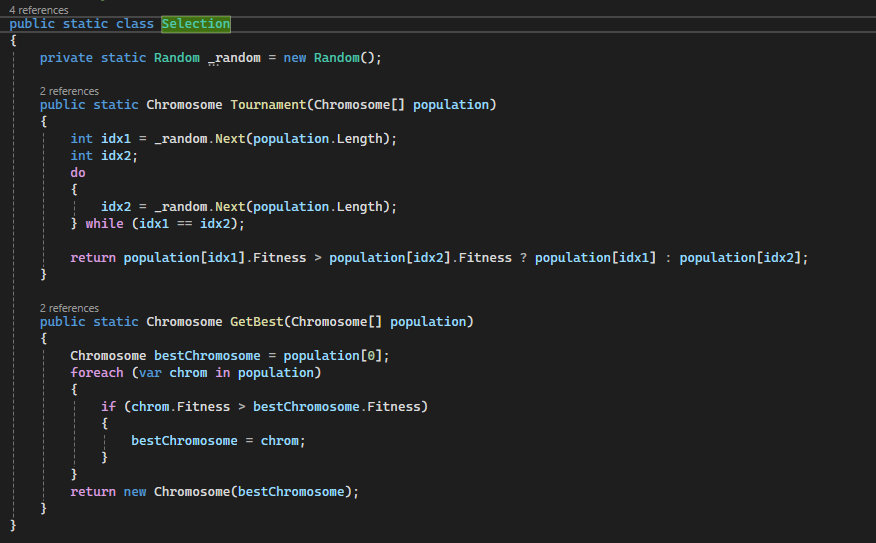
Clasa Iris calculeaza Fitness-ul și inițializează genele cromozomilor cu valori



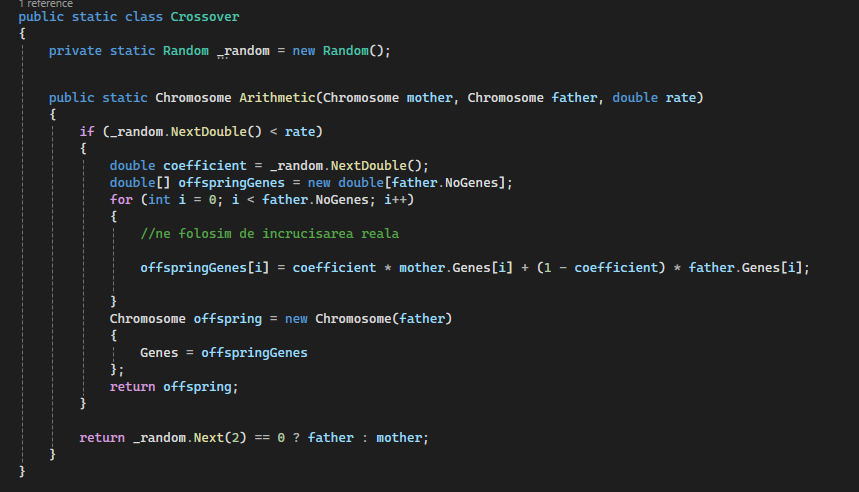
Clasa principala EvolutionaryAlgorithm ce citește datele de intrare din setul de date iris.data, inițializează populația cu metoda InitializaPopulation și aplică etapele algoritmului pentru selecție.



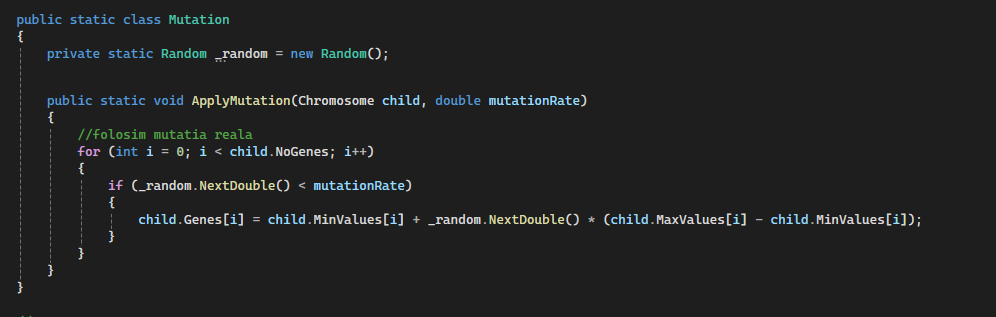
//Clasa Selection ce definește metoda de selecție Tournament pentru a face selecția între indivizi



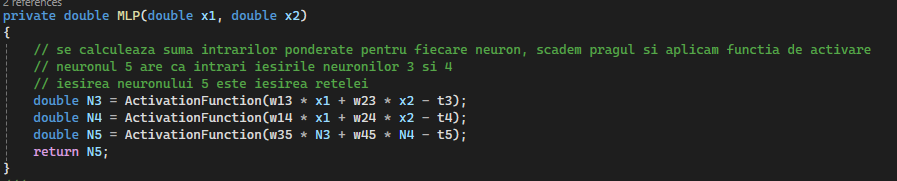
//Clasa crossover ce efectuează încrucișarea aritmetică a 2 părinți



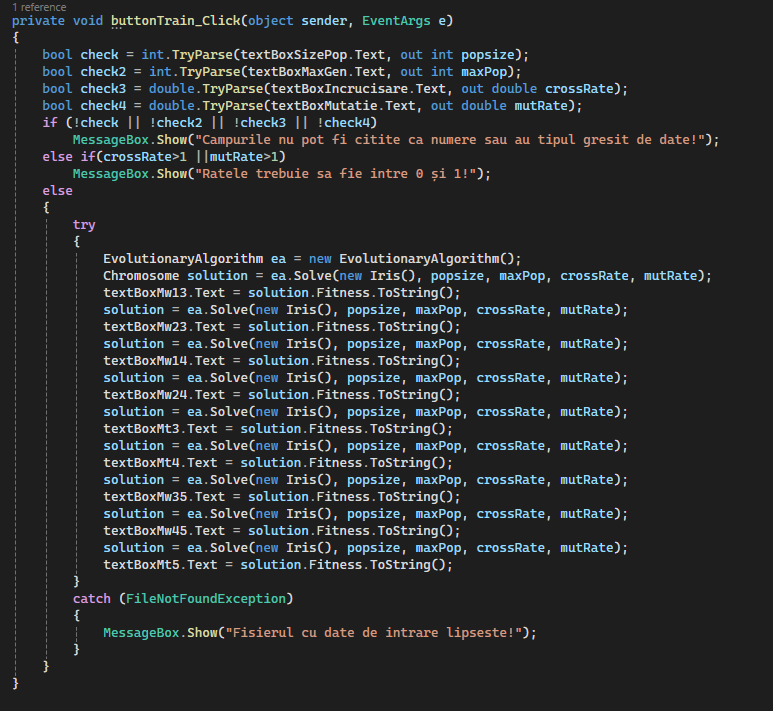
//Funcția de aplicare a mutațiilor asupra populației



//Calculul ieșirii perceptronului

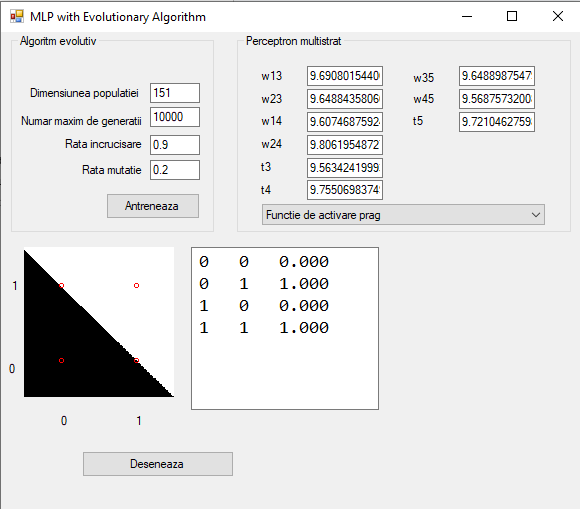


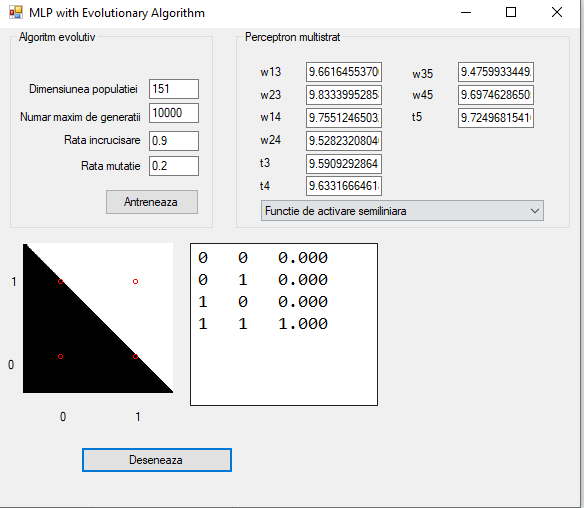
//Metoda de antrenare a perceptronului multistrat și setarea pragurilor și a ponderilor

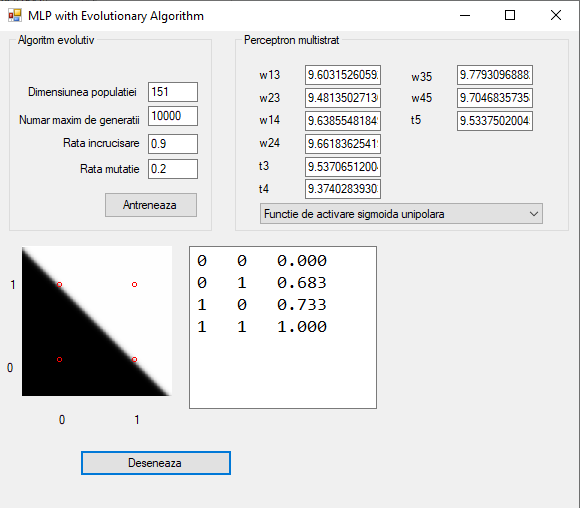


# Rezultatele obținute prin rularea programului în diverse situații, capturi ecran și comentarii asupra rezultatelor obținute

Mai jos sunt prezentate rezultate ale rulării programului cu funcție de activare prag, semilinară și sigmoidă respectiv







# Concluzii

În concluzie, algoritmul genetic poate fi folosit pentru determinarea ponderilor conexiunilor și a pragurilor neuronilor într-o rețea neuronală de tip perceptron multistrat. Totuși, este necesar un număr considerabil de generații și valori mari pentru factorii de încrucișare și mutație, pentru a preveni convergența la o soluție locală.

# [Bibliografie](https://docs.google.com/document/d/1iEvwqEatPrv8aGGJPhRa5ZpWAkAdiu7txUj2rdMVVFQ/edit#heading%3Dh.37jp6mlg6tvw)

<https://archive.ics.uci.edu/ml/datasets/Iris>

[https://www.researchgate.net/profile/Shifei-Ding/publication/220637688\_An\_optimizing\_BP\_ne](https://www.researchgate.net/profile/Shifei-Ding/publication/220637688_An_optimizing_BP_neural_network_algorithm_based_on_genetic_algorithm/links/00b7d537bf2fe1657b000000/An-optimizing-BP-neural-network-algorithm-based-on-genetic-algorithm.pdf) [ural\_network\_algorithm\_based\_on\_genetic\_algorithm/links/00b7d537bf2fe1657b000000/An-opt](https://www.researchgate.net/profile/Shifei-Ding/publication/220637688_An_optimizing_BP_neural_network_algorithm_based_on_genetic_algorithm/links/00b7d537bf2fe1657b000000/An-optimizing-BP-neural-network-algorithm-based-on-genetic-algorithm.pdf) [imizing-BP-neural-network-algorithm-based-on-genetic-algorithm.pdf](https://www.researchgate.net/profile/Shifei-Ding/publication/220637688_An_optimizing_BP_neural_network_algorithm_based_on_genetic_algorithm/links/00b7d537bf2fe1657b000000/An-optimizing-BP-neural-network-algorithm-based-on-genetic-algorithm.pdf)

<http://florinleon.byethost24.com/Curs_IA/IA04_Optimizare1.pdf>

# Contribuțiile membrilor

Radu Ștefan-Vlăduț - partea de algoritm genetic

Donici Leonardo Mario - MainForm-ul, interfața și metodele din spate