UNIVERSITATEA “LUCIAN BLAGA” DIN SIBIU

FACULTATEA DE INGINERIE

**Proiect/Lucrări de laborator la**

**Simularea și optimizarea arhitecturilor de calcul**

Autori: Rațiu Iulia – 41/2

Onofrei Darius-Ioan – 41/2

Coordonator științific laborator:

asist. drd. ing. Andrei Pătrăușanu

2024-2025

Cuprins

[1. Cerința temei și descrierea arhitecturii implementate 3](#_Toc185100300)

[2. Ghid de utilizare a simulatorului 4](#_Toc185100301)

[3. Structura software 5](#_Toc185100302)

[4. Resurse hardware și software necesare rulării aplicației 5](#_Toc185100303)

[5. Rezultate obținute cu simulatorul implementat 6](#_Toc185100304)

[6. Concluzii 6](#_Toc185100305)

[7. Idei de dezvoltări viitoare 6](#_Toc185100306)

# Cerința temei și descrierea arhitecturii implementate

Cerința temei

Scopul acestui proiect este implementarea unui simulator pentru predicția salturilor condiționate folosind un model simplu bazat pe perceptron. Acest simulator permite evaluarea acurateței predicțiilor și a altor metrici de performanță, cum ar fi rata de predicții greșite și ratele pozitive/negative adevărate.

Arhitectura implementată

Arhitectura aplicației include următoarele componente esențiale:

* Predictorul bazat pe perceptron:
  + Responsabil pentru simularea și ajustarea predicțiilor pe baza istoricului instrucțiunilor.
* Registrul istoric (HR):
  + Stochează istoricul instrucțiunilor recente pentru a influența predicțiile viitoare.
* Registrul perceptronilor:
  + Conține o colecție de perceptroni care gestionează ponderile asociate fiecărei intrări istorice.
* Parser de fișiere:
  + Permite încărcarea fișierelor cu instrucțiuni de tip branch pentru simulare.
* Metrici de performanță:
  + Calculează rezultate precum acuratețea, rata de predicții greșite, rata pozitivă și rata negativă adevărată.

Flux de funcționare

1. Fișierul cu instrucțiuni este încărcat și analizat pentru a genera un registru de branch-uri.
2. Simularea predicțiilor este efectuată pentru fiecare branch din registru folosind perceptronii disponibili.
3. Metricile de performanță sunt calculate și afișate pentru evaluarea rezultatului.

# Ghid de utilizare a simulatorului

Pașii principali:

1. Adăugarea fișierului: Utilizează butonul Add File pentru a încărca un fișier cu extensia .tra.
2. Vizualizarea branch-urilor: După încărcare, branch-urile extrase sunt afișate în lista Branches.
3. Setarea parametrilor: Configurează numărul de perceptroni și dimensiunea registrului istoric folosind controalele Perceptrons Number și HR Size.
4. Simularea: Apasă butonul Simulate pentru a iniția procesul de predicție.
5. Analiza rezultatelor: Metricile de performanță și predicțiile sunt afișate în interfață.

Input și output:

* Input: Fișiere .tra care conțin informații despre tipurile și acțiunile branch-urilor.
* Output: Metrici calculate (acuratețea, rata de predicții greșite etc.) și predicțiile efectuate pentru fiecare branch.

# Structura software

Clase principale:

1. BranchPredictor: Conține logica pentru simularea branch-urilor și ajustarea ponderilor perceptronilor.
2. HRRegistry: Gestionează istoricul instrucțiunilor (HR) utilizat de predictor.
3. PerceptronRegistry: Stochează perceptronii și gestionează operațiile asupra acestora.
4. BranchMetrics: Calculează metrici precum acuratețea și rata predicțiilor greșite.
5. FileParser: Analizează fișierele de intrare pentru a genera branch-uri.
6. Form1: Interfața grafică principală pentru utilizatori.

Interacțiuni:

* Predictorul utilizează date din registrul istoric și perceptronii pentru a calcula predicțiile.
* FileParser încarcă branch-urile înregistrate în BranchRegistry.

# Resurse hardware și software necesare rulării aplicației

Cerințe software:

* Windows 10/11.
* .NET Framework 4.7.2 sau mai recent.

Cerințe hardware:

* Procesor: minim 2 GHz.
* Memorie RAM: minim 2 GB.
* Spațiu de stocare: minim 100 MB pentru fișierele aplicației și simulărilor.

# Rezultate obținute cu simulatorul implementat

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Branch-uri Totale | Predicții Corecte | Predicții Incorecte | Acuratețea | Rata Predicțiilor Greșite |
|  |  |  |  |  |

# Concluzii

Simulatorul SaltyPerceptron demonstrează că perceptronul simplu poate oferi rezultate rezonabile în predicția salturilor condiționate. Performanța acestuia depinde de configurarea parametrilor (dimensiunea HR, numărul perceptronilor) și de calitatea datelor de intrare.

# Idei de dezvoltări viitoare

1. Extinderea simulatorului pentru a suporta perceptroni mai avansați (ex. perceptron fast path-based).
2. Adăugarea suportului pentru simularea pe mai multe benchmark-uri (ex. SPEC 2017).