

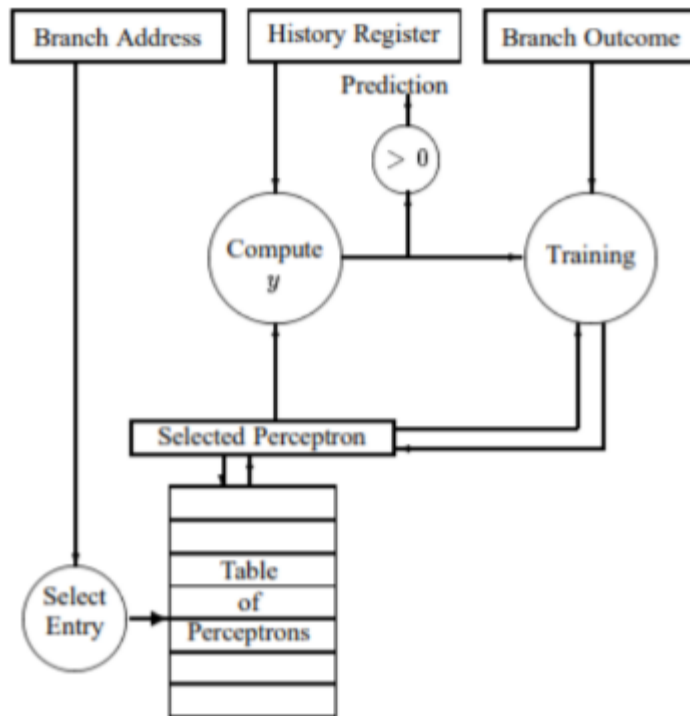
Proiect la Simularea si optimizarea arhitecturilor de calcul

Autor: Alexandrin Caraganciu 244 / 2

Indrumator: Conf.univ.dr.ing. Adrian FLOREA

ULBS, 2023

Implementarea unui predictor de salturi condiționate de tip perceptron simplu, fast path-based și idealized piecewise. Analiză comparativă asupra costurilor. Modelare și implementare sub platformă academică SPEC 2000 și industrială standardizată CBP – campionatul mondial de predicția salturilor. Simularea prin programare distribuita a benchmark-urilor.



Pentru predictor tip perceptron noi avem nevoie de tabela de perceptroni, unde fiecare perceptron are 2 vectori de ponderi, unde numarul de ponderi este variabil. Registru de istoria globala HR, pe k biti, unde k este variabil. Perceptronul din tabela de perceptroni este selectat ca PC curent % numarul de perceptroni.

Requirements:	2
Run build	2
Run project	2
Ghid de Utilizare	3
Ghid de Dezvoltare	3
Ierarhie clase	3
Program class	3
Metode	3
Data class	3
Atribute	3
Metode	4
HRg Class	4
Atribute	4
Metode	4
Perceptron Class	4
Atribute	4
Metode	4
Predictor Class	4
Atribute	4
Metode	5
Form Class	5
Atribute	5
Metode	5
Posibilitate de Extindere	5
Interpretari	5
Concluzii	6

Requirements:

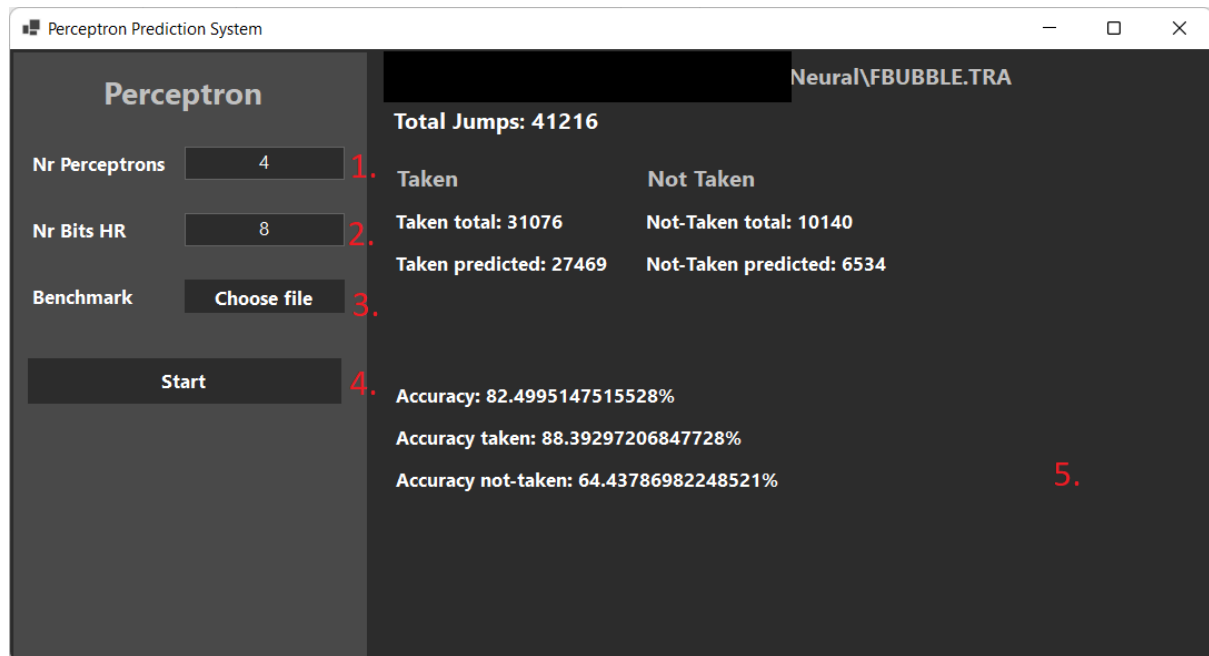
Run build

- .Net Framework 4.8

Run project

- .Net Framework 4.8
- Windows Forms C#

Ghid de Utilizare



1. Select perceptron number
2. Select number of bits in History Register (HR)
3. Select a benchmark file (.tra)
4. Press “*Start*” button
5. See results and statistics

Ghid de Dezvoltare

Ierarhie clase

Program class

Initializeaza clasa formei (Form).

Metode

- **Main**

Data class

Reprezenteaza o linie din fisierul .tra

Atribute

- **isTaken: bool** - Contine informatie daca saltul era facut sau nu
- **PC: int** - Contine PC-ul curent a instructiunii de salt

Metode

- **Data: constructor** - Initializeaza attribute
- **GetData(fileName: string): Data[]** - metoda statica, care citeste date din fisierul dat ca parametru si genereaza un obiect a clasei *Data* pentru fiecare linie din fisierul .tra. Returneaza un vector de obiecte a clasei *Data*

HRg Class

Atribute

- **hr: Queue<bool>** - o coada, care reprezinta istoria globala a salturilor

Metode

- **HRg(maxSize: int = 8): constructor** - umple hr cu maxSize false-uri
- **AddValue(addValue: bool): void** - Adauga un salt la hr si scoate primul adaugat (ca sa nu avem mai multe de maxSize salturi)

Perceptron Class

Atribute

- **wt: int[]** - un vector de ponderi pentru saltul facut
- **wnt: int[]** - un vector de ponderi pentru saltul NE facut

Metode

- **Perceptron(size: int = 8): constructor** - initializeaza ponderi a lui *wt* cu 1 si *wnt* cu -1
- **ActivationFunction(value: double): bool** - Aplica functia de activare pentru valoarea transmisa ca parametru. Returneaza rezultatul ultimei.
- **Calculate(hrg: HRg): bool** - Calculeaza iesirea din perceptron. Returneaza valoarea la iesire dupa aplicarea functiei de activare
- **AdjustWeights(realValue: bool): void** - Ajusteaza ponderile perceptronului, daca saltul era predictionat gresit

Predictor Class

Atribute

- **perceptrons: Perceptron[]** - Vector de perceptroni
- **hrg: HRg** - Obiectul clasei HRg (History Register)
- **data: Data[]** - Datele citite din fisierul .tra
- **totalJumps: int** - Numarul total de salturi din fisierul .tra
- **totalJumpsTaken: int** - Numarul total de salturi taken din fisierul .tra
- **totalJumpsTakenPredicted: int** - Numarul total de salturi taken predictionate
- **totalJumpsNotTaken: int** - Numarul total de salturi not-taken din fisierul .tra
- **totalJumpsNotTakenPredicted: int** - Numarul total de salturi not-taken predictionate

Metode

- **Predictor(perceptronNumber: int, hrgSize: int, data: Data[]): constructor** - initializeaza attributele
- **Train(): void** - Parcurge datele din atributul *data* si incearca sa predictioneze salturi

Form Class

Atribute

- **data: Data[]?** - Contine datele citite din fisierul curent selectat

Metode

- **startButton_Click(sender: object, e: EventArgs): void** - start button click handler. Initializeaza obiectul clasei *Predictor* si apeleaza metoda *Train*
- **chooseFileButton_Click(sender: object, e: EventArgs): void** - choose file button click handler. Deschide fereastra pentru selectarea fisierilor. Daca fisierul a fost selectat, apeleaza metoda *GetData* din clasa *Data* si citeste datele din fisier in atributul *data*

Posibilitate de Extindere

Programul poate lucra si cu alte clase de predictie. Pentru asta doar trebuie schimbata clasa de la care se initializeaza un obiect in metoda *startButton_Click*. Metoda *GetData* din Clasa *Data* poate fi schimbata pentru a prelucra datele din orice tip de fisier.

Interpretari

HR = 8 biti

	Nr Bits HR	
Nr Perceptron	FSort	FPuzzle
4	62.80%	86.35%
8	71.43%	88.26%
16	71.43%	91.47%
32	73.93%	93.36%
64	72.75%	93.25%

HR = 16 biti

Nr Perceptron	Nr Bits HR	
	FSort	FPuzzle
4	62.10%	86.29%
8	71.17%	88.26%
16	71.20%	91.55%
32	74.27%	93.41%
64	73.35%	93.29%

HR = 32 biti

Nr Perceptron	Nr Bits HR	
	FSort	FPuzzle
4	61.97%	86.22%
8	71.19%	88.23%
16	71.20%	91.54%
32	74.43%	93.40%
64	73.34%	93.28%

FSort - cel mai scurt fisier (cele mai putine salturi)

FPuzzle - cel mai lung fisier (cele mai multe salturi)

Concluzii:

- Cu cat mai lung este fisierul .tra, cu atat mai mare e acuratetea.
- Cresterea numarului de presceptroni - creste acuratetea, dar la un moment dat ajunge la saturarea, dupa care incepe sa scade. Creste folosirea memoriei si afecteaza viteza executiei (nu cu foarte mult, asta e treaba lui HR 😊)
- Numarul de biti la HR - creste acuratetea, la un moment dat ajunge la saturarea, dupa care nu afecteaza acuratetea cu mult, dar creste memorie necesara pentru procesarea predictiilor, ca si timpul de executie.

Concluzii

Metoda de predictie perceptron - e interesanta ca idee. La programe mari arata rezultate excelente. Problema vine cu programe cu numarul de salturi mic, ce este putin probabil, deci putem nu lua in considerare problema asta. Ca dezavantajul major, eu pot evidentia faptul ca noi avem nevoie de memorie aditionala, pentru a salva isotia salturilor si perceptroni, aferente programului curent.

