

Zadanie domowe

Jakub Piecuch
Seminarium ASK

23 października 2017

Zadanie 1. Załóżmy, że w tej chwili pewna instrukcja skoku warunkowego ma następującą historię wykonania, gdzie wcześniejsze skoki znajdują się z lewej strony (0 oznacza że skok się nie wykonał, 1 oznacza że się wykonał):

1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0

Weźmy predyktor bimodalny z osobnym 2-bitowym licznikiem nasyceniowym zainicjowanym do stanu $10_{(2)}$ przypisanym do każdej instrukcji skoku warunkowego.

- Jaki będzie przewidywany następny rezultat tego skoku?
- Ile razy do tej pory pomylił się predyktor?

Weźmy predyktor korelacyjny z 2-bitowym rejestrem historii (BHR – *Branch History Register*) lokalnym dla każdej instrukcji skoku. Wartości w tym rejestrze używamy jako indeks do tablicy historii wzorców (PHT – *Pattern History Table*), również lokalnej dla każdej instrukcji skoku warunkowego, zawierającej $2^2 = 4$ 2-bitowe liczniki nasyceniowe. Rejestr historii na początku jest wyzerowany, a liczniki są zainicjowane do wartości $10_{(2)}$.

- Ile razy pomylił się ten predyktor?
- Załóżmy, że wyniki skoku w zadaniu powtarzają się w nieskończoność z okresem 8 skoków. Czy zwiększenie liczby bitów w rejestrze historii (i odpowiednie zwiększenie PHT) wpłynie na skuteczność predyktora?

Zadanie 2. Oto (błędna) implementacja popularnego programu FizzBuzz:

```
for (int i = 1; i <= 100; ++i) {
    if (i % 3 == 0 && i % 5 == 0)
        printf("FizzBuzz\n");
    if (i % 3 == 0)
        printf("Fizz\n");
    if (i % 5 == 0)
        printf("Buzz\n");
}
```

Wskaż po jednym przykładzie korelacji lokalnej i globalnej. Dla każdego skoku wewnątrz pętli napisz, ile bitów lokalnej historii potrzebujemy aby móc (po pewnym czasie) bezbłędnie przepowiadać jego kierunek.

Zadanie 3. Oto algorytm z prezentacji używany do aktualizacji wag w perceptronie:

```
if sign(y) != t or |y| ≤ θ
  for i := 0 to n do
    wi := wi + t * xi
```

Algorytmu tego używamy po poznaniu rzeczywistego rezultatu skoku (nazwijmy go s), którego rezultat wcześniej przewidywaliśmy.

Definicje:

- $\text{BHR}[i]$ – i -ty bit w rejestrze historii tuż przed poznaniem prawdziwego rezultatu skoku s ($\text{BHR}[1]$ – najpóźniejszy skok; $\text{BHR}[n]$ – najwcześniejszy skok)

- $$x_i = \begin{cases} 1 & i = 0 \\ 1 & i > 0, \text{BHR}[i] = 1 \\ -1 & i > 0, \text{BHR}[i] = 0 \end{cases}$$

- $y = \sum_{i=0}^n w_i * x_i$

- $$\text{sign}(y) = \begin{cases} 1 & y \geq 0 \\ -1 & y < 0 \end{cases}$$

- $$t = \begin{cases} 1 & \text{jeśli skok } s \text{ wykonał się w rzeczywistości} \\ -1 & \text{w p.p.} \end{cases}$$

- n – długość (w bitach) rejestru historii (BHR)

- θ – pewna dodatnia stała

Zaproponuj równoważny algorytm, który nie wykorzystuje pętli oraz mnożenia.
(Wskazówka: co można zrównoleglić?)