

Predição de Saltos usando Perceptrons

Jiménez, Daniel A., and Calvin Lin. "Dynamic branch prediction with perceptrons." High-Performance Computer Architecture, 2001. HPCA. The Seventh International Symposium on. IEEE, 2001.

Eric Eduardo Bunese





Perceptron

Algoritmo de aprendizado supervisionado - *Frank Rosenblatt, 1957*

Tomada de decisão com base em classificador binário

Online Learning

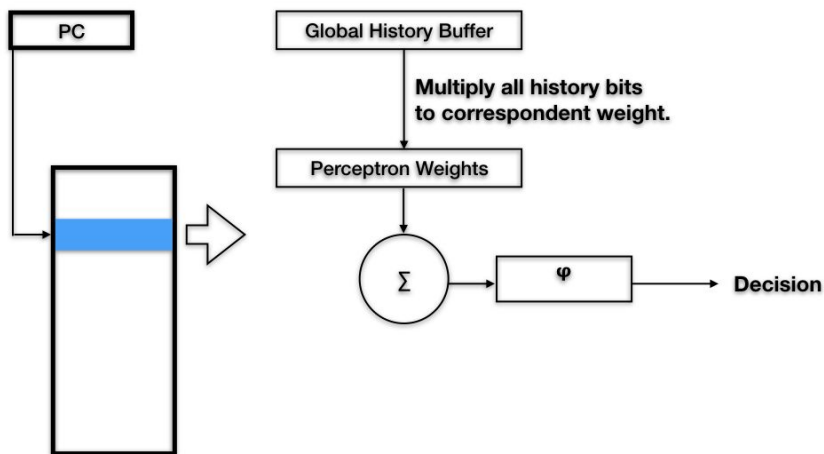
Classificador Binário:

-> Uma entrada E pertence a uma classe C ?

Online Learning:

-> Processa elementos um de cada vez e realimenta seu conhecimento

Perceptron



1. Fazer um hash do endereço da instrução, para um índice i da tabela de perceptrons.
2. Buscar o perceptron i
3. Computar o produto escalar dos pesos do perceptron com o histórico global.
4. Se o resultado for negativo: Not Taken, caso contrário Taken.
5. Ao descobrir o resultado real do branch, atualizar os pesos ao perceptron utilizado.



Perceptron

Exemplo: Histórico global: [1, -1, -1, -1] (Acertou, Errou, Errou, Errou)

O endereço da instrução obteve o perceptron P, da tabela com N perceptrons.

Os pesos do perceptron P são [1, 2, 3, 4]

O resultado computado será:

$$1*1 + -1*2 + -1*3 + -1*4 = 1 + (-2) + (-3) + (-4) = -8 \rightarrow \text{Resultado} = \text{NOT TAKEN.}$$

Função de treinamento:

if (sign(yout) != t or yout <= THETA)

for (i=0;i<n;++i)

$$w[i] = w[i] + t*X[i]$$



Parâmetros de simulação

Não há parâmetros adicionais.

```
./orcs -t <arquivo trace>
```

Configurações adicionais nos #defines das estruturas

-btb.hpp

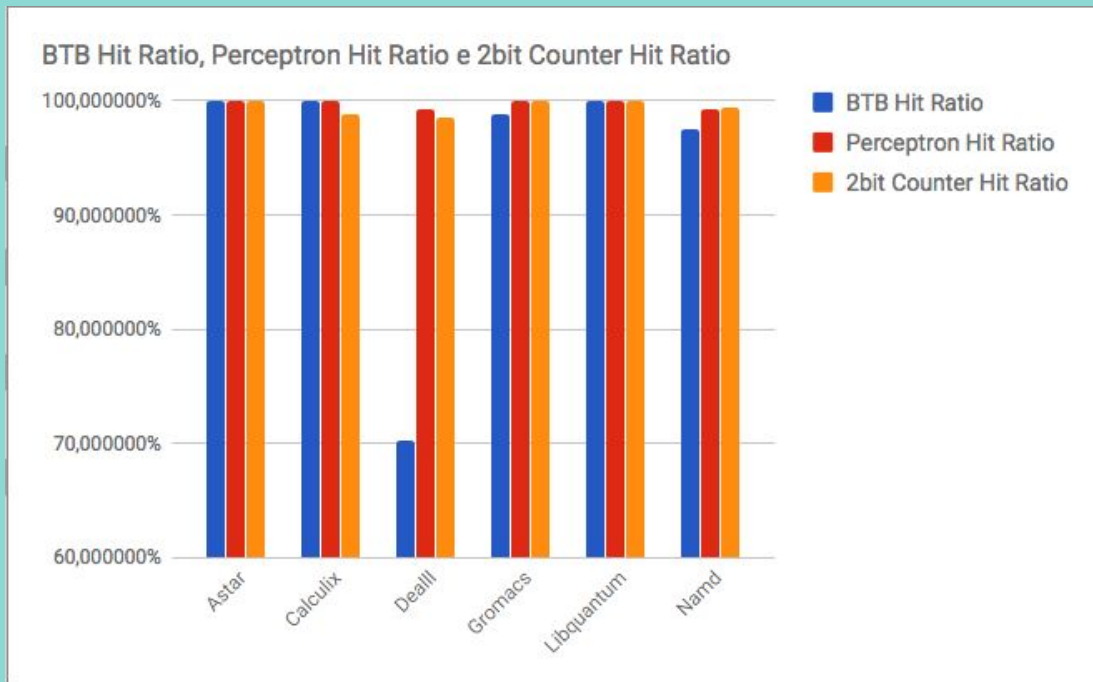
-twobitcounter.hpp

-perceptron.hpp

Resultados Obtidos

	Astar	Calculix	Dealll	Gromacs	Libquantum	Namd
BTB Hit Ratio	99,992047%	99,955926%	70,203210%	98,799938%	99,999923%	97,492746%
Perceptron Hit Ratio	99,995277%	99,914677%	99,313897%	99,979054%	100,000000%	99,272252%
2bit Counter Hit Ratio	99,990924%	98,757171%	98,470774%	99,940370%	99,999991%	99,419450%
BTB Queues	30.831.777,00	3.562.157,00	21.723.125,00	4.730.589,00	34.952.024,00	29.084.208,00
BTB Hits	30.829.325,00	3.560.587,00	15.250.331,00	4.673.819,00	34.951.997,00	28.354.993,00
Perceptron Queues	30.829.324,00	3.560.586,00	15.250.330,00	4.673.818,00	34.951.996,00	28.354.992,00
Perceptron Good Guesses	30.827.868,00	3.557.548,00	15.145.697,00	4.672.839,00	34.951.996,00	28.148.639,00
2bit Queues	30.829.325,00	3.560.587,00	15.250.331,00	4.673.819,00	34.951.997,00	28.354.993,00
2bit Good Guesses	30.826.527,00	3.516.335,00	15.017.119,00	4.671.032,00	34.951.994,00	28.190.378,00
Total Fetched Instructions	200.000.000,00	200.000.000,00	200.000.000,00	200.000.000,00	200.000.000,00	200.000.000,00
Total Number of Cyles	200.032.782,00	200.038.416,00	252.620.920,00	200.463.485,00	200.001.720,00	207.486.047,00
Cycles Used / Fetched Instrucs	1,00016391	1,00019208	1,2631046	1,002317425	1,0000086	1,037430235

Resultados Obtidos



Resultados:

A tabela com 64 Perceptrons utilizada, junto de um histórico de 62 entradas (Jiménez et. al.) provou ser mais eficaz que o 2bit counter global para os traces executados.



Referência

- Jiménez, Daniel A., and Calvin Lin. "Dynamic branch prediction with perceptrons." High-Performance Computer Architecture, 2001. HPCA. The Seventh International Symposium on. IEEE, 2001.