퍼셉트론을 이용하는 다중 분기 예측법에 대한 연구

*이종복
한성대학교 정보통신공학과
e-mail : jblee@hansung.ac.kr

Multiple Branch Prediction with Perceptrons

*Jongbok Lee
Dept. of Information and Communications
Hansung University

Abstract

This paper presents a multiple branch predictor with perceptrons. We describe our design and evaluate it with the SPEC 2000 benchmarks. Our predictor achieves increased accuracy than the previous multiple branch predictors.

I. 서론

고성능 수퍼스칼라 마이크로 프로세서의 명령어 대역폭을 증가시키는 방편으로 다중 분기 예측법 또는 트레이스 캐쉬와 같은 방법이 널리 연구되고 있다. 본 논문에서는 신경망 분야에서 활용되는 퍼셉트론 방식을 마이크로 프로세서의 다중 분기 예측법에 적용하였다. 이것을 위하여 SPEC 2000 벤치마크 프로그램을 대상으로 하여 모의실험을 수행하여, 기존의 2 단계 적응형 다중 분기 예측법과 비교하여 그 정확도의 우수성을 보였다.

II. 본론

2.1 퍼셉트론 분기 예측법

퍼셉트론이란 입력 값들을 가중치와 결합하여 출력을 산출하며, 학습 기능을 갖는 신경망이다. 퍼셉트론 분기 예측법은 이러한 신경망 회로에서 이용하는 학습 방식을 분기 예측기법에 적용한 것이다 [1]. 이것을 위하여 길이 N의 분기 히스토리에 N 개의 분기어가 각각 분기한 결과를 1로, 분기하지 않은 결과를 -1로 기록한다. 이렇게 하여 얻은 분기어 N 개의 분기 행태를, 마찬가지로 1 또는 -1로 나타내는 현재 분기어의 결과와 곱하여 길이 N인 가중치 벡터를 생성한다. 따라서, 만일 임의의 분기 명령어가 계속 분기를 한다면 가중치 벡터의 필드값이 점점 커지고, 분기를 하지 않는다면 그 값이 점점 작아지게 된다. 분기 히스토리 레지스터와 가중치 벡터가 마련된 후에, 다음의 분기 명령어에 대한 예측을 수행하고자 할 때, 현재의 분기 히스토리 레지스터와 가중치 벡터에 대하여 내적을 계산하여, 그 연산 결과가 양이면 분기할 것으로 예측하고, 음이면 분기하지 않는 것으로 예측한다. 추후에 분기 여부가 실제로 판명되었을 때, 가중치의 상관 관계와 일치하면 가중치를 증가시키고 일치하지 않으면 감소시키는 알고리즘으로 학습시킴으로써 퍼셉트론의 가중치 벡터를 계속적으로 업데이트한다.

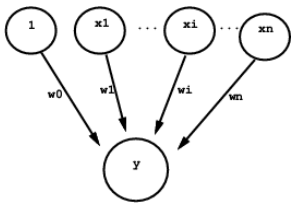

그림 1 퍼셉트론 모델

그림 1은 퍼셉트론의 그래픽 모델이다. 퍼셉트론은 가중치를 요소로 갖는 벡터로 표현되며, 가중치는 양 또는 음의 정수로 나타낸다. 출력은 가중치 벡터 $w_{0..N}$과 입력벡터 $x_{0..N}$의 내적 (dot product)이며, x_0는 항상 1로 설정하여 바이어스 입력으로 공급한다. 따라서, 이전 분기 결과와의 상관 관계를 학습하기 이전에 바이어스 가중치 w_0는 히스토리와 상관없이 항상 그 분기어의 바이어스를 학습하게 된다. 퍼셉트론의 출력 y는 수식 1과 같이 표현된다. 퍼셉트론에 대한 입력은 양극성이며 x_i가 -1이면 분기가 일어나지 않은 것이고, 1이면 분기가 일어난 것이다. 음의 출력일 때는 분기를 하지 않는 것으로 예측하며, 양의 출력일 때는 분기를 하는 것으로 예측한다. 프로세서가 분기어를 만나면, 분기어의 어드레스를 이용하여 0 부터 N-1 사이의 인덱스 i를 생성하여 퍼셉트론 테이블을 접근한다. 이렇게 하여 테이블에서 i번째 퍼셉트론을 인출하여 가중치 벡터 $P_{0..N}$을 얻은 후에, 가중치 벡터와 전역 히스토리 레지스터와의 내적으로

$$y = w_o + \sum_{i=1}^{n} x_i w_i$$

수식 1 퍼셉트론의 출력

출력 y를 계산하여 그 부호에 따라 다음 명령어의 분기 여부를 예측한다. 추후에 분기어의 실제 분기 결과가 알려지면, 그 결과와 y 값을 가지고 학습 알고리즘을 이용하여 벡터 P의 가중치를 수정한다. 그리고 벡터 P를 테이블의 i 번째 항목에 기록하여 그 결과를 반영한다.

2.2 퍼셉트론을 이용한 다중 분기 예측법

기존의 다중 분기 예측법은 2 단계 적응형 분기 예측법을 응용 및 확장한 것이다 [2]. 퍼셉트론 분기 예측은 일정한 길이의 분기 결과를 히스토리 레지스터에 기록하고 임의의 분기 어드레스를 가지고 가중치 벡터로 구성된 테이블을 접근하여 2 단계로 예측을 수행하므로, 전역 히스토리 방식에 의한 2 단계 적응형 예측법 및 다중 분기 예측법을 동일하게 적용할 수 있다. 그 방법을 살펴 보면, , M 개의 다중 분기 예측을 수행하기 위하여, 첫번째, 두번째, ..., M 번째 분기 명령어에 대하여, 각각 히스토리 레지스터의 k 비트, k-1 비트, ..., k-M+1 비트로 가중치 벡터 테이블을 각각 인덱스하여 각 분기 명령어를 예측하는 것이다. 본 논문의 퍼셉트론을 이용한 분기 예측 모의실험에서 분기 히스토리 레지스터의 길이는 8로 하였고, 가중치 벡터로 구성되는 테이블의 항목 수는 4096개로 하였다.

III. 모의실험 결과

표 1에 SPEC 2000 벤치마크에 대하여 측정한 퍼셉트론 방식의 다중 분기 예측도를 나타내었다. 이 때, 2 개 및 3 개의 연속적인 블럭에 대한 예측도를 표시하였다. 2 차의 분기 예측에서, 대부분의 예측도는 2-블럭에 속하며, 1-블럭의 예측도는 crafty에서 최대 10.2 %를 넘지 않았다. 3 차의 분기 예측에서도 마찬가지로 대부분이 3-블럭에 속하며, 2-블럭과 1-블럭은 perlbmk에서 10.6 %를 초과하지 않았다. 한편, 14 비트의 분기 히스토리 레지스터와 16 K 개의 패턴 히스토리 테이블을 갖는 2 단계 적응형 분기 예측을 10 개의 정수형 벤치마크에 대하여 모의실험하여 분기 예측 정확도를 비교한 결과, 퍼셉트론 방식이 평균 1.0 %만큼 우수한 정확도를 나타내었다.

IV. 결론 및 향후 연구 방향

높은 명령어 대역폭이 요구되는 고성능 수퍼스칼라 마이크로 프로세서에서는 보다 정밀한 다중 분기 예측기가 필요하다. 본 논문에서는 퍼셉트론을 적용하여, 기존의 방식보다 우수한 성능의 다중 분기 예측기를 제안하였다.

벤치	2 차		3 차		
마크	2블럭	1블럭	3블럭	2블럭	1블럭
bzip2	94.80	2.21	72.17	8.06	2.02
crafty	79.27	10.20	67.88	7.89	0.30
gap	82.87	7.52	65.59	9.57	2.51
gcc	82.77	7.30	73.34	7.09	1.25
mcf	92.67	3.58	88.70	4.76	0.07
perlbmk	87.33	5.61	60.81	10.63	2.21
twolf	88.08	5.79	83.82	4.45	0.91
vortex	80.87	8.24	70.81	8.45	1.88
vpr	89.16	5.05	82.70	6.24	0.52
ammp	88.52	3.85	82.41	5.47	0.64
apsi	92.33	3.38	64.19	10.37	2.66
art	68.54	11.00	48.20	12.93	0.37
equake	63.76	12.56	42.61	10.16	0.29
mesa	91.50	5.88	87.95	5.47	0.13
mgrid	92.95	3.15	83.56	4.57	1.15
swim	70.44	10.40	51.25	26.76	0.90
wupwise	82.75	7.39	70.20	8.41	2.24

표 1 다중 분기 예측 정확도 (%)

참고문헌

[1] D. A. Jimenez and C. Lin, "Dynamic Branch Prediction with Perceptrons," International Symposium on High Performance Computer Architeture, 2001. pp. 197-206.

[2] T-Y Yeh, D. T. Marr, and Y. N. Patt, "Increasing the Instruction Fetch Rate via Multiple Branch Prediction and a Branch Address Cache," International Conference on Supercomputing, Jul. 1993, pp.67-76.