순환 신경망의 하드웨어 구현

김정욱 오종훈
포항공과대학교 물리학과
{cwkim, jhoh}@postech.ac.kr

Hardware Implemetation of Recurrent Neural Network

Cheong-Wook Kim Jong-Hoon Oh
Dept. of Physics, Pohang University of Science and Technology

요 약

최근에는 순환 신경망의 생성모델이 비교사 학습에 관련하여 활발히 연구되고 있다. 이러한 형태의 신경망은 형태 추출이나 인식에 효과적으로 사용될 수 있는 반면 반복 loop를 사용하므로 대단히 많은 계산이 필요하다. 본 논문에서는 Oh와 Seung에 의해 제안된 상향전파(Up-propagation) network이라는 순환 신경망을 FPGA를 이용해서 구현하였다. 단층 신경망은 9개의 상층 neuron과 256개의 하층 neuron으로 구성되 있으며 4만 게이트의 FPGA 하나로 효과적으로 구현할 수 있다. pipeline된 곱셈기로 계산 속도를 향상시켰고 sigmoid 전달함수는 유한 정밀도의 2차 다항식으로 근사될 수 있다. 구현된 하드웨어는 hand-written 숫자 영상인 USPS data를 재생하는데 사용되었으며 좋은 결과를 얻었다.

1. 서론

인공 신경망은 신호의 전달 방법에 따라 전방향 공급(Feed-Forward) 신경망과 순환(Recurrent) 신경망으로 나뉜다. 퍼셉트론(Perceptron) 과 같은 전방향 공급 신경망에서는 입력신호는 되먹임 없이 한 방향으로 전달되어 출력을 낸다. 이러한 형태의 전방향 공급 신경망은 단순한 입출력 관계를 가지므로 쉽게 하드웨어로 구현할 수 있다.[1-4] 되먹임 연결을 갖는 순환 신경망은 전방향 공급 신경망에 비해 다양한 입출력 관계를 표현할 수 있으므로, 복잡한 문제를 해결할 수 있다. 그러나 되먹임 연결 루프를 하드웨어로 구현하기가 어려워 단순한 모델 외에는 실제 문제에 적용할 수 있는 신경망 구현은 거의 이루어지지 않았다.[5,6]

인공 신경망은 주어진 데이터로부터 입출력 관계를 배우며 이를 학습이라고 한다. 학습은 크게 교사가 있는 학습(Supervised learning)과 교사가 없는 학습(Unsupervised learning)이 있다. 교사가 있는 학습에서는 입력에 대한 정답이 존재하여 출력과의 비교를 통해 규칙을 학습한다. 대표적으로는 역전파(Back-propagation) 알고리듬이 있다. 이에 반해 교사가 없는 학습에서는 특정한 예제로부터 학습하는 것이 아니라 입력데이터의 상관관계로부터 규칙을 배운다.

교사가 없는 학습 방법으로서 최근 생성 모형(Generative model)을 구현한 신경망 구조가 많은 각광을 받고있다. Oh와 Seung[7]에 의해 제안된 상향 전파 알고리듬은 간단한 순환 신경망 구조와 오류 역전파 방법을 써서 입력 데이터의 비선형 주성분 분석(Principal Component Analysis)을 구현한 간단하지만 강력한 신경망이다. 상향 전파 알고리듬은 이미지 데이터의 특징 추출이나 그 특징을 이용하여 패턴 인식을 하거나 이미지 데이터 압축하는 등의 응용에서 좋은 결과를 얻을 수 있었다. 그러나 많은 순환 신경망 학습 알고리듬이 그렇듯이 상향 전파 신경망에서는 학습 단계에서 뿐만 아니라 학습된 가중치를 이용한 연산 단계에서도 반복 루프를 사용하게 된다. 따라서 다층 퍼셉트론과 같은 전방향 공급 신경망에 비해 연산 단계에서 훨씬 더 많은 계산이 필요로 하게 되므로 일반 컴퓨터에서 소프트웨어로 구현하면 많은 계산시간을 요구한다. 그러나 대개의 영상신호 처리나 음성데이터 등에 응용할 때는 실시간으로 데이터를 처리하는 것이 필수적이다. 따라서 하드웨어로 효율적으로 구현을 하여 실시간 연산이 가능하게 한다면 여러 분야에서 응용 가능성이 커질 것이다.

최근에는 재구성이 가능한 대용량 하드웨어인 FPGA(Field Programmable Gate Array)가 상용화 되고 있다. 원래 신경망은 비선형 전달함수를 필요로 하며 연결 가중치도 실수값으로 주어지므로 아날로그 회로로 구현함이 원칙이나 만약 이를 최소한의 디지털 연산으로 근사할 수 있다면 비교적 작고 경제적인 하드웨어로 구현이 가능할 것이다. 이에 본 논문에서는 상향 전파 알고리듬을 이용한 단층 순환 신경망을 FPGA를 이용해서 구현해 보고자 한다.

2. 상향 전파(Up-propagation) 알고리듬[7]

상향 전파 알고리듬은 다층 신경망으로 확장이 용이하나 단층 신경망으로도 좋은 결과를 얻을 수 있었으므로 그림 1 과 같은 단층 순환 신경망의 상향 전파 알고리듬에 대해서 알아본다.

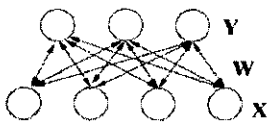

그림 1 단층 순환 신경망

X, Y 는 각각 입력층과 상층 뉴런의 상태를 나타내는 벡터이고 W 는 상층에서 입력층으로 시냅스 연결을 나타내는 가중치 행렬이다. 가중치의 학습은 미리 소프트웨어에서 수행되므로 하드웨어로 구현될 부분을 요약하면 다음과 같다.

1 단계: 가설을 생성한다.

$$x_i = g(h_i) = g(\sum_j w_{ij} y_j) \quad (1)$$

, $g(x) = 1/(1+e^{-x})$ 이다.

2 단계: 입력된 패턴과 생성된 가설의 오차를 측정한다.

$$\delta_i = g'(x_i)(d_i - x_i) \quad (2)$$

3 단계: 숨은 변수를 다음 식에 따라 보정 한다.

$$y_j^{new} = y_j^{old} + \eta\, dy_j = y_j^{old} + \eta \sum_i w_{ij}\delta_i \quad (3)$$

, η 는 학습률(learning rate)이다.

4 단계: 주어진 반복 횟수만큼 2 단계로 가서 과정을 되풀이한다.

x_i, y_j, w_{ij} 는 각각 X, Y, W 의 성분을 나타낸다.

3. 신경망 하드웨어의 구조

구현된 단층 순환 신경망의 구조는 하층(입력층) 뉴런이 하나, 상층(숨은 층) 뉴런이 9개로 되어 있다. 본 논문에서 USPS 영상 데이터의 재생 시 사용될 신경망의 구조가 입력층 뉴런 256개 숨은 층 뉴런 9개 구조이다. 그러므로 하나의 입력층 뉴런을 반복해서 기동함으로써 전체적으로 256x9의 네트웍을 구현했다. 그림 2 는 전체 데이터 흐름을 나타낸 블록 다이어그램이다. 상향 전파 부분(Up-prop block)은 상층 뉴런의 가설과 입력의 오차인 식 (2) 을 계산해서 상층 뉴런이 값을 수정 할 수 있도록 한다. 식(1) 과 (3) 의 연산에서 매 루프 당 많은 곱셈 계산이 필요하므로 곱셈기는 빠른 연산 속도를 요구한다. 그래서 룩업 테이블을 이용해 3 클럭 pipeline 된 곱셈기를 구현했다.

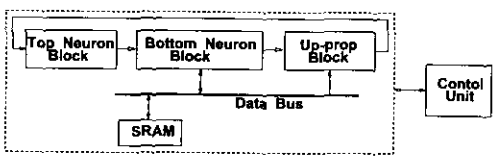

그림 2 Block Diagram

소프트웨어에서는 모든 연산이 실수로 이루어지므로 실제 알고리듬의 구현이 어렵다. 하드웨어 구현에서는 실수 연산과 비슷한 결과가 나올 수 있도록 적절한 정밀도를 갖도록 해야 한다. 그래서 시뮬레이션으로 얻은 8 비트 정밀도의 데이터와 1/8 학습률을 사용하였다.

신경망 하드웨어에서 비 선형 함수의 구현은 어려운 문제이다. 기존의 하드웨어들[2,3]은 롬(ROM)으로 활성화 함수를 구현하여 시스템의 구조가 커지는 단점이 있다. 본 논문에서는 FPGA가 롬을 구현 할 수 있다는 특성을 이용, 활성화 함수를 구현하여 이러한 문제를 해결하였다. Kwan[8]이 제안한 2차 활성화 함수로 시그모이드 함수를 근사한다.

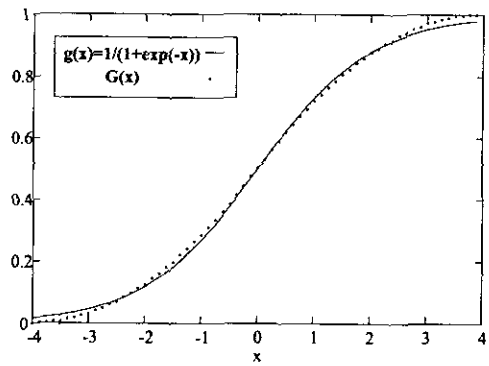

그림 3 Sigmoid 함수의 근사

시그모이드 함수 포화 지역의 범위 ±4에 맞춰 2차 활성화 함수의 $L = \pm 2^{12}$로 하여 근사 함수를 다음과 같이 구할 수 있다.

$$G(x) = \begin{cases} 255 & \text{for} \quad 2^{12} \le x, \\ 2^{-7}\frac{x}{2^5}(2^8 - \frac{x}{2^5}) + 2^7 & \text{for} \quad 0 \le x < 2^{12}, \\ 2^{-7}\frac{x}{2^5}(2^8 + \frac{x}{2^5}) + 2^7 & \text{for} \quad -2^{12} < x < 0, \\ 0 & \text{for} \quad x \le -2^{12}. \end{cases} \quad (4)$$

그림 3 에서 보듯이 포화지역 근방을 제외한 부분에서 G(x)는 시그모이드 함수, g(x)를 잘 근사함을 알 수 있다. 위 근사식 G(x)는 256 byte x 8 의 롬으로의 구현할 수 있다. 그러므로 FPGA로 시그모이드 함수를 간단히 구현할 수 있다. 본 논문에서 사용된 FPGA는 FLEX10K40[9]이며 약 4만 게이트를 실현할 수 있다.

5. 응 용

USPS 데이터 베이스를 이용하여 하드웨어 신경망을 테스트하였다. USPS 데이터는 16x16 픽셀로 구성된 손으로 씌여 진 0 ~ 9 까지의 숫자 영상이며 각 픽셀은 0 ~ 255의 정수로 되어 있다. 우선 예제 영상으로 학습을 시켜서 특징(feature) 벡터, 즉 가중치를 추출한다. 학습에 사용하지 않은 새로운 영상이 주어지면 추출된 특징 벡터로 주어진 영상을 재생한다. Oh와 Seung[7]은 9개의 상층 뉴런으로 좋은 결과를 얻었다. 그러므로 신경망의 전체 구조는 256X9가 된다. 10개의 숫자 중 '2' 에 대해

적용하였다. 신경망은 보통 20번 정도 루프를 돌고 나면 재생 된 데이터의 값은 수렴을 하게 된다. 그림 4(a) 는 목적 영상이고 그림 4(b) 는 각각 4(위 왼쪽), 8(위 오른쪽), 16(아래 왼쪽), 32(아래 오른쪽)번의 루프를 돌려 얻은 영상이다.

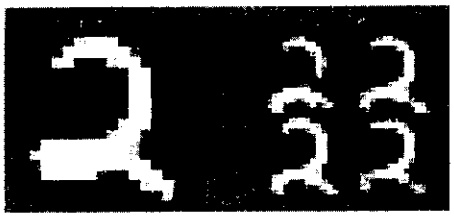

그림 4 Iteration 횟수에 따른 패턴의 변화

그림 5(a) 는 학습에 사용되지 않은 목적 영상이며 그림 5(c) 는 신경망 하드웨어로 재생된 영상이다. 그림 5(b) 는 비교를 위해 컴퓨터 실수연산으로 재생된 영상이다.

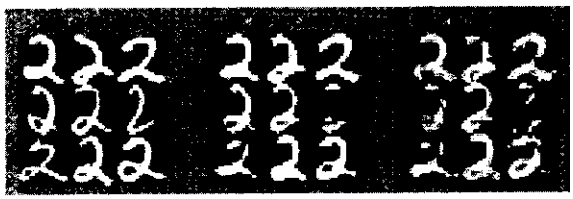

그림 5 Software와 Hardware의 결과 비교

정확성을 측정하기 위해 오차율을 다음 식과 같이 정의를 한다.

$$Error(\%) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{256}|d_i^j - x_i^j|}{255\times 256\times N}\times 100 \qquad (14)$$

여기서 N은 사용된 패턴의 수이다. 소프트웨어의 경우는 10.37%, 그리고 하드웨어의 경우는 11.68%가 나왔다. 하드웨어 신경망은 실수 연산의 소프트 웨어 신경망과 비슷한 성능을 보여 준다.

6. 결 론

상향 전파 알고리듬을 이용한 단층 순환 신경망을 하나의 칩(one-chip) 하드웨어로 구현하였다. 9x1 하드웨어 구조로 9x256의 전체 네트웍을 실현하였다. 모든 연산은 pipeline 되어 있어서 직렬 연산의 부하를 효과적으로 줄일 수 있었다.

구현 된 신경망으로 hand-written 숫자 영상인 USPS 데이터를 재생하는데 적용하였다. 컴퓨터 실수 연산을 이용한 것과 전체 8 비트의 정밀도를 갖는 신경망 하드웨어의 결과가 거의 비슷하였다.

최근에는 200만 게이트가 넘는 FPGA가 보급되고 있어 이를 이용하여 전체 구조를 더 늘릴 수 있고, 그럼으로써 더 빠른 병렬 연산이 가능해져 실제 문제에 응용이 가능할 것이다.

7. 참고 문헌

[1] Zhu J., G. J. Milne and B. K. Gunther, Towards an FPGA based reconfigurable computing environment for neural network implementations, 9th International Conference on Artificial Neural Networks, 2, 661-66(1999).

[2] Cox, C. E. and W. E. Blanz, GANGLION - A Fast Field-Programmable Gate Array Implementation of a Connectionist Classifier. IEEE Journal of Solid-State Circuits, 27, 228-299(1992).

[3] Botros, N. M. and M. Abdul-Aziz, Hardware implementation of an artificial neural network using field programmable gate arrays(FPGA's), IEEE Transactions on Industrial Electronics, 41, 665-667(1993).

[4] Suzuki, D. and O. Hammami., SOM on multi-FPGA ISA board-hardware aspects, The 6th IEEE International Conference on Electronics, Circuits and Systems, 3, 1401-1405(1999).

[5] Gschwind, M., V. Salapura and O. Maischberger., A Generic Building Block For Hopfield Neural Networks With On-chip Learning, IEEE International Symposium on Circuits and Systems, Supplement, 49-52(1996).

[6] Abramson, D. K., Smith, P. Logothetis and D. Duke., FPGA based implementation of a Hopfield neural network for solving constraint satisfaction problems, Proceedings. 24th Euromicro Conference, 2 , 688-693(1998).

[7] Oh, J. H. and H. S. Seung, Learning Generative Models by Up-Propagation Algorithm, Advances in Neural Information Processing Systems, 10, 605-611(1998).

[8] Kwan, H. K., Simple Sigmoid-like Activation Function Suitable for Digital Hardware Implementation, Electronics Letters, 28, 1379-1380(1992).

[9] *Altera Device Data Book* (Altera Corp., 1999).