

FPGA에 의한 블록기반 신경망의 설계

장정두, 공성곤
승실대학교 전기공학과

Hardware Design of Block-based Neural Networks Using FPGA

Jung-Doo Jang and Seong-Gon Kong
Soongsil University, Department of Electrical Engineering

Abstract - 본 논문에서는 BNN, 블록기반 신경망 모델을 재구성가능 하드웨어(FPGA)로 설계한다. 블록기반 신경망은 재구성가능 하드웨어에 의하여 구현이 용이하고 구조 및 가중치의 최적화에 진화 알고리즘을 적용시킬 수 있다. 블록기반 신경망의 구조와 가중치를 표현하는 바이너리 스트리밍을 오프라인으로 진화시킨 후, 재구성가능 하드웨어로 구현한다. FPGA로 구현된 블록기반 신경망의 성능을 확인하기 위하여 간단한 성능시험에 사용되는 대표적인 패턴들을 사용하여 블록기반 신경망의 패턴분류 성능을 알아본다.

1. 서론

최근 하드웨어 진화에 대한 관심이 높아지고 있다. 하드웨어 진화에 사용되는 방법은 크게 두 가지로 나뉜다. 첫째는 하드웨어 기술언어(VHDL)를 이용하여 함수 레벨에서 오프라인으로 설계하는 것이고, 둘째는 적절한 FPGA 모듈과 해당 소프트웨어를 이용하여 게이트 레벨에서 온라인으로 설계하는 것이다[1].

블록기반 신경망은 FPGA와 같은 재구성 가능한 디지털 하드웨어로의 적용을 용이하게 하기 위하여 최소단위인 블록들의 2차원 배열로서 전체 신경망을 구성한다. 이러한 구조는 뉴런간 임의의 연결을 허용하는 신경망 모델에 비하여 구조 최적화가 용이한 특징을 가진다. 또한 신경망의 구조와 정수로 제한된 가중치를 고정된 길이를 갖는 비트 스트리밍으로 표현하며, 최적화 방법으로 진화 연산의 한 분야인 전역 최적화 특성을 가지는 유전자 알고리즘을 사용한다. 최적화 방법으로 사용한 유전자 알고리즘에서는 구조와 가중치의 동시 최적화가 가능하도록 개체를 구성한다. 또한 블록기반 신경망의 최적화를 위하여 새로운 유전 연산자를 사용한다.

본 논문에서는 유전자 알고리즘을 사용한 BNN의 진화는 오프라인으로, 진화가 끝난 후 최적화된 개체를 사용한 하드웨어 BNN의 성능 평가는 VHDL을 사용한 함수 레벨의 설계를 통해서 한다.

본 논문의 구성은 다음과 같다. 본문에서 블록기반 신경망과 재구성 가능 하드웨어란 무엇인가에 대해서 설명한다. 그리고, 실험은 진화 후 획득한 각 패턴의 최적화된 비트 스트리밍을 사용해 VHDL을 이용한 하드웨어 설계를 하고 설계 과정과 패턴 분류를 통해서 본 실험 결과를 보인다.

2. 블록기반 신경망

2.1 블록기반 신경망의 구조

블록기반 신경망은 4개의 가변 입출력을 가지는 블록을 기본 구성요소로 하고 블록들의 2차원 배열 형태로 이루어진 새로운 신경망 모델이다[2].

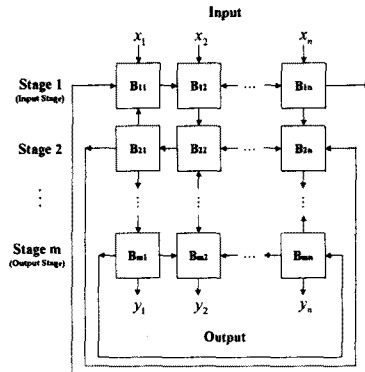


그림 1. 블록기반 진화신경망의 구조

그림 1은 기본 블록들로 이루어진 $m \times n$ 블록기반 신경망의 구성을 나타낸다. 블록기반 신경망의 입력은 $-1 \leq x_i \leq 1$ 의 범위를 가진다. 블록간 정보의 흐름을 나타내는 화살표와 같이 중간 stage에 속하는 블록들은 인접한 상하좌우 4개의 블록들과 직접 연결되어 정보를 주고받는다. 이러한 연결 방법에 의하여 블록기반 신경망은 내부에서의 신호 전달에 있어서 전방향적인 성질, 재귀적인 성질, 동일 stage의 블록간 신호의 교환, 그리고 블록의 자기 궤환 형태의 연결을 허용하는 특징을 가진다.

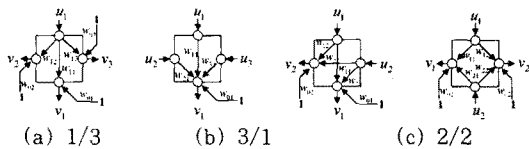


그림 2. 기본 블록의 내부구조

그림 2는 블록기반 신경망을 구성하는 각 블록 B_{mn} 의 기본 블록을 나타낸다. 각 블록은 '입력/출력'의 개수에 따라서 3가지 형태로 구분한다. B_{mn} 는 내부에 4개의 입출력 뉴런이 존재하는 전방향 단층 신경망과 같은 형태로 구성된다.

2.2 블록기반 신경망의 최적화

재구성 가능 디지털 하드웨어로의 구현을 위하여 디자인된 블록기반 신경망은 비트 스트리밍으로 표현이 가능하며 최적화 알고리즘으로는 비트 스트리밍을 최적화 하는데 적합한 알고리즘인 유전자 알고리즘을 사용한다.

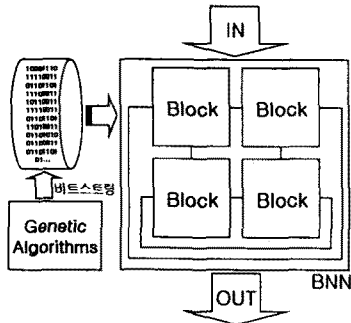


그림 4. 유전자 알고리즘의 최적화 과정

그림 4는 본 논문에서 사용한 유전자 알고리즘의 최적화 과정을 나타낸다. 본 논문에서 사용한 유전자 알고리즘의 최적화 진행과정에서는 새로운 교배 연산자, 개선한 돌연변이 연산자, 그리고 추가적으로 내부복제 연산자를 적용한다.

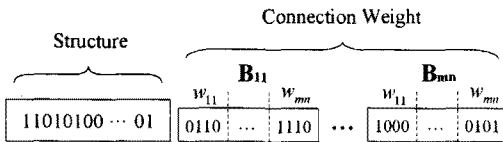


그림 5. 인코딩된 바이너리 스트림

그림 5는 블록기반 신경망의 구조와 가중치 정보를 포함하는 개체의 구성 방법을 나타낸다. 구조를 나타내는 비트 스트림은 개체의 시작 부분에 위치시키고, 가중치를 나타내는 비트 스트림은 구조를 나타내는 비트 스트림 이후에 연속하여 배치한다.

유전연산을 거친 개체들 중 일부를 다음 세대의 부모 개체로 선택하기 위해서는 선택 압력과 샘플링 방법을 결정해야 한다. 본 논문에서는 선택 압력으로 변형된 분열압력(Disruptive Pressure)을, 샘플링 방법으로 바이너리 토너먼트 선택(Binary Tournament Selection)을 사용한다. 교배연산을 위하여 개체간 차이를 이용한 교배 개체쌍 선택 방법과 다중모드 교배 연산자(Multi-mode Crossover)를 사용한다(3). 돌연변이 연산자는 개체 내의 각 비트에 독립적으로 돌연변이 확률을 적용하고 이에 따라서 해당 비트를 반전시키는 동작을 수행한다. 재귀적인 구조를 가지는 신경망의 진화성능을 개선시키고, 교배 후의 개체들이 충분한 다양성을 가지게 하기 위하여 교배 연산자를 보조하는 내부복제 연산자를 사용한다.(4)

3. 재구성 가능 하드웨어에 의한 BNN 구현

하드웨어 구현이 용이한 형태의 기본블록을 제시하고, 기본블록의 결합에 의해 구성되는 블록기반 신경망 모델 개발했다. 또한, 컴퓨터 시뮬레이션 상으로 블록기반 신경망의 진화에 적합한 진화방법을 사용해 제한한 블록기반 신경망의 성능을 패턴분류를 통해 확인했다. 그리고, 실제로 블록기반 신경망을 하드웨어로 설계해 제한한 신경망의 성능을 확인한다.

3.1 재구성 가능 하드웨어의 구조

FPGA는 Field Programmable Gate Array의 약자로 Field Program 이란 반도체 제조 업체를 통하지 않고도 산업 현장에서 엔지니어가 직접 Device를 프로그래밍하여 설계한 회로를 반도체 Chip상에 구현할 수 있다는 것을 의미한다.

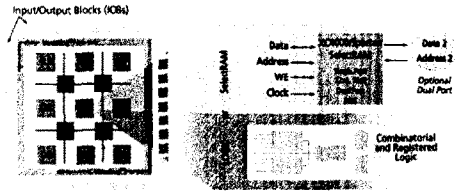


그림 6. FPGA의 구성

그림 6은 FPGA의 일반적인 구성을 나타낸다. 실험에서 사용한 Xilinx의 재구성 가능 하드웨어는 CLB(Configuration Logic Block)과 SM(Switch Matrix)로 구성되어 있어 CLB와 SM의 조정을 통해 구조적인 변화를 얻을 수 있다. Gate Array는 원래 ASIC(Application Specific Integrated Circuits) Device종류의 하나로 FPGA의 내부 구조가 ASIC의 Gate Array와 유사하기 때문에 붙여진 것이다.

3.2 FPGA에 의한 BNN 구현

블록기반 신경망의 구조와 웨이트를 결정하는 개체인 비트 스트림을 오프라인으로 진화시킨 후, 최종적인 비트 스트림을 메모리에 저장하고, 메모리에서 최종의 비트 스트림을 가져와 구조와 웨이트를 결정한 후, 주어진 입력에 대한 출력을 구하게 된다.

각 모듈의 구성 및 구현 방법은 다음과 같다.

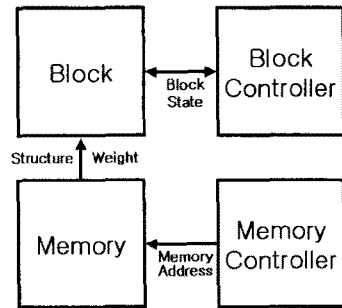


그림 7. 구현된 블록기반 신경망의 블록 다이어그램

Block

제한한 BNN의 블록의 기능과 연산 기능을 가지는 로직이다. 메모리에서 구조와 웨이트 정보를 가져와 출력을 계산한다.

Block Controller

구조와 가중치의 결정을 위해 Memory에서 읽어들이는 비트 스트림의 해석과 블록의 제어를 위한 기능을 가지는 로직이다. 블록의 제어는 현재 어느 블록의 연산이 어떻게 진행되고 있는지에 대한 판단이 이루어진다.

Memory

개체인 비트 스트림의 입출력을 위한 메모리의 기능을 가지고, 개체 저장을 위한 개체군의 데이터는 FPGA내부의 메모리 블록으로 구현될 수 있다.

Memory Controller

시스템의 내부의 연산 모듈과 내부 메모리 모듈 사이의 인터페이스 및 데이터 입출력을 관리한다.

4. BNN 하드웨어 구현 실험

유전자 알고리즘에 의하여 오프라인 상으로 실험해본 분류에 적합한 블록기반 신경망의 구조와 가중치를 획득한다. 제한한 BNN 블록의 기능을 가지는 하드웨어 모듈을 설계하고, 블록을 제어하는 로직과 메모리, 그리

고 메모리 제어를 위한 콘트롤러를 설계한 후, 획득한 비트 스트링을 이용해 패턴분류를 수행한다.

4.1 패턴분류를 위한 블록기반 신경망

블록기반 신경망과 이의 최적화 방법의 성능을 확인하기 위하여 XOR, 3비트 패리티, 4비트 대칭 패턴을 사용한다.

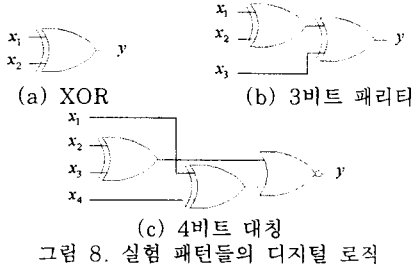


그림 8. 실험 패턴들의 디지털 로직

그림 8은 구현된 블록기반 신경망의 성능을 확인하기 위하여 사용한 패턴들의 디지털 로직이다.

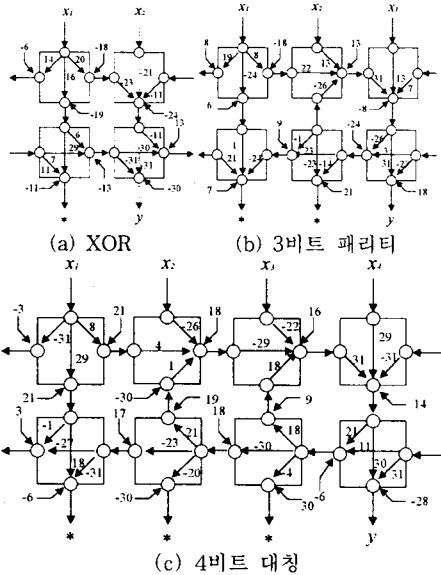
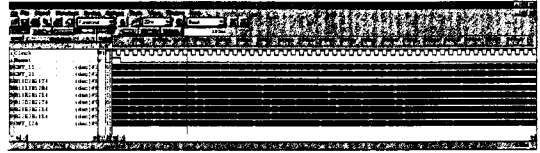


그림 9. 최적화된 블록기반 신경망의 구조와 가중치

그림 9는 진화후 획득한 최종 개체를 해석한 각 패턴에 최적화된 블록기반 신경망의 구조와 가중치를 나타낸다. 유전자 알고리즘에 의해 획득한 비트 스트링을 이용해 블록기반 신경망을 구성하는 하드웨어의 구현은 Xilinx의 Foundation Series를 사용해 VHDL로 코딩을 하였다[5][6]. 테스트 장비는 Elysis사의 EDA Kit 2015를 사용했고, FPGA Chip은 Xilinx VIRTEX이다. 구현된 하드웨어는 바이너리 스트링이 저장된 메모리 모듈에서 구조와 웨이트 정보인 개체, 즉 비트스트링을 받아서 각각의 블록의 구조와 웨이트를 결정한 후, 주어진 입력에 대한 출력을 계산한다.

4.2 실험 결과

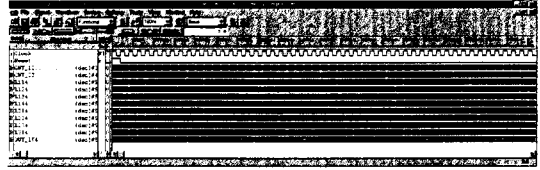
재구성 가능 하드웨어인 FPGA로 설계한 블록기반 신경망은 실험을 위한 3가지의 XOR, 3비트 패리티, 4비트 대칭 패턴을 잘 분류해 주었다. 이미 컴퓨터 시뮬레이션 상으로 그 패턴분류의 성능을 입증하였고[2], 본 논문에서는 하드웨어로 설계한 블록기반 신경망의 성능을 입증하였다.



(a) XOR



(b) 3비트 패리티



(c) 4비트 대칭

그림 10. BNN의 시뮬레이션 결과

그림 10은 블록기반 신경망의 시뮬레이션 결과를 나타낸다. 블록의 맨 상단에 입력 패턴을 가했을 때, 출력이 나오는 시점은 블록이 $m \times n$ 인 경우 mn 번의 블록 연산이 끝난 후에 나온다. 그 이유는 $m \times n$ 블록기반 신경망에서 동일 stage내에서 블록간 거리가 가장 먼 경우는 stage내의 블록간 연결이 모두 같은 때이고 이때 가장 긴 블록간 정보전달 경로를 가지게 된다. 그러므로 각 stage에서의 출력은 stage내의 블록을 순차적으로 n 회 반복연산 함으로서 안정된 출력을 얻을 수 있다..

5. 결 론

본 논문에서는 재구성 가능 하드웨어에 의해서 구현이 용이한 블록기반 신경망 모델을 제안하고, 이를 하드웨어로 구현해 보았다. 블록에 기반한 진화신경망에 유전자 알고리즘을 적용하여 얻은 최종 비트 스트링을 재구성 가능 하드웨어인 FPGA에 적용하고, 패턴의 분류를 수행함으로써 하드웨어로 구현한 블록기반 진화신경망이 올바르게 동작함을 증명하였다.

(참 고 문 헌)

- [1] X. Yao and T. Higuchi, "Promises and challenges of evolvable hardware," IEEE Trans. on System, Man & Cybernetics Part C: Applications & Reviews, Vol. 29, No. 1, pp. 87-07, 1999.
- [2] 문상우, 공성근, "블록기반 신경망을 이용한 패턴분류," 퍼지 및 지능시스템학회 논문지, 9권, 4호, 396-403, 1999. 8.
- [3] B. Thang Nguyen and M. Byung Ro, "On Multi-Dimensional Encoding/Crossover," Proc. 6th International Conf. on Genetic Algorithms(ICGA), pp. 49-56, 1995.
- [4] T. Kumagai, M. Wada, S. Mikami, and R. Hashimoto, "Structured Learning in Recurrent Neural Network Using Genetic Algorithm with Internal Copy Operator," Proc. IEEE International Magnetics Conf., pp. 651-656, 1997.
- [5] 박인정, 장호성, "VHDL을 이용한 디지털 시스템 설계," 한성출판사, 1998.
- [6] Douglas J. Smith, "HDL Chip Design," Doone Publications, 1996.