디지털 신경회로망의 하드웨어 구현을 위한 재구성형 모듈러 디자인의 적용

윤석배*, 김영주*, 동성수**, 이종호*
인하 대학교 전기공학과*, 용인 송담 대학 디지털 전자정보과**

A reconfigurable modular approach for digital neural network

Seokbae Yun*, Youngjoo Kim*, Sungsoo Dong**, Chongho Lee*
Inha University*, Yong-in Songdam College**

Abstract - In this paper, we propose a new architecture for hardware implementation of digital neural network. By adopting flexible ladder-style bus and internal connection network into traditional SIMD-type digital neural network architecture, the proposed architecture enables fast processing that is based on parallelism, while does not abandon the flexibility and extensibility of the traditional approach. In the proposed architecture, users can change the network topology by setting configuration registers. Such reconfigurability on hardware allows enough usability like software simulation. We implement the proposed design on real FPGA, and configure the chip to multi-layer perceptron with back propagation for alphabet recognition problem. Performance comparison with its software counterpart shows its value in the aspect of performance and flexibility.

1. 서 론

신경회로망은 일반적인 폰 노이만형 컴퓨터와 달리 인간의 뇌 구조에 기반한 정보처리 방법이다. 신경회로망은 데이터의 병렬, 분산 처리함으로써 연상, 추론, 학습등의 기능을 구현하는데 유리한 반면, 그 응용 범위가 한정 되어 있고 하드웨어 구현에 어려움이 있기 때문에, 실질적인 하드웨어로서는 널리 쓰이지 못하고 있다.

하지만 신경회로망은 병렬성을 기본으로 하기 때문에, 기존의 디지털 컴퓨터에서의 시뮬레이션만으로는 한계를 가지며, 그에 따라 실제적인 신경회로망의 하드웨어 구현을 필요로 하게 된다. 이에 본 논문에서는 가변구조를 기반으로 하는 모듈러 신경회로망을 통해서 확장성 및 가변성을 확보하는 방법을 소개하고 이에 대해서 실제적인 하드웨어로 구현함으로써 그 유용성을 입증하였다.

2. 신경회로망 하드웨어

2.1 배 경

최초의 하드웨어 신경회로망의 개발은 F. Rosenblatt에 의해 1958년에 개발된 MARK I Perceptron이었다. 이후 40여년 동안 신경회로망은 발전과 쇠퇴를 반복해왔으나, 디지털 컴퓨터가 IC의 출현에 힘입어 급속도로 발전한 반면 신경회로망 컴퓨터는 일반화된 문제에 적용될 만큼 발전하지 못했었다.

이러한 초창기의 신경회로망은 대부분이 아날로그 기술로 만들어졌었다. 일반적으로 아날로그 신경회로망은 논리적인 신경회로망을 직관적으로 만들 수 있는 장점이 있으며, 가중치 적용에 대해서 곱셈기를 사용하지 않음으로써 적은 면적을 차지하면서도 매우 빠른 병렬 구조를 획득할 수 있다. 반면에, 범용성이 크게 제한되고 학습된 가중치 값의 저장이 어려우며, 전기적 잡음과 온도에 민감하여 오차율이 크고, 기존의 디지털 컴퓨터와의 인터페이스가 어려운 단점들이 있어서 상업적 제품으로는 성공하지 못하였다. 이후 90년대 중반까지 이러한 신경회로망 칩 개발은 다소 침체된 양상을 보였었으나, 90년대 후반에 들어서 디지털 반도체 기술의 비약적인 발전에 힘입어 신경회로망 칩에 대한 연구가 다시 활성화되기 시작했다. ASIC 기술이 발전하고 FPGA가 등장함에 따라 활발하게 연구된 디지털 신경회로망은 아날로그 신경회로망 칩에 비해 면적을 많이 차지하고 느린 속도를 가진다는 단점이 있으나, 가중치 값의 저장이 용이하고 기존의 디지털 프로세서 기술을 사용함으로써 쉽게 제작할 수 있으며, 아날로그 신경회로망에 비해 비교적 자유로운 구조를 가질 수 있는 장점이 있다. 그림 1은 하드웨어 신경망에 대한 분류이다.

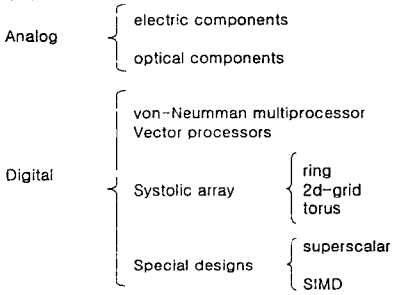


그림 1 하드웨어 신경회로망의 분류

2.2 디지털 하드웨어 신경회로망의 연구

디지털 신경회로망을 구현하는 방법은 크게 두 가지로 나눌 수 있다. 첫 번째는 DSP 혹은 CPU를 이용한 마이크로 프로세서 기반의 신경회로망이며, 두 번째는 ASIC 형태로써 직접적으로 시냅스, 뉴런등의 회로를 구성하여 신경회로망을 구현하는 방법이다.

일반적인 완전연결(Fully Connected)형태를 갖는 신경망 구조에서 뉴런의 증가에 대해서 시냅스의 개수는 뉴런수의 제곱의 형태로 증가하게 되고, 이에 비례하여 곱셈기의 양이 증가하게 된다. 이러한 곱셈기의 증가에 따른 면적 소모 및 속도 저하는 곱셈기가 사용되지 않는 아날로그 신경회로망에서 보다 디지털 신경회로망에서 더 큰 단점으로 나타나게 되며, 이를 극복하기 위한 방법으로 곱셈기가 없는 디지털 신경망[1], 혹은 곱셈기의 크기를 줄이기 위한 시리얼 데이터 방식의 디지털 신경망[2]과 같이 구조를 그대로 두고 곱셈기의 크기를 줄이는 연구들이 이루어졌다. 그리고 다른 방법으로는 SIMD 형태와 같이 버스 구조를 통해서 곱셈기를 재사용하는 디지털 신경망[3]과 사슬 배열 구조(Systolic Array Architecture)를 채용하여 곱셈기의 수를 선형적으로 늘이는 신경회로망, 셀룰러 신경회로망(Cellular Neural Network)처럼 인접연결(Neighborhood Connected)을 사용하는 신경망과 같이 기존의 완전연결 구조를 물리적으로 구현하지 않고 모듈러 아키텍쳐 형태로 구현함으로써 뉴런 및 시냅스의 증가에 따른 곱셈기의 증가 문제를 해결하는 연구가 이루어졌다.

또한, 범용성의 확보 즉, 여러 가지 목적에 적합한 신경회로망을 구현하기 위하여 신경회로망의 재구성, 혹은

재 프로그래밍 방법이 연구되어 왔는데, 첫 번째는 범용 프로세서를 기반으로 하고 목적에 맞는 신경회로망의 프로그램을 사용하는 방법과 FPGA처럼 간단한 재구성 비트를 사용하여 기본 PE의 기능과 버스 구조를 변경함으로써 목적에 맞는 신경회로망으로 구성하는 재구성형 하드웨어에 기반한 신경회로망[4]이 있다.

2.3 재구성형 모듈러 신경회로망의 구현

2.3.1 SIMD 구조

SIMD(Single Instruction Multiple Datas) 구조는 단일 명령에 대해서 다중의 프로세서가 병렬로 동작하는 특성으로 인해서 디지털 신경회로망의 하드웨어 구현 방법으로 적합한 구조이다. 이러한 구조는 뉴런 병렬처리를 구현하기가 쉬우나, 시냅스 연산에 필요한 곱셈기를 재 사용함으로써 각각의 뉴런에 대한 시냅스 병렬처리가 어려운 단점이 있으며, 시냅스 병렬 처리가 구현되지 않을 경우 시냅스의 개수 증가에 따라 뉴런의 연산량이 증가하는 단점이 있다.

제안된 신경회로망에서는 프로세싱 모듈의 내부에만 SIMD구조가 사용되었으며, 각각의 모듈은 마스터 슬레이브 형태로 연결된다.

2.3.2 기본 프로세싱 유닛

본 논문에서는 하나의 레이어를 구성하는 하나 또는 다수의 뉴런을 구현할 수 있는 회로 블록을 프로세싱 모듈(Processing Module)이라 지칭하였다. 프로세싱 모듈은 기본적으로 두가지의 PE(Processing Element)로써 구성되어 있으며, 그중에서 입력 노드에 대한 가중치를 곱하는 시냅스 기능과 각 시냅스의 누적합 기능을 가지고 있는 SPE(Synapse Processing Element)이라 하였고, 누적된 시냅스의 결과에 대한 활성화 함수값을 출력하거나 PE간의 데이터 전송방향을 제어하는 LPE(Layer Processing Element)라고 명명하였다. 그림 2는 4개의 SPE와 1개의 LPE로 구성된 형태의 프로세싱 모듈의 구조를 보여준다.

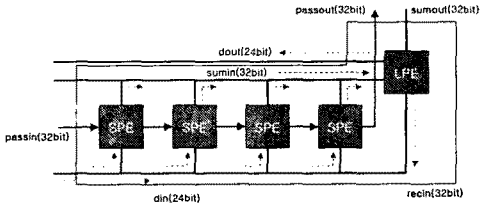


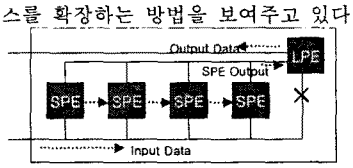
그림 2 기본 프로세싱 모듈의 구조

2.3.3 가변 버스 구조

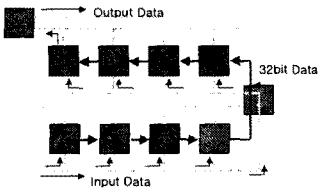
제안된 구조는 모듈간의 연결 구조를 바꿈으로써 여러가지 신경회로망을 구현할 수 있도록 되어 있다. LPE에 저장되는 구성비트에 따라서 버스의 방향 및 AFU(Activation Function Module)의 기능이 결정된다. 구성비트의 설정에 의해서 외부, 혹은 다른 모듈로부터의 입력이 아닌 자신의 모듈, 혹은 같은 레이어상에 존재하는 다른 모듈로부터의 출력값을 입력으로 사용하는 것이 가능하다. 이 기능으로 홉필드 같은 역방향 신경망(Feedback Neural Network)을 구현하는 것이 가능하다.

2.3.3 내부 연결 버스

본 논문에서는, 32비트 크기를 갖는 내부 연결 버스를 사용하여 SPE에서 누적된 시냅스 값을 인접 SPE로 전달함으로써 시냅스를 확장하는 방법을 사용하고 있다. 이것은 단지 모듈의 내부 뿐 아니라 그림 4의 프로세싱 모듈 구조에서도 나타나듯이 모듈 간에도 적용이 가능하다. 따라서 시냅스의 확장은 모듈 내부에 국한되지 않고 모듈을 초과하는 양의 시냅스도 하나의 뉴런에서 처리하는 것이 가능하다. 그림 3 은 내부 연결버스를 사용하여 시냅스를 확장하는 방법을 보여주고 있다. 그림 3의 (a)는 하나의 모듈 내에서 시냅스를 확장하는 형태를 보여주고 있으며, 그림 3의 (b)는 두개의 모듈을 이용하여 시냅스를 확장하는 방법을 보여주고 있다.



(a) Synapse expansion in a single module

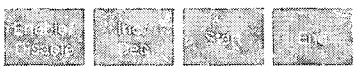


(b) Synapse expansion using 2 modules

그림 3 내부 연결 버스를 이용한 시냅스의 확장

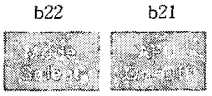
2.3.4 재구성 및 확장

구성 형태와 확장 형태를 정의하기 위해서는 먼저 구성 비트에 대해 정의해야 한다. SPE의 경우는 그림 4와 같은 구성비트 형식을 사용하고 LPE의 경우는 그림 5와 같은 구성비트 형식을 사용한다.



■Enable/Disable : 사용되는 SPE인지 아닌지를 결정
■Inc/Dec : 누산기(Accumulator)의 덧셈/뺄셈을 결정
■Start : 뉴런내의 최초 시냅스 블록임을 알림
■End : 뉴런내의 최종 시냅스 블록임을 알림

그림 0 SPE의 구성 비트



■00 : 모듈을 Feed Forward형태로 설정, 비선형 함수를 출력
■01 : 모듈을 Feed Forward형태로 확장, 비선형 함수 사용안함
■10 : 모듈을 Feed Back형태로 설정하고 비선형 함수를 출력
■11 : 모듈을 Feed Back형태로 확장,비선형 함수 사용안함

그림 0 LPE의 구성비트

구성 가능한 모듈의 형태는 기본적으로 구성 비트들과 LUT에 의존하게 된다. 그 외의 구성 형태에 미치는 요소는 내부연결 버스 형태와 시냅스의 완전연결, 부분연결등이 영향을 미치게 된다. 이와 같은 요소에 의해 구성 가능한 형태는 SPE 구성비트(2^4), LPE 구성비트(2^2), 내부연결 버스형태 (2^1),시냅스의 완전연결,부분연결 (2^1)으로써 최소 $2^8=256$가지가 되며, 실제적으로 모듈 내부의 SPE 개수와, 비선형 함수의 출력, 부분 Feed Back에 의한 TDNN(Time Delayed Neural Network) 구성등에 의해서 수 많은 형태의 신경회로망으로 조합이 가능하다.

이러한 구성들 가운데, 실제적으로 자주 쓰이는 신경망인 다층 퍼셉트론(Multilayer Perceptron)에 대한 구성의 예를 그림 6에 나타내었고, 마찬가지로 Recurrent 신경회로망의 대표적인 홉필드 네트워크(Hopfield Network)를 그림 7에 나타내었다.

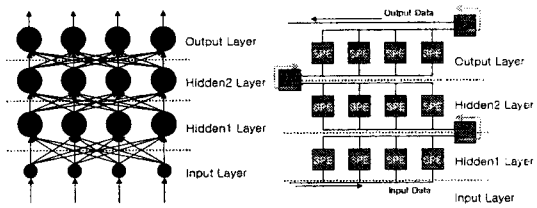


그림 6 (4X4X4X4) 의 다층 퍼셉트론 구현예

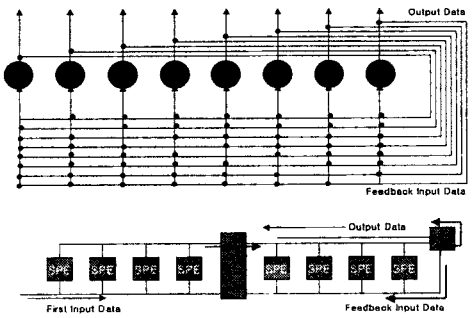


그림 7 8개의 뉴런을 갖는 홉필드 네트워크의 구현

2.4 실험 결과

2.4.1 FPGA 구현

디자인에 사용된 언어는 Verilog 언어를 사용하였고 설계환경으로는 PC와 Workstation을 사용하였다. 사용된 디자인 툴은 PC용 Xilinx Foundation ISE 4.1을 통해서 세부 모듈을 디자인 했으며, Workstation용 HDL 합성 툴인 LeonardoSpectrum을 사용하여 합성하였다. 합성된 회로는 Workstation용 Modelsim 5.5 SE를 통해서 시뮬레이션 하였으며, Xilinx Alliance I SE 4.1을 통해서 Implementation 과정을 수행하였다. 최종적으로 생성된 bit화일은 Celoxica의 DK-1 디자인 툴을 이용해서 RC-1000pp 보드에 탑재되어 있는 Xililnx FPGA인 VirtexE 2000 BG560-6에 로딩하였다. 그림 8은 FPGA상에서 구현된 본 신경망의 가중치 저장 및 시냅스 연산에 대한 파형을 보여주고 있다.

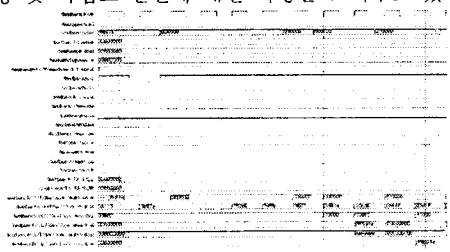


그림 8 가중치 저장 및 시냅스 연산 시작시 파형

2.4.2 실 험

본 논문에서는 제안된 하드웨어의 유용성을 증명하기 위해서 알파벳 영문자 인식 실험을 수행하였다. 사용된 신경망은 MLP 구조를 사용했으며 학습에는 BP알고리즘을 사용하였다. 본 디자인에는 학습 모듈이 내장되어 있지 않기 때문에, PC상에서 C언어로 구현된 소프트웨어 시뮬레이터를 통해서 학습된 가중치를 다운로드해서 사용하는 Pre-trained Model을 사용하였다. 그림 9는 학습에 사용된 문자 패턴 및 테스트용으로 사용된 손상된 문자 패턴이다. 입력 패턴으로써는 영어 대문자 폰트 26개에 대해서 5가지 폰트를 사용했으며 각각의 왼쪽은 정상 패턴, 오른쪽은 손상된 패턴을 나타낸다.

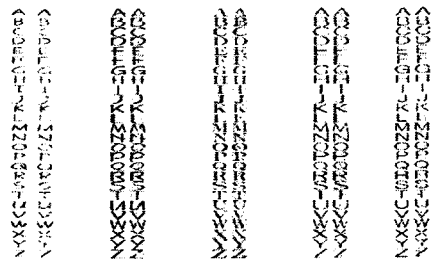


그림 9 실험에 사용된 알파벳 패턴

손상된 패턴은 신경망의 적응성을 테스트하기 위하여 사용되었다. 구성된 신경회로망의 크기는 입력층의 뉴런 개수 256개 (16X16 폰트사용) 은닉층 1개에 은닉층의 뉴런 개수 32개 출력층의 뉴런개수 5개를 사용하였으며, 각 네트워크에 저장된 가중치는 소프트웨어 학습 알고리즘으로 구했으며, 오류 역전파 알고리즘을 사용하였다. 학습률은 0.05 이며, 에러 임계값 보다 에러가 작아지면 학습을 중지 시키는 방법을 사용하였다. 학습에 걸린 반복 회수는 약 2000회 이며, 출력 형태는 그레이 코드(Gray Code)형태로 학습 시켰다.

하드웨어 시뮬레이션 실험은 구현된 신경회로망 모델을 통해서 256 * 32 * 5 의 MLP구조를 구성한 뒤, 입력 패턴과 가중치를 하드웨어 사양에 맞게 24비트 데이터 포맷으로 변환하여 입력시키고, 출력된 데이터를 소프트웨어 결과의 오차와 비교하였다.

실험 결과에 대해서는 에러 임계값을 0.0001로 주었을 때 양쪽 모두 100%의 인식률을 보였고, 연산 중간결과는 하드웨어의 연산 결과와 소프트웨어의 연산 결과를 비교했을 때, 시냅스 연산에 대해서0.0002% 이내의 오차율을, 비선형 함수에 대해서 0.0003%의 오차율을 보였다.

4. 결 론

본 논문에서 연구된 재구성형 하드웨어 신경회로망은 목적에 맞게 그 구조와 크기를 변경할 수 있다는 점에서 기존의 하드웨어 신경회로망 에 비해 그 활용도가 높으며, 소프트웨어 신경회로망이 사용되기 어려운 실시간 문제나 소형의 모듈이 필요한 곳에 사용될 수 있다는 장점이 있다. 또한 동작 중에 구조를 바꾸는 것이 가능하여 진화형 신경회로망을 구성 할 수도 있으며, 칩의 외부 확장이 가능하기 때문에 초대형 신경회로망 칩을 구성하는 것도 가능하다. 하지만, 아직 자체 학습 모듈 및 진화 모듈 등을 갖추지 못한 상태이기 때문에, 이 부분에 대한 연구가 더 필요할 것으로 생각된다.

[참 고 문 헌]

[1] Miroslav Skrbek, "Fast neural network implementation", Neural Network World, p357-391, 1999.5

[2] Tamás Szabó, Lőrinc Antoni, Gábor Horváth, Béla Fehér, "A full-parallel digital implementation for pre-trained NNs" , IJCNN 2000, ,PP.49-54 vol.2, 2000

[3] B. Pino, F.J.Pelayo, J. Ortega and A. Prieto, "Design and Evaluation of a Reconfigurable Digital Architecture for Self-Organizing Maps", Microelectronics for Neural, Fuzzy and Bio-Inspired Systems, pp.395-402, 1999

[4] Bernard Girau, "Digital hardware implementation of 2D compliatible neural networks", IJCNN 2000, Volume: 3, pp.506-511, 2000