

## 유전자 알고리즘을 이용한 블록 기반 진화신경망의 최적화

문상우, 공성곤  
송실태학교 전기공학과  
전화 : (02) 822-3812 / 팩스 : (02) 817-7961

## Optimization of Block-based Evolvable Neural Network using the Genetic Algorithm

Sang-Woo Moon and Seong-Gon Kong  
Department of Electrical Engineering, Soongsil University  
E-mail : kagami@ee.soongsil.ac.kr

### Abstract

In this paper, we proposed an block-based evolvable neural network(BENN). The BENN can optimize it's structure and weights simultaneously. It can be easily implemented by FPGA whose connection and internal functionality can be reconfigured. To solve the local minima problem that is caused gradient descent learning algorithm, genetic algorithms are applied for optimizing the proposed evolvable neural network model.

### 1. 서론

인간 신경의 모델링으로부터 출발한 신경망은 기존의 방법으로는 해결하기 어려웠던 과다하게 복잡하거나 또는 비교적 정보가 부족한 문제에 대한 해결방안을 제시해 주었다. 하지만 신경망 자체도 몇 가지 문제점을 안고 있다. 그 대표적인 예로 문제에 따른 사전지식으로서 요구되는 신경망의 구조를 결정하는 문제와 신경망의 학습방법으로 경사 하강법을 사용함으로서 생기는 지역 최소점으로의 수렴 문제를 들 수 있다.

지역 최소점으로의 수렴을 극복하기 위하여 고정된 신경망 구조 하에서 가중치를 결정하기 위하여 기존의 경사 하강법에 근거한 학습 알고리즘을 사용하지 않고 신역 탐색이 가능한 유전자 알고리즘을 이용한 연구가 있다[1][2]. 또한, 학습은 기존의 방법을 따르면서 신경망의 구조를 사전지식에 의존하여 결정하지 않고 진화 알고리즘을 사용하여 놓적으로 변화시키는 방법에 관한 연구[3], 마지막으로 구조와 가중치 모두를 진화 알고리즘을 이용하여 최적화 시키는 연구가 수행되고 있다[4][5].

본 논문에서는 신경망의 구조와 가중치를 놓적으로 변화시키며 FPGA등의 재구성이 가능한 하드웨어로의 적용이 용이한 새로운 진화신경망을 제안하고 이의 구조와 가중치를 진화 연산의 하나인 유전자 알고리즘을 통하여 최적화 한다. 사용한 유전자 알고리즘은 제안한 신경망 모델을 진화시키는데 적합하도록 수정한다. 또한 제안한 진화신경망과 이의 성능을 몇 가지 폐턴 분류 문제에 적용하여 성능을 평가한다.

### 2. 블록기반 진화신경망

#### 2.1 진화신경망의 구조

블록기반 진화신경망(BENN) : Block-based Evolvable Neural Network 모델은 기본 단위인 블록들의 2차원 배열로 이루어진다.

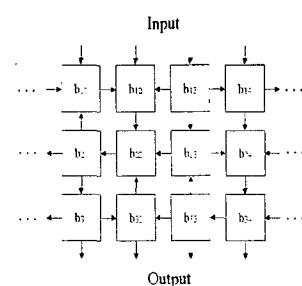


그림 1: 블록기반 진화신경망의 구조

그림 1은 기본 블록들로 이루어진  $3 \times 4$  진화신경망의 구조를 나타낸다. 각 블록  $b_{ij}$ 는 4개의 입출력 노드를 가진다. 이들 입출력 노드들은 블록의 내부에서 서로

로 완전히 연결된다. 블록들을 연결하는 화살표는 블록들 간의 또는 외부와의 정보의 흐름을 나타낸다. 블록의 출력은 인접한 블록의 입력이 되며 좌우측 끝에 위치한 블록들은 서로 연결되어 있다. 블록은 각기 인접한 4개의 블록들과의 연결만을 허용하며 모두 입력 또는 출력인 경우 또한 허용한다.

## 2.2 기본블럭의 내부구조

각 기본 블록은 총 4개의 입출력 뉴런으로 구성되어 있는 간단한 전방향 신경망 모델로 이루어진다. 블록들의 내부 구조는 입출력 형태에 따라 결정된다. 각 블록에 사용되는 가중치는 하드웨어 구현을 용이하게 하기 위하여 정수만을 허용한다.

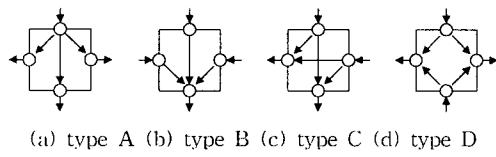


그림 2: 입출력 형태에 따른 4가지 기본 블록의 내부 구조

기본 블록은 모두 입력 또는 출력인 경우를 제외할 경우 4가지 각기 다른 형태를 가지고 있다. 그림 2는 4가지 형태의 기본 블록을 보여주고 있다. 각 블록은 '입력/출력'의 개수에 따라서 type A, B, C, D의 4가지 형태로 구분한다. 예로서, 그림 2(a)는 type A(그림 1의  $b_{13}$ ), 그림 2(b)는 type B(그림 1의  $b_{11}$ ), 그림 2(c)는 type C(그림 1의  $b_{34}$ ), 마지막으로 그림 2(d)는 type D(그림 1의  $b_{23}$ )의 형태를 나타낸다. 이들은 각기 4개의 입출력 뉴런을 가지는 전방향 단층 신경망과 같다.

입력 뉴런은 선형 활성함수를, 출력 뉴런은 0과 1의 제한된 범위 내에서 연속된 값을 나타내기 위하여 다음과 같이 정의된 시그모이드 활성함수를 사용한다.

$$f(x) = \frac{1}{1 + e^{-\alpha x}} \quad (1)$$

블록간의 신호의 흐름은 각 블록들의 입출력 형태를 결정하고, 이는 블록들의 내부 구조를 결정하게 된다.

## 3. 진화과정

블록기반 진화신경망의 진화는 전체 시스템의 구조 최적화를 위한 각 블록의 입출력 형태 결정과 블록 내부의 가중치 결정으로 나뉜다. 이들을 함께 최적화하기 위하여 유전자 알고리즘을 사용한다.

제안한 진화신경망의 유전자 알고리즘을 이용한 최적화 과정에는 사용한 유전자 알고리즘의 동작 방법, 개체의 인코딩, 사용 유전 연산자의 동작, 적합도 함수의 정의가 필요하다.

### 3.1 유전자 알고리즘의 적용

보통의 유전자 알고리즘과는 달리 본 연구에는 보다 개선된 성능을 위하여 유전자 알고리즘의 적용 방법을 변화시켰다. 보통의 유전자 알고리즘은 재생산 후 사용하는 유전 연산자를 모두 사용하고 적합도를 평가하는 과정을 반복한다.

변화시킨 방법은 보통의 유전자 알고리즘과는 다르게 재생산 이전의 개체를 저장하고 유전 연산자를 적용한 후 1%의 노이즈를 추가하여 적합도를 평가한다. 그리고 저장했던 전체 개체 중 상위 30% 적합도의 개체를 유전 연산자 적용 후의 개체 중 하위 30% 적합도를 가지는 개체와 교환한다.

이러한 과정을 보통의 유전자 알고리즘에 추가하여 진화시 탐색의 방향성을 유지하였다.

### 3.2 인코딩

블록간 정보의 전달방향은 각 블록의 입출력 형태를 결정하고 이는 블록의 내부구조를 결정하게 된다. 건국 제안한 진화신경망의 구조는 블록간 정보의 전달방향에 따라 결정된다. 따라서 구조와 가중치를 동시에 진화시키기 위해서는 블록간 정보 전달방향과 가중치를 인코딩하여야 한다.

블록기반 진화신경망에 적용하는 유전자 알고리즘에서 주 연산자인 교배의 성능을 높이기 위하여 본 논문에서는 블록간 정보 전달방향과 가중치를 함께 2차원으로 인코딩 하였다[6].

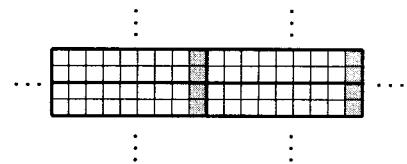


그림 3: 블록의 조합으로 이루어진 블록기반 진화신경망의 2차원 인코딩

그림 3은 가중치를 4비트 표현한 블록의 인코딩과 이를 조합한 블록기반 진화신경망의 2차원 인코딩을 나타낸다. □는 가중치를, ■는 아래쪽(↓)과 왼쪽(←)을 0으로, 위쪽(↑)과 오른쪽(→)을 1로 정의한 입출력 정보의 전달방향을 표현한다. 입력과 출력 정보 전달방향은 항상 고정되어 있기 때문에  $m \times n$  블록으로 구성된 블록기반 진화신경망의 구조는  $(2m-1)n$  비트로 표현하고, 각 블록은 2/2 입출력 형태에서 최대 4개의 가중치를 가지기 때문에  $m \times n$  블록기반 진화신경망의 가중치는  $4l \times m \times n$  비트로 인코딩 한다. 인코딩된 개체의 각 블록에서 2진수로 변환된 가중치는 최하위 비트가 마주보도록 배열한다.

연결강도  $w$ 는 연결이 끊어졌음을 의미하는 연결강도 0을 2개 포함하도록  $-(2^{l-1}-1)$ 에서  $2^{l-1}-1$ 의

범위 내에서 표현하였다.

$$w = w_b - 2^{(l-1)} + s(w_b) \quad (2)$$

$$s(w_b) = \begin{cases} 1 & \text{if } w_b < 2^{l-1} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

식 (2)는  $l$ -비트로 표현된 양의 정수값을 가지는  $w_b$ 를 가중치  $w$ 로 변환해 준다.

### 3.3 유전연산자

#### 제생산 (Reproduction)

유전자 알고리즘을 이용하여 블록기반 진화신경망의 가중치를 방향성을 가지고 진화시키기 위하여, 또한 개체분포에 따른 진화 초기와 말기의 바람직하지 못한 탐색성능을 개선하기 위하여 고정된 분포를 이용하여 재생산을 수행하는 지수함수를 이용한 적합도 순위(fitness ranking)를 사용한다[7][2].

#### 교배 (Crossover)

본 논문에서는 블록기반 진화신경망의 구조에 적합한 교배 연산자를 도입한다. 도입한 교배 연산자는 교배률  $p_c$ 에 의하여 교배 대상 개체를 선택한 후 개체 내의 코딩되어진 모든 블록의 각 행에 대하여 그림 4에 나타낸 확률에 의하여 선택된 교배점에 의한 2점 교배의 형태를 취하고 있다.

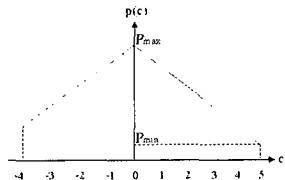


그림 4: 교배점 선택시 사용한 확률밀도함수

그림 4는 가중치를 4비트로 표현한 경우 제안한 교배 연산자에서 사용된 확률밀도함수를 나타내고 있다. 각 행을 구성하는 두 가중치의 최하위 비트들을 같은 확률로 교배시키기 위하여  $l+1$ 번째의 교배 가능점으로부터 교배점으로 선택될 확률을 좌우 대칭으로 분포 시킨다.  $p_{\max}$ 는 인코딩시  $l$  비트로 가중치를 표현한 경우  $2(l+1)$ 개의 교배점 중 가운데인  $l+1$ 번째 교배점이 선택될 확률,  $p_{\min}$ 은 1번째 또는  $2(l+1)$ 번째 교배점이 선택될 확률을 나타낸다.

$$p(c) = -\frac{c(m-n) + m(l+1)}{\sum_{k=-l}^l k(m-n) + 2m(l+1)^2} \quad (4)$$

식 (4)에서  $c$ ,  $m$ ,  $n$ 은 각각 교배점의 위치 색인, 상대 최대값, 상대 최소값을 나타낸다.

#### 돌연변이 (Mutation)

블록기반 진화신경망의 구조와 가중치를 표현하는

비트들에 대하여 각기 다른 돌연변이 확률을 적용하였다.  $p_{mc}$ 는 구조,  $p_{mw}$ 는 가중치를 표현하는 비트에 대한 돌연변이 확률이다.

#### 내부복제 (Internal Copy)

블록기반 진화신경망의 정보 전달은 전방향적인 성질과 재귀적인 성질을 가지고 있다. 따라서 진화의 성능을 개선하기 위해서는 전방향적인 성질 뿐만 아니라 재귀적인 성질 또한 고려하여야 한다. 내부복제 연산자는 재귀적인 구조를 가지는 신경망의 진화성능을 개선시키는 역할을 한다[5]. 그러나 본 논문에서 사용한 교배연산자가 같은 블록들간에만 정보를 교환하는 약점을 보완하기 위하여 내부복제 연산자를 사용한다.

내부복제 연산자는 2차원으로 인코딩 된 개체에서 일부를 다른 부분으로 복사한다. 개체내의 복사 위치와 복사 정도는 임의로 결정한다.

### 3.4 적합도 함수

구조와 가중치를 최적화하기 위한 적합도 함수는 다음과 같이 정의하였다.

$$e_{ij} = d_{ij} - y_{ij} \quad (5)$$

$$f(e_{ij}) = \frac{1}{0.1 + \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^m e_{ij}^2} \quad (6)$$

$d_{ij}$ ,  $y_{ij}$ ,  $n$ ,  $m$ 은 각각 기대치, 실제 출력, 학습 데이터 개수, don't care(\*)를 제외한 출력 블록의 개수를 나타낸다.

## 4. 패턴분류 시뮬레이션

XOR, 패리티 패턴의 분류를 수행함으로서 블록기반 진화신경망의 성능을 시험하였다.

### 4.1 XOR 문제

XOR 패턴을 분류하기 위하여  $2 \times 2$  블록기반 진화신경망을 사용하였다.

사용한 블록기반 진화신경망의 가중치는 6비트로 코딩하였고, 최대적합도의 95% 이상을 만족할 때 진화를 종료하였다.

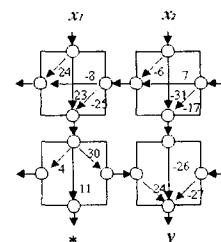
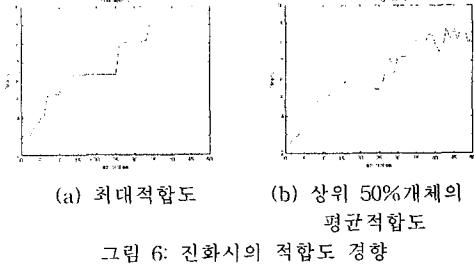


그림 5: 진화에 의한  $2 \times 2$  블록기반 진화신경망의 최종구조와 가중치

그림 5는 진화 후에 얻어진 블록기반 진화신경망의 구조와 가중치의 설정상태를 나타낸다.

그림 6은 진화시의 각 세대에서의 최대적합도와 상위 50% 개체들의 평균 적합도를 나타낸다.



50세대만에 목표한 적합도를 가지는 개체를 생성할 수 있었다.

진화 결과 획득한 블록기반 진화신경망을 이용하여 얻은 결정평면의 형태는 그림 7과 같다.

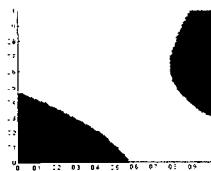


그림 7: XOR에 대한 진화 결과 얻은 블록기반 진화신경망에 의한 결정평면

#### 4.2 패리티 문제

XOR의 확장된 형태인 4비트 패리티 패턴을 분류하기 위하여 가중치를 6비트로 코딩한  $3 \times 5$  블록기반 진화신경망을 사용하였다. XOR 실험에서와 같이 최대적합도 95% 이상을 만족할 때 진화를 종료하였다.

그림 8은 패리티 패턴을 분류하기 위하여 사용한 블록기반 진화신경망의 진화 후에 얻어지는 구조와 가중치를 나타낸다.

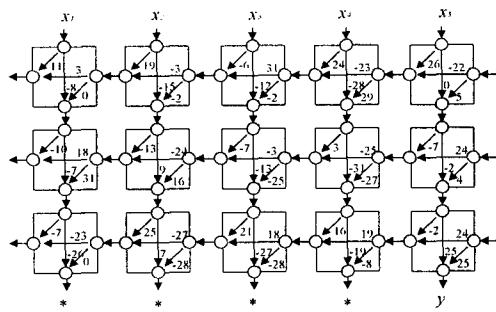
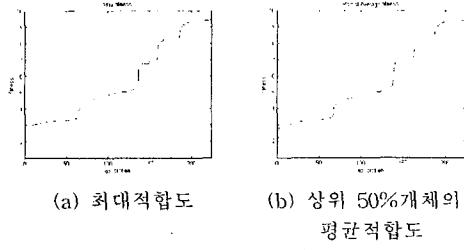


그림 8:  $3 \times 5$  블록기반 진화신경망의 최종구조와 가중치

그림 9는 진화시 개체들의 적합도를 나타낸다.



#### 5. 결론

본 논문에서는 블록에 기반한 진화신경망을 제안하고 이에 유전자 알고리즘을 적용하였다. 제안한 진화신경망 모델은 PLD에 적용하기 쉽도록 디자인되었다. 실험에서는 성능 평가를 위하여 제안한 진화신경을 몇 가지 패턴분류 문제에 적용하였다. 이 경우 블록기반 진화신경망이 문제에 따라서 최적 구조와 가중치를 가지도록 진화함을 확인할 수 있었다.

#### 참고문헌

- [1] D. J. Montana and L. Davis, "Training feedforward neural networks using genetic algorithms," *Proc. 11th Joint Conf. on Artificial Intelligence(IJCAI)*, pp. 762-767, 1989.
- [2] P. G. Kornig, "Training of neural networks by means of genetic algorithm working on very long chromosomes," *Technical Report, Computer Science Department, Aarhus C, Denmark*, 1994.
- [3] D. Whitley, T. Starkweather, and C. Bogart, "Genetic Algorithm and Neural Networks Optimizing Connections and Connectivity," *Parallel Computing*, Vol. 14, pp. 347-361, 1990.
- [4] M. Vittorio, "Genetic Evolution of the Topology and Weight Distribution of Neural Networks," *IEEE Trans. on Neural Networks*, Vol. 5, No. 1, pp. 39-53, 1994.
- [5] 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.
- [6] B. Thang Nguyen and M. Byung Ro, "On Multi-Dimensional Encoding/Crossover," *Proc. 6th International Conf. on Genetic Algorithms(ICGA)*, pp. 49-56, 1995.
- [7] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, 1989.