

# 블록기반 신경망을 이용한 패턴분류

## Pattern Classification using the Block-based Neural Network

문상우 · 공성곤

Sang-Woo Moon and Seong-Gon Kong

숭실대학교 전기공학과

### 요 약

본 논문에서는 새로운 블록기반 신경망을 제안하고, 블록기반 신경망의 패턴분류 성능을 확인하였다. 블록기반 신경망은 4개의 가변 입출력을 가지는 블록을 기본 구성요소로 하고 있으며 블록들의 2차원 배열 형태로 이루어진다. 블록기반 신경망은 재구성가능 하드웨어에 의하여 구현이 용이하고 구조 및 가중치의 최적화에 진화 알고리즘을 적용시킬 수 있는 새로운 신경망 모델이다. 블록기반 신경망의 구조와 가중치를 재구성가능 하드웨어(FPGA)의 비트열에 대응시키고 유전자 알고리즘에 의하여 전역최적화를 하여 구조와 가중치를 최적화한다. 유전 알고리즘에 의하여 설계된 블록기반 신경망을 비선형 결정평면을 가지는 여러 학습패턴에 적용하여 패턴분류 성능을 확인하였다.

### ABSTRACT

This paper proposes a block-based neural network(BNN) model for pattern recognition. The structure and weights of the BNN are optimized with the genetic algorithms. The architecture of the proposed BNN consists of a two-dimensional array of fundamental blocks which can easily be implemented with reconfigurable hardwares. To obtain optimized solution through global search and match the FPGA's reconfigurable bit strings with the BNN, binary chromosomes are implemented the modified genetic algorithm. The BNN are applied to training patterns having nonlinear decision boundaries to confirm its classification performance.

## 1. 서 론

인간두뇌의 정보처리 기능을 모델링한 인공신경망은 분산된 형태의 정보저장, 가중치 학습에 의한 지식 획득, 정보의 병렬 분산처리와 같은 특징을 가지며 여러 경사면을 통한 학습에 의하여 기존의 방법으로는 해결하기 어려운 복잡하고 충분한 정보가 없는 문제에 대한 해결 가능성을 제시해 준다. 그러나 적용 대상에 대한 신경망 구조의 결정문제는 신경망이 가지고 있는 근본적인 제한점 중 하나이고, 경사하강법에 기초한 학습방법은 비유함수로서 순시오차를 사용하므로 학습결과가 지역 최소점에 수렴하는 문제가 발생할 수 있다. 또한 기존의 다층 신경망에서 오차역전파 학습알고리즘의 구현을 위하여 많이 사용하는 시그모이드 함수는 신경망의 하드웨어 구현을 제한하는 요소가 되기도 한다.

기존의 신경망의 단점을 극복하기 위해서는 신경망의 구조와 가중치의 최적탐색이 가능하여야 하고, 실제 문제에 대한 적용을 위하여 하드웨어 구현이 간단해야 한다. 뉴런간에 임의연결을 허용하는 형태의 신경망은 불필요한 노드가 없으므로 가장 효율적이라고

할 수 있으나 최적구조를 찾아내기 위한 탐색은 가중치의 탐색을 고려하지 않더라도 노드의 수가 증가함에 따라 지수함수 형태로 복잡도가 증가하므로 이를 해결하기 위하여 구조 최적화가 용이한 구조의 신경망 모델이 요구된다. 이러한 문제점들을 해결하기 위하여 신경망의 최적화 방법으로 자연계의 적자생존 원리에 아이디어를 둔 진화연산이 적용되고 있다. 진화연산은 문제에 따라 정의된 적합도 함수를 이용하여 전역 최적화를 시도한다[1][2]. 진화연산을 적용한 신경망 최적화 기법에는 먼저 기존의 경사하강법에 기반한 학습 방법을 그대로 사용하면서 신경망의 구조만을 사전지식에 의존하지 않고 진화연산을 이용하여 동적으로 변화시키는 방법에 관한 연구가 있다[3]. 또한 지역 최소점으로의 수렴을 피하기 위하여 고정된 신경망 구조 하에서 가중치를 최적화를 위하여 진화연산을 적용한 연구가 있으며[4][5], 구조와 가중치 모두를 진화 연산을 이용하여 최적화 시키는 연구도 수행되고 있다[6][7].

본 논문에서는 블록기반 신경망(Block-based Neural Network: BNN)을 제안한다. 블록기반 신경망은 위에서 언급한 신경망의 특징을 포함할 수 있도록 설계하

였다. 블록기반 신경망은 동적으로 내부구조를 변화시킬 수 있는 형태를 가지면서도 구조 최적화가 용이한 특징을 가지고 있다. 또한 FPGA 등의 재구성가능 하드웨어를 이용한 구현이 용이하게 하기 위하여 최소 단위인 블록들을 2차원으로 배열하여 전체 신경망을 구성하며, FPGA의 재구성 비트 스트링으로 변환할 수 있도록 신경망을 고정된 바이너리로 표현한다. 구조와 가중치를 함께 표현하여 동시에 최적화가 가능하도록 하고, 학습시 지역 최소점으로서의 수렴 문제를 해결하기 위하여 학습방법으로 블록기반 신경망의 최적화에 알맞도록 개체 표현방법과 유전 연산자 등을 변화시킨 유전자 알고리즘을 사용한다. 유전자 알고리즘으로 신경망을 최적화할 경우 바이너리 표현이 가능하며 기존의 에러 경사기반 최적화 알고리즘에 비하여 구조와 가중치를 동시에 최적화 하는 것이 가능하다.

2절에서는 블록기반 신경망 모델을 설명하고, 3절에서는 블록기반 신경망의 구조 및 가중치 최적화를 위한 유전자 알고리즘에 대해 설명한다. 4절에서는 비

선형 결정평면을 가지는 학습패턴들을 대상으로 하여 제한한 블록기반 신경망의 패턴분류 성능을 평가하였다.

## 2. 블록기반 신경망 모델

### 2.1 블록기반 신경망의 구조

블록기반 신경망 모델은 기본 구성요소인 블록들의 2차원 배열로 이루어진다. 그림 1은 기본 블록들로 이루어진 3×4 블록기반 신경망의 구조를 나타낸다. 상위층과 하위층은 각각 입력 및 출력층이고, 2개 이상의 중간층을 가질 수 있다. 중간층 블록의 출력은 인접한 4개 블록들과 직접 연결되어 있으며 블록들간의 화살표는 신호의 흐름을 나타낸다. 좌우측 끝에 위치한 블록들의 입출력은 서로 연결되어 있다.

각 블록  $b_{ij}$ 은 내부에는 4개의 입출력 뉴런으로 구성되어 있는 간단한 전방향 신경망으로 구성되며, 입출력 연결에 따라 그림 2와 같이 4가지 형태를 가질 수 있다. 블록들의 내부연결은 구조에서 정의된 입출력 형태에 따라 결정되며, 각 블록에서 사용되는 가중치는 재구성가능 하드웨어 구현을 용이하게 하기 위하여 정수값을 가진다. 그림 2에서  $x_i$ 는 입력,  $y_j$ 는 출력을 나타내는데, type A와 type B는 각각 입력/출력이 1/3 및 3/1인 경우이고, type C와 type D는 모두 2/2인 경우이나 내부 연결방법이 다르다.

입력 뉴런은 선형 활성화함수를, 출력 뉴런은 -1과 1의 제한된 범위 내에서 그림 3과 같이 정의된 대칭 포화선형 형태의 활성화함수를 사용한다. 시그모이드 함수 형태의 경우에는 지수부분에 의하여 하드웨어 구현이 제한을 받게 된다. 그러므로 활성화함수는 포화상태가 존재하는 불연속 함수를 사용해야 한다. 기존의 대칭포화선형 활성화함수의 기울기는 1이나, 정수 가중치 하에서 제한된 수의 학습패턴에 대한 비교적 연속적인 출력 공간을 생성해 내기 위하여 기울기의 값을 조절한다.

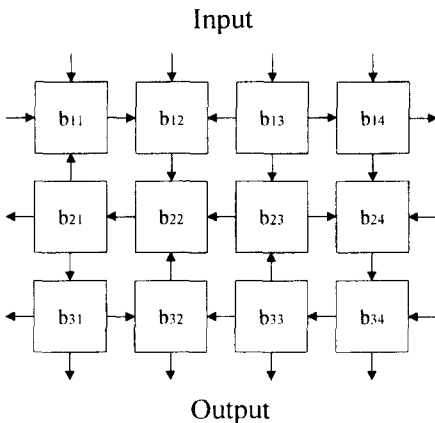


그림 1. 블록기반 신경망의 구조  
Fig. 1. Structure of the block-based neural network

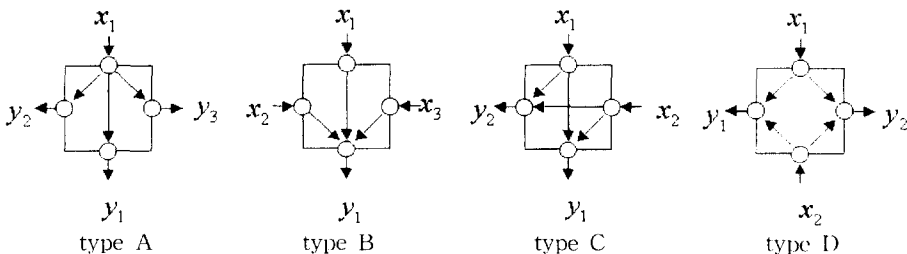


그림 2. 블록기반 신경망 기본블록의 내부 구조  
Fig. 2. Internal structure of basic blocks of the BNN

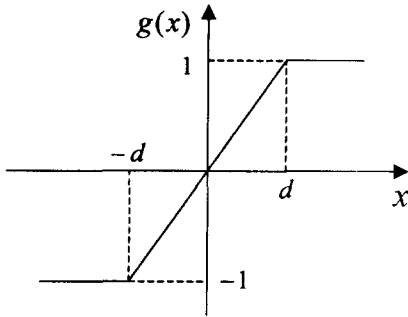


그림 3. 블록기반 신경망의 활성화함수  
Fig. 3. Activation function of the BNN

블록기반 신경망은 문제에 따라서 전방향적인 요소와 재귀적인 요소를 포함하는 형태로 변화가 가능하다. 이러한 블록기반 신경망은 시스템 입력에 대하여 출력이 안정화 되었을 때 이를 입력에 대한 블록기반 신경망의 출력으로 허용한다. 출력이 안정되는 시간은 블록기반 신경망의 가로방향 또는 세로방향 블록들 간의 재귀적 특성에 의하여 영향을 받는다. 블록간의 신호의 흐름은 각 블록들의 입출력 형태를 결정하고, 이는 블록들의 내부구조를 결정하게 된다.

### 2.2 블록기반 신경망의 특징

크기가  $m \times n$ 인 블록기반 신경망은 각 층의 뉴런수가  $n(n \leq 4)$ 인 임의의 다층 신경망 구조를 성공적으로 나타낼 수 있다. 예를 들어 3비트 홀수 패리티의 경우 입력층-중간층-출력층이 3-3-1의 전방향 다층 신경망을 사용하여 해결할 수 있는데,  $2 \times 3$  블록기반 신경망 역시 3비트 홀수 패리티 패턴을 성공적으로 분류할 수 있다. 이 논문에서 제시하는 블록기반 신경망의 가중치값은 모두 정수로 제한하였으므로 불연속적인 결정평면을 형성한다. 그러므로 XOR과 같이 학습 패턴의 수가 너무 적은 경우에는 진화 알고리즘을 사용한 학습의 경우 해의 급격한 변화를 통하여 학습이 이루어지는 경향이 발생한다. 블록기반 신경망은 전방향 또는 재귀적 특성을 표현할 수 있으므로 진화 알고리즘으로 학습시 주어진 문제에 적합한 최적구조를 획득할 수 있다. 또한 뉴런단위의 연결에 의한 구조진화의 복잡도에 비해 블록기반 신경망의 구조진화 문제는 상대적으로 복잡도가 낮다.

### 3. 블록기반 신경망의 학습을 위한 유전자 알고리즘

블록기반 신경망의 진화는 전체 시스템의 구조 최



(a) 블록기반 신경망의 개체 표현



(b) 각 블록의 2차원 코딩

그림 4. 블록기반 신경망의 2차원 인코딩  
Fig. 4. Two-dimensional encoding for the BNN

적화를 위한 각 블록의 입출력 형태 결정과 블록 내부의 가중치 결정으로 나눌 수 있는데, 유전자 알고리즘을 사용하여 동시에 최적화하였다. 유전자 알고리즘은 또한 전역탐색 특성과 적합도의 설정에 따른 목적지향 최적화가 가능하다. 블록기반 신경망의 최적화에 적합하도록 유전자 알고리즘의 인코딩과 재생산, 교배의 연산자를 다시 정의하고, 진화성능의 향상을 위하여 내부복제와 역치 연산자를 사용한다.

### 3.1 인코딩

블록간 정보의 전달방향은 각 블록의 입출력 형태를 결정하고 이는 블록의 내부구조를 결정하게 된다. 결국 제안한 신경망의 구조는 블록간 정보의 전달방향에 따라 결정된다. 따라서 구조와 가중치를 동시에 진화시키기 위해서는 블록간 정보 전달방향과 가중치를 인코딩 하여야 한다. 보통의 다차원 인코딩에서는 각각의 유전인자에 대하여 하나의 차원을 할당한다. 하지만 본 논문에서는 블록기반 신경망에 적용하는 유전자 알고리즘에서 주 연산자인 교배의 성능을 높이기 위하여 블록기반 신경망의 시스템 특성에 따라서 블록간 정보 전달방향과 가중치를 함께 2차원으로 인코딩 한다.

그림 4(a)는 가중치를 4비트로 표현한 블록의 인코딩과 이들을 조합한 블록기반 신경망 전체 시스템의 2차원 인코딩 형태를 나타낸다.

□는 가중치를, ■는 신호의 전달방향이 ↓과 ←인 경우 0으로, ↑과 →을 1로 정의한다. 입력과 출력 정보 전달방향은 항상 고정되어 있기 때문에  $m \times n$  블록으로 구성된 블록기반 신경망의 구조는  $(2m-1)n$  비트로 표현하고, 각 블록은 입출력수가 2/2 인 경우 바이어스를 포함하여 최대 6개의 가중치가 필요하기 때문에 1-비트로 표현한  $m \times n$ 블록기반 진화신경망의 가중치는 총  $6lmn$  비트로 표현한다. 인코딩된 개체에서 각 블록의 2진수로 변환된 가중치는 최하위 비트가 마주보도록 배열한다. 그림 4(b)는 하나의 블

록에 대한 2차원 인코딩 정보를 표현한다. 좌측 상단과 우측 상단에 첨가된 전달방향 표현비트는 인접한 블록과 공유하며 시스템에서 첫번째 행의 블록들은 우측 상단의 전달방향 표현비트를 0으로 가정하며, 마지막 행의 블록들은 우측 하단의 전달방향 표현비트를 0으로 가정한다. 연결강도  $w$ 는 연결이 끊어졌음을 의미하는 연결강도 0을 2개 포함하도록  $-(2^{l-1}-1)$ 에서  $2^{l-1}-1$ 의 범위 내에서 표현하였다.

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

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

식 (1)은  $l$ -비트로 표현된 양의 정수값을 가지는  $w_b$ 를 가중치  $w$ 로 변환해 주는 과정을 나타낸다.

### 3.2 유전연산자

#### 재생산

유전자 알고리즘에서 기본적으로 사용하는 재생산 방법은 룰렛 휠에 의한 확률적인 선택방법이다. 이는 적합도의 비율에 따른 확률에 의존하는 방법으로 몇 가지 약점을 가지고 있다. 먼저 진화초기에 상대적으로 적합도가 매우 큰 개체가 발생할 경우 이 개체는 재생산시 높은 생존확률을 가지며 이는 개체군의 조기수렴 문제를 야기시킨다. 또한 개체들이 해 부근으로 수렴한 경우에는 개체들의 적합도가 매우 유사하여 유전 연산자에 의한 개체들의 경쟁이 활발히 일어나지 않게 되고 결국 개체들은 해 주위를 맴돌게 된다. 이러한 문제점을 해결하기 위해서는 상대적으로 매우 큰 개체의 적합도는 낮추고, 개체들간의 적합도 차를 개체군의 분포 상태에 따라서 적절히 크게 조절해야 한다. 이러한 개념에서 룰렛 휠에 의한 확률적 선택방법의 약점을 개선하기 위하여 적합도 순위 (fitness ranking), 적합도 스케일링 (fitness scaling) 등의 방법이 있다[2]. 적합도 순위 방법은 개체군의 적합도 순위를 결정하고 이를 지수함수의 변수로 사용하는 방법이다. 이렇게 해서 얻어진 지수함수의 함수값은 개체의 적합도로 받아들인다. 이 방법은 적합도와 목적함수 간의 관계에 있어서 이론에 기반을 두지 않았으며, 개체군의 목적함수에 의한 적합도 분포를 고려하지 못한다는 단점을 가지고 있다. 적합도 스케일링 방법중 선형 스케일링 방법은 적합도의 최소, 평균, 최대값을 이용하여 개체군의 적합도 분포를 파악하고 이를 기반으로 하여 선형 함수에 의하여 적합도를 조절하는 방법이다. 이 방법은 개체군의 적합도 분포를 반영할 수 있으나 개체군의 적합도 평균이 최대

적합도에 근접할 경우 상대적으로 낮은 값의 적합도가 음으로 조절될 수 있다. 이를 해결하기 위하여 선형 적합도 스케일링의 경우 음으로 평가되는 적합도를 발견할 경우 스케일링 함수를 최대와 평균 적합도에서 평균과 최소 적합도에 의한 선형 함수로 변환한다. 이 경우 해 부근으로 진화가 이루어진 경우 개체군의 표류(genetic drift)가 발생하게 된다.

본 논문에서는 위에서 언급한 선형 적합도 스케일링 방법의 문제점을 개선하기 위하여 변형된 선형 스케일링 방법을 사용하였다. 선형함수에 의한 적합도 변환시 음의 적합도가 발생하지 않으면서, 해 부근에서의 개체군의 표류가 일어나지 않도록 개체군의 최대, 평균, 최소 적합도를 모두 이용하여 각기 다른 2개의 선형함수를 도입하여 적합도를 스케일링하였다.

·  $f_{\min} \geq f_s$  인 경우

$$f'(x_i) = a_1 x_i + b_1 \quad (3)$$

·  $f_{\min} < f_s$  인 경우

$$f'(x_i) = \begin{cases} f_1(x_i) = a_2 x_i + b_2 & \text{if } x_i < f_{avg} \\ f_2(x_i) = a_2 x_i + b_2 & \text{otherwise} \end{cases} \quad (4)$$

여기서 각 계수는 각각  $a_1 = (\alpha - 1)f_{avg}/(f_{max} - f_{avg})$ ,  $b_1 = f_{avg}(f_{max} - \alpha f_{avg})/(f_{max} - f_{avg})$ ,  $a_2 = f_{avg}/(f_{max} - f_{avg})$ ,  $b_2 = -f_{min}f_{avg}/(f_{max} - f_{avg})$ 이다. 식 (3)와 (4)는 각 경우에 대한 선형 스케일링 함수를 나타낸다.  $f(x_i)$ 는 목적함수에 의한  $i$ 번째 개체의 적합도,  $f'(x_i)$ 는 스케일링 과정을 거친 적합도를 나타낸다.  $f_{min}$ ,  $f_{avg}$ ,  $f_{max}$ 는 각기 목적함수에 의한 최소, 평균, 최대 적합도를 의미한다. 또한  $f_s$ 는  $f_{avg}$ 와  $f_{max}$  이용한 스케일링에서  $f_{min} < 0$ 인 경우를 확인하기 위한 기준값으로  $(\alpha f_{avg} - f_{max}) / (\alpha - 1)$ 으로 표현한다.  $\alpha$ 는 선형 스케일링 상수로서 선형 함수의 기울기에 영향을 미친다. 보통 작은 수의 개체를 사용할 때는  $\alpha = 1.2 \sim 2$ 의 값을 사용한다. 변형된 선형 스케일링 방법 이외에 재생산시 최대 적합도를 가지는 개체를 보존하기 위하여 엘리트스트 (elitist)를 사용한다.

#### 교배

2차원 인코딩한 개체에 적용하는 연산자 역시 이에 알맞도록 변형해야 한다. 다차원 교배의 경우 1차원 교배에서 3가지 확장 형태를 고려할 수 있다. 먼저 각 차원별로 1개의 교배점을 선택하는 방법이 있다. 다음으로 각 차원별로 고정된 개수의 교배점을 선택하여 교배연산을 수행하는 방법이 있는데, 첫 번째 방법의 확장형이라 할 수 있다. 마지막 방법은 각 차원에 대하여 선택하는 교배점  $k_i$ 의 총 수를  $\sum_{i=1}^n k_i = k$

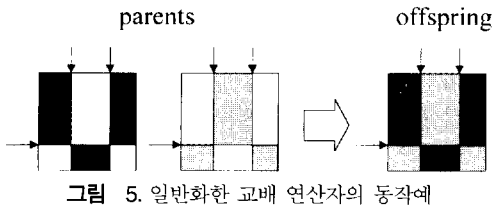


그림 5. 일반화한 교배 연산자의 동작예  
Fig. 5. Example of operation for generalized 2D crossover

( $k_i \geq 0$ )로 고정하는 것이다. 마지막 방법은 다른 방법들에 비하여 일반화된 형태이다. 가장 일반화된 형태의 교배연산에 의하여 각 차원별로 고정된 교배점을 선택하는 교배 연산에 비하여 보다 다양한 스키마를 허용할 수 있다.

그림 5는 일반화한 교배 연산자의 동작중  $k=3$ 이고 row 쪽으로 1, column 쪽을 2개의 교배점이 임의로 선택된 경우의 교배연산 예를 보여준다. 2차원 인코딩에 사용한 2차원 교배의 경우 1차원에서 알려진 바와 같이 교배점이 1개일 때는 진화 초기에서 개체의 다양성을 유지하며 2점 교배의 경우 개체의 수렴을 촉진하는 현상을 보인다.

**돌연변이**

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

**내부복제**

블록기반 신경망의 정보 전달은 전방향적인 성질과 재귀적인 성질을 가지고 있다. 따라서 진화의 성능을 개선하기 위해서는 이 두가지 성질을 모두 고려하여야 한다. 내부복제 연산자는 재귀적인 구조를 가지는 신경망의 진화성능을 개선시키는 역할을 한다[7]. 본 논문에서는 교배 후의 개체들이 충분한 다양성을 가지게 하기 위하여 교배 연산자를 보조하는 연산자로서 내부복제 연산자를 사용한다. 내부복제 연산자는 2차원으로 인코딩 된 개체군에서  $p_{ic}$ 에 의하여 선택된 개체를 각각에 대하여 일부를 다른 부분으로 복사하는 동작을 수행한다. 개체내의 복사 위치와 복사 정도는 임의로 결정한다.

**역치**

진화시 탐색 성능을 향상시키기 위하여 개체내의 저장 정보를 파괴하지 않는 연산자인 역치를 사용하였다. 역치는 확률  $p_r$ 에 의하여 선택된 개체에 대해서만

적용한다. 또한 가중치 탐색의 성능을 높이기 위하여 구조를 나타내는 유전인자에는 적용하지 않았다.

**3.3 적합도 함수**

유전자 알고리즘은 목적함수의 설정에 따른 최적화가 가능하다. 본 논문에서는 몇가지 패턴에 대한 분류 성능을 확인할 것이므로 최적화 목적으로 실제 출력이 목표 출력에 근접하는 것, 즉 에러가 줄어드는 것이다. 에러에 의한 적합도 함수 설정 방법은 크게 두 가지를 들 수 있다.

$$fitness = \frac{b}{a + \sum_j e_j^2(x)} \tag{5}$$

$$fitness = c - \sum_j e_j^2(x) \tag{6}$$

식 (5)와 (6)은 에러를 줄이는 것을 목표로 할 경우의 가능한 적합도 정의식을 나타낸다. 식 (5)의 경우  $a$ 와  $b$ 를 조절함으로써 진화 시점에 따른 불균일한 적합도 분포를 얻을 수 있다. 이는 재생산시 적합도에 따른 누적확률에 기반한 방법만을 사용하여도 빠른 수렴성을 보인다. 하지만 식 (5)와 같은 진화 시점에 따른 불균일한 적합도 평가는 진화 초기와 말기에 각기 개체들의 표류와 조기수렴 현상이 발생할 수 있다. 본 논문에서는 블록기반 신경망의 구조와 가중치를 동시에 최적화 하기 위하여 진화 시점에 관계없이 균일한 적합도 평가가 가능한 식 (6) 형태의 적합도 함수를 사용한다.

$$fitness = p_1 p_2 \left( Nm - \sum_{j=1}^N \sum_{k=1}^m e_{jk}^2 \right) \tag{7}$$

$$e_{jk} = d_{jk} - g_{jk}(x) \tag{8}$$

식 (7) 은 본 논문에서 사용한 적합도 함수를 나타낸다.  $N$ 는 학습 데이터의 수,  $m$ 은 don't care(\*)를 제외한 출력 블록의 개수,  $d_{jk}$ 와  $g_{jk}$ 는 각각  $j$ 번째 패턴에 대한  $k$ 번째 출력의 기대치와 실제출력을 나타낸다.  $p_1$ 과  $p_2$ 는 적절한 구조진화를 위한 패턴티 항이다.  $p_1$ 은 4개의 뉴런이 모두 입력/출력 뉴런인 블록이 존재하는 경우,  $p_2$ 는 시스템의 행 블록들 간의 정보가 출력에서 입력 방향으로만 전달되는 경우에 사용된다. 진화 초기에 적절한 구조 탐색과 진화속도 향상을 위하여 위와 같은 경우가 발생할 때  $p_1, p_2 = 0.9$ 로 설정하여 구조에 문제가 있다고 판단된  $i$ 번째 개체의 적합도를 감소시킨다. 이외의 경우에는  $p_1 = p_2 = 1$ 로 한다.

## 4. 블록기반 신경망을 이용한 패턴분류

### 4.1 실험 패턴

블록기반 신경망과 이의 최적화 방법의 성능을 확인하기 위하여 XOR, 3비트 패리티, 4비트 대칭 패턴을 사용한다. XOR은 1개 출력이 2개 입력값들의 배타적 OR로서 나타난다. 3비트 패리티는 3개의 입력 1개의 출력으로 구성되는 문제로서 입력 중 1의 개수가 홀수일 때 1을 출력하고 그렇지 않을 경우 0을 출력한다. 4비트 대칭 패턴은 상위 2개와 하위 2개의 입력이 서로 대칭일 때 1을 출력하는 문제이다. 블록기반 신경망의 입력은 0과 1사이로 정규화 하였으며, 출력은 각 블록을 구성하는 활성화수가 -1에서 1사이의 범위를 가지는 대칭포화선형 형태이므로 -1과 1로 구성한다.

### 4.2 유전자 알고리즘에 의한 파라미터 결정

블록기반 신경망의 최적화에 사용한 유전자 알고리즘은  $\alpha, p_c, p_{mw}, p_{mc}, p_{ic}, p_{iv}, n$ 의 모두 7개 파라미터를 가진다. 이들은 각기 재생산시의 스케일링 함수, 교배확률, 가중치의 돌연변이 확률, 구조의 돌연변이 확률, 내부복제 확률, 역치 확률, 마지막으로 진화시 사용한 개체의 수를 나타낸다. 진화시 조기수렴이 발생하지 않으면서 속도가 가장 빠르도록 파라미

표 1. 블록신경망의 최적화를 위하여 사용된 파라미터  
Table 1. Parameter settings for optimizing the BNN

|          |       |
|----------|-------|
| $a$      | 1.6   |
| $p_c$    | 0.35  |
| $p_{mw}$ | 0.001 |
| $p_{mc}$ | 0.005 |
| $p_{ic}$ | 0.002 |
| $p_{iv}$ | 0.002 |
| $n$      | 100   |

터를 설정하였다.

설정한 파라미터는 표 1과 같다.  $p_{ic}$ 와  $p_{iv}$ 은 스카마를 과도하게 파괴하지 않는 범위 내에서 탐색성능을 높이기 위하여 매우 낮은 수치를 적용하였으며  $n$ 의 경우 설정치보다 많이 사용할 경우 진화 세대수는 감소에 비하여 속도가 떨어지므로 적은 수의 개체를 사용한다.

그림 7은 세가지 패턴분류 문제에 대한 진화를 통한 블록기반 신경망의 최적화시의 적합도 경향을 나타낸다. 점선은 각 세대에서의 최대 적합도를, 실선은 평균 적합도를 나타낸다. 적합도 설정에 사용한 식 (7)과 같이 단일 출력 형태인 패턴들의 최대 적합도는 각 패턴이 가지는 학습 데이터의 개수와 같다. 세가지 문제에 대한 적합도 변화 경향 모두 초기 진화 시간에 비하여 진화 말기의 진화 종료 조건으로 사용한 최대 적합도를 99.9% 이상 만족시키기 위하여 진화 초기에 비하여 많은 탐색이 필요함을 알 수 있다. 적합도를 식 (5)와 같은 형태로 설정할 경우 진화 말기의 보다 나은 진화 말기의 탐색 성능을 보인다. 하지만 진화시의 조기수렴 현상을 초대한 억제하기 위하여 사용하지 않았다. 진화 후 획득한 블록기반 신경망을 이용하여 얻은 평균제곱오차는 XOR의 경우 0, 3비트 패리티의 경우 0.029, 그리고 4비트 대칭의 경우 0.062이다.

그림 6은 세가지 실험 패턴에 대한 진화시 적합도 경향을 나타낸다. 적합도 설정 방법의 특성상 99.9% 이상의 적합도를 만족하도록 진화를 수행하였으나 실제 출력이 목표치와 보이는 차이는 비교적 크다. 이는 정의한 적합도 함수의 형태가 전 구간에서 균일한 적합도 분포를 갖기 때문이며, 적합도가 높은 부분에 대하여 적합도 사이의 간격을 넓힘으로서 이 문제를 해결할 수 있다.

### 4.3 블록기반 신경망에 의한 패턴분류 결정평면

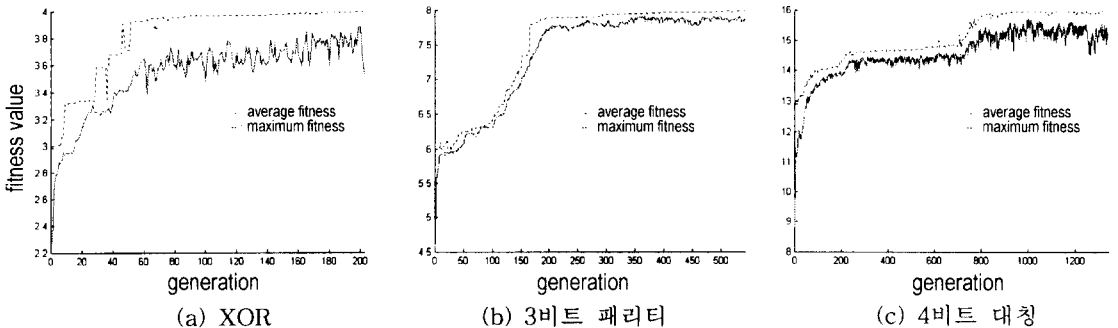


그림 6. 진화시 적합도 경향  
Fig. 6. Trend of fitness values

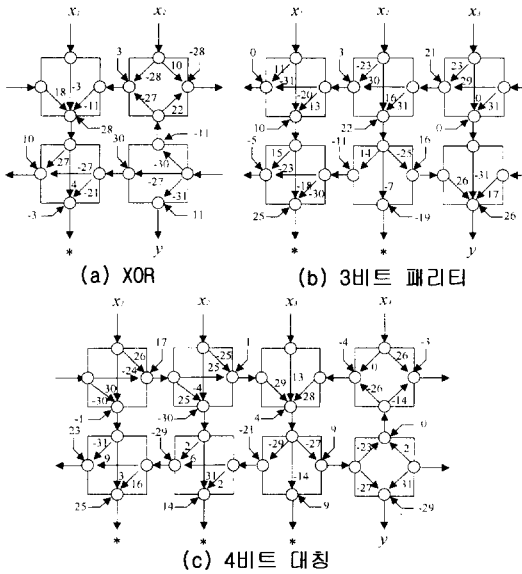


그림 7. 최적화된 블록기반 신경망의 구조와 가중치  
 Fig. 7. Structure and weights of the BNN optimized with the genetic algorithm

3가지 다른 학습패턴에 대해서 유전자 알고리즘을 이용하여 최적화된 블록기반 신경망의 구조와 가중치는 다음과 같다. 블록기반 신경망의 크기는 입력의 갯수에 의해 결정하였으며, 출력노드는 임의로 가장 오른쪽 노드로 하였다. 가중치외에 표시한 숫자는 출력 뉴런에서 사용된 바이어스 항을 나타낸다.

그림 7은 진화후 획득한 각 패턴에 최적화된 블록기반 신경망의 구조와 가중치를 나타낸다.  $m \times n (n \leq 4)$  블록기반 신경망은  $2m-1$ 의 입력, 중간, 출력층을 가지는 전방향 다층 신경망과 같은 구조를 표현할 수 있음을 알 수 있다. 세가지 패턴을 분류할 수 있는 블록기반 신경망의 형태는 기존의 전방향 다층 신경망과 큰 차이가 있다. 예로 XOR의 경우 획득한 블록기반 신경망의 구조는 기존의 전방향 다층 신경망에서 XOR을 해결할 수 있다고 알려져 있는 2-2-1과 다른 구조를 가진다. 또한 제한된 범위의 정수 가중치만을 사용하여 실험 패턴에 대한 분류가 성공적으로 이루어졌음을 알 수 있다.

그림 8은 유전자 알고리즘에 의하여 최적화된 블록기반 신경망에 의한 결정평면 중 2차원 상에 표현가능한 XOR문제의 결정평면을 나타낸 것이다. 2개의 부류를 어려없이 잘 분류하였다는 것을 알 수 있다. 오차 역전과 학습 알고리즘을 사용하여 4개의 XOR 학습패턴을 학습하여 얻은 2-2-1 전방향 다층 신경망에 의한 결정평면과 유사한 형태를 보여준다.

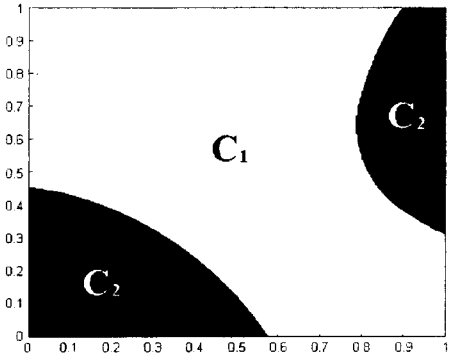


그림 8. XOR 패턴의 결정평면  
 Fig. 8. Decision Boundary for XOR Pattern

### 5. 결 론

본 논문에서는 재구성가능 하드웨어에 의하여 구현이 용이한 블록기반 신경망 모델을 제안하고 유전자 알고리즘에 의하여 구조와 가중치를 전역탐색 하였으며, 설계된 블록기반 신경망을 여러 가지 패턴분류에 적용하여 패턴분류 성능을 확인하였다. 이를 위하여 블록의 구조와 가중치를 함께 2차원으로 코딩하였고, 재생산 과정에서 변형된 스케일링과 엘리티스트를 사용하였다. 2차원으로 코딩된 개체에 적합하도록 일반화된 2차원 교배 연산자를 적용하였고, 가중치와 구조에 각기 다른 돌연변이 확률 적용하였다. 성능의 향상을 위하여 내부복제 연산자와 역치 연산자를 진화에 적용하였다.

블록기반 신경망의 패턴분류 성능을 시험하기 위하여 역전과 알고리즘의 성능 평가에 사용되었던 패턴 중 일부를 사용하였다. 이들을 통하여 학습방법으로 사용한 유전자 알고리즘에서 나타나는 최종적으로 획득한 구조와 가중치에 대한 분석, 적합도 경향, 패턴 분류가 끝난 경우의 결정평면에 대하여 분석하였다. 제안한 블록기반 신경망과 이의 학습방법이 적용 패턴을 성공적으로 분류해 낼 수 있었다.

### 감사의 글

이 연구는 1998년 (국책)연구개발사업의 지원에 의해서 수행되었습니다.

### 참고문헌

[1] T. Bäck, U. Hammel, and H.-P. Schwefel, "Evolutionary Computation: Comments on the History and Current State," *IEEE Trans. on*

*Evolutionary Computation*, Vol. 1, No. 1, pp. 3-17, 1997.

- [2] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, 1989.
- [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] 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.
- [5] P. G. Korning, "Training Neural Networks by means of Genetic Algorithms working on very long chromosomes," *Int'l J. of Neural systems*, Vol. 5, No. 3, pp. 299-316, 1995.
- [6] 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.
- [7] 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.
- [8] T. N. Bui and B. R. Moon, "On Multi-Dimensional Encoding/Crossover," *Proc. 6th International Conf. on Genetic Algorithms(ICGA)*, pp. 49-56, 1995.



#### 문 상 우 (Sang-Woo Moon)

1998년 : 숭실대학교 전기공학과 졸업 (공학사)

1998년~현재 : 숭실대학교 전기공학과 대학원

주관심분야 : 인공생명, 진화연산, 로봇틱스



#### 공 성 곤 (Seong-Gon Kong)

1982년 : 서울대학교 공과대학 전기공학과 졸업 (공학사)

1987년 : 서울대학교 대학원 전기공학과 졸업 (공학석사)

1991년 : 미 University of Southern California 전기공학과 졸업 (공학박사)

1992년 3월~현재 : 숭실대학교 전기공학과 부교수, 전기학회 편 집위원 및 연구회 간사장, 퍼지 및 지능시스템학회 논문지 편집위원장

주관심분야 : 지능시스템이론, 영상 및 신호처리, 로봇틱스, 진화연산, 인공생명.