

# 유전 알고리즘에서의 자기 조직화 신경망의 활용 (New Usage of SOM for Genetic Algorithm)

김정환<sup>†</sup>      문병로<sup>\*\*</sup>  
(Jung-Hwan Kim)    (Byung-Ro Moon)

**요약** 자기 조직화 신경망 (SOM: Self-Organizing Map)은 자율 학습 신경망으로 사전 지식이 존재하지 않는 자료에 존재하는 구조적 관계성을 보존하는데 이용된다. 자기 조직화 신경망은 벡터 양자화, 조합 최적화, 패턴 인식과 같은 복잡한 문제 해결을 위한 연구에 많이 이용되어 왔다. 이 논문에서는 좀더 효율적인 유전 알고리즘을 얻기 위한 스키마 변환 도구로서 자기 조직화 신경망을 이용하는 새로운 사용법에 대해서 제안한다. 즉, 각 자식해는 탐색 공간에서 좀더 바람직한 모양을 가지는 동질의 인공 신경망으로 변환된다. 이 변환으로 인해 강한 상위(epistasis)를 가지는 유전자들은 염색체 상에서 서로 인접하게 되는 것이다. 실험 결과는 기존 결과에 비해서 주목할만한 성능 개선이 있음을 보여준다.

**키워드** : 자기 조직화 신경망, 유전 알고리즘, 순환 신경망, 동질 신경망, 변환

**Abstract** Self-Organizing Map (SOM) is an unsupervised learning neural network and it is used for preserving the structural relationships in the data without prior knowledge. SOM has been applied in the study of complex problems such as vector quantization, combinatorial optimization, and pattern recognition. This paper proposes a new usage of SOM as a tool for schema transformation hoping to achieve more efficient genetic process. Every offspring is transformed into an isomorphic neural network with more desirable shape for genetic search. This helps genes with strong epistasis to stay close together in the chromosome. Experimental results showed considerable improvement over previous results.

**Key words** : Self-Organizing Map, Genetic Algorithm, Recurrent Neural Network, Isomorphic Neural Network, Transformation

## 1. 서론

다층형 인공 신경망은 패턴 인식 문제에 광범위하게 이용되어 왔다[1,2]. 역전파 알고리즘(BP: Backpropagation)은 다층형 인공 신경망을 위한 지도 학습 기법으로 가장 유명하고 많이 이용되어진 알고리즘이다[1]. 그러나 역전파 알고리즘은 지역 경사도를 이용하는 방법으로 탐색 능력의 한계를 가지고 있다. 이 알고리즘은 인공 신경망(NN: Neural Network)의 정보 처리 능력을 개발하는 지역 계산법에 기반을 하고 인공 신경망을 학습 시키는데 있어서 전역 정보를 이용하지 않는다.

인공 신경망 최적화를 위한 유전 알고리즘에 대한 연구는 많이 진행되어 왔다[3-5]. 유전 알고리즘을 이용한 인공 신경망 최적화를 위해서는 문제를 표현하는 방법

이 필요하다. 유전 알고리즘에서 신경망을 표현하기 위한 방법은 크게 직접 인코딩과 간접 인코딩으로 나눌 수 있다. 직접 인코딩은 신경망의 가중치, 연결 정보 등과 같은 인공 신경망의 매개변수를 염색체 상에 직접적으로 표현하는 인코딩 정책을 말한다. 이 개념은 신경망 표현을 위하여 신경망의 구성을 나타내는 규칙이나 정보를 염색체 상에 나타내는 간접 인코딩과는 상반되는 개념이다. 이 연구에서는 직접 인코딩 방법을 사용한다. 신경망을 위한 대부분의 직접 인코딩 방법은 연결 기반 인코딩이다. 이 방법은 미리 정의된 구조를 가지는 신경망의 가중치들을 최적화하는데 많이 이용된다. 연결 기반 인코딩의 가장 기본적인 방법은 신경망 구조의 모든 가중치들을 비트 문자열로 나타내는 이진 가중치 인코딩이다. 그림 1은 이진 가중치 인코딩의 예를 보이고 있다. 이 인코딩에서 각 가중치들은 인덱스 비트와 가중치 비트를 이용하여 표현된다. 인덱스 비트는 뉴런 사이의 간선이 가중치를 가지는지의 여부를 나타내고, 가중치 비트는 실제 가중치 값을 이진수로 표현한다. 이진 가중

<sup>†</sup> 학생회원 : 서울대학교 컴퓨터공학부  
jhkim@soar.snu.ac.kr

<sup>\*\*</sup> 정회원 : 서울대학교 컴퓨터공학부 교수  
moon@soar.snu.ac.kr

논문접수 : 2005년 4월 23일  
심사완료 : 2006년 2월 17일

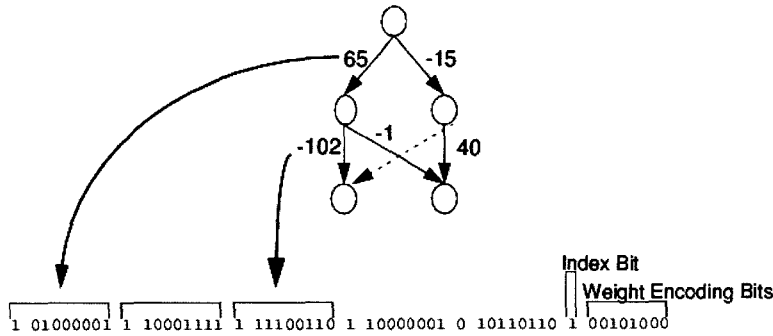


그림 1 이진 가중치 인코딩의 예제

치 인코딩은 가중치 최적화 문제나 신경망 구조 최적화 문제에서 사용된다. 가중치 최적화 문제에서는 인코딩 중에서 가중치 비트만을 사용하고 구조 최적화 문제에서는 인덱스 비트를 포함한 전체 인코딩을 사용한다. 하지만 이진 가중치 인코딩을 사용하는 유전 알고리즘에서의 문제는 주어진 인공 신경망을 표현하기 위하여 매우 긴 문자열을 사용하는 경우가 발생한다는 것이다. 다른 직접 인코딩 방법인 뉴런 기반 인코딩은 연결 기반 인코딩의 단점을 보완하는 인코딩 방법으로 이름에서 들어나는 것처럼 염색체는 가중치를 가지는 것이 아니라 뉴런에 대한 정보를 포함한다. 그 외에도 층 기반 인코딩, 방향 기반 인코딩이 있다[6]. 대부분의 유전 알고리즘을 이용하는 연구들은 이진 가중치 인코딩을 사용했다. 즉, 인공 신경망의 모든 가중치들은 선형 염색체에서 하나의 자리를 차지하게 된다. 그러나 선형 표현법은 유전자의 지역적 연결성을 반영하는데 있어서 주어진 정보를 모두 표현하지 못하는 단점을 가지고 있다[7,8]. 이 문제를 해결하기 위하여 이차원 인코딩이 유전자의 지역적 연결성을 좀더 잘 표현하기 위해 사용되었다[7-10]. 인공 신경망의 가중치들은 이차원 행렬로 표현 가능하므로 이차원 인코딩으로 표현하는 것이 자연스럽다. 이 논문에서는 유전 알고리즘을 위한 표현법으로 이차원 인코딩을 사용하고 인공 신경망 최적화([11])와 그래프 분할 문제([10,12,13])에서 좋은 성능을 보인 이차원 지리적 교차 연산자(2-dimensional geographic crossover)를 사용한다.

일반적인 다층형 인공 신경망은 입력층, 은닉층, 출력층으로 구성되어 있다. 은닉층의 뉴런들은 입력 데이터로부터 의미 있는 특징들을 추출함으로써 신경망이 복잡한 작업을 학습할 수 있게 하고 입력 데이터들을 분류하는 규칙을 생성한다. 인공 신경망이 이차원 인코딩으로 표현되고 인공 신경망의 구조가 고정된다면 인공 신경망의 유전자형(genotype)은 신경망상에 존재하는 뉴런들의 순서에 따라 결정된다. 신경망에 존재하는 뉴

런들의 연결 관계만을 고려한다면 은닉 뉴런들은 모두 같은 성질을 갖게 된다. 하지만 연결 관계에 가중치가 포함되면 은닉 뉴런들은 더 이상 같은 성질을 갖는 것이 아니라 서로 다른 성질을 갖게 된다. 어떤 은닉 뉴런들은 임의의 은닉 뉴런들의 쌍보다 강한 관계성을 가진다. 그러나 은닉 뉴런들 사이에 존재하는 관계성은 뉴런들의 위치만으로는 쉽게 드러나지 않는다. 은닉 뉴런들의 위치는 은닉 뉴런들이 가중치를 가지기 전에 결정되므로 강한 관계성을 가지는 뉴런들이 서로 인접한 위치를 갖는다는 것을 보장할 수 없다. 이런 현상은 고품질의 스키마들이 유전 알고리즘의 진행 과정 동안 살아 남지 못하게 한다. 이 논문에서는 각 인공 신경망을 뉴런들이 다른 위치를 가지지만 동일한 성질을 가지는 동질의 인공 신경망(Isomorphic Neural Network)으로 변환한다. 그리하여 인공 신경망의 좀더 나은 표현법을 유도하여 좋은 성능을 가지는 유전 알고리즘을 개발하는 것을 목적으로 한다. 이 변환을 위해 이 연구에서는 자기 조직화 신경망을 이용한다.

자기 조직화 신경망은 자율 학습 신경망으로 사전 지식이 존재하지 않는 데이터에서 구조적 연관성을 추출하고 보존하는데 이용된다. 자기 조직화 신경망은 벡터 양자화, 조합 최적화, 패턴 인식 문제와 같은 복잡한 문제의 해결에 많이 이용되어 왔다. 이 논문에서는 스키마를 변화하는 도구로서의 자기 조직화 신경망 용법을 제안한다. 이 변환을 통해 좀더 효율적인 유전 알고리즘의 개발을 유도하는 것이다.

이 논문의 나머지 부분은 다음과 같은 구성을 가진다. 다음 절에서는 강한 연관성을 가지는 뉴런들을 군집화해야 하는 필요성과 자기 조직화 신경망의 특성에 대해서 기술한다. 3절에서는 유용한 빌딩 블록이 유전 알고리즘의 진행 과정에서 살아남도록 하는 변환 법에 대해서 설명한다. 4절은 인공 신경망의 성능 개선을 위해 제안된 알고리즘에 대해 묘사한다. 5절에서는 실험 결과를 보이고 6절에서는 결론을 제시한다.

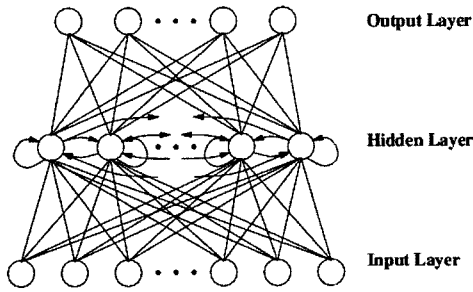


그림 2 순환형 인공 신경망 구조

## 2. 배경 지식

### 2.1 신경망 구조

이 연구에서는 Elman의 순환형 인공 신경망 (RNN: Recurrent Neural Network)[14]에 기반한 순환형 인공 신경망 구조를 사용한다. 이 구조는 그림 2와 같이 입력층, 은닉층, 출력층의 세 개의 층으로 구성되어 있다. 각각의 은닉 뉴런들은 자기 자신과 연결되어 있고 인공 신경망 은닉층의 모든 다른 뉴런들과 연결되어 있다. 이 연결들은 역전파 알고리즘에 의해 학습되고 가중치가 변하게 된다.

### 2.2 자기 조직화 신경망의 위상 순서화

자기 조직화 신경망은 고차원 데이터의 가시화를 위한 효율적인 도구이다. 자기 조직화 신경망은 고차원 데이터에 존재하는 복잡하고 비선형적인 통계적 연관성을 저차원 표현법을 이용하여 단순한 기하학적 연관성으로 표현한다[15]. 자기 조직화 신경망은 주어진 데이터의 가장 중요한 위상적 연관성을 보존하면서 데이터에 존재하는 정보를 압축하기 때문에 주어진 데이터에 대한 요약으로 간주할 수 있다. 이 두 가지 특성인 가시화와 요약은 다양하고 복잡한 문제에 수없이 이용될 수 있다. 자기 조직화 신경망으로 인해 주어진 데이터는 의미를 가지는 이차원 순서로 재조직 된다. 여기서 유사한 성질을 가지는 데이터들은 서로 인접하도록 조정된다. 이런 의미에서 자기 조직화 신경망을 유사성 그래프 또는 군집 그림으로 볼 수 있다[15].

### 2.3 유전자 재배열

대부분의 유전 알고리즘에서 각 유전자의 위치는 명시적인 의미를 갖는다. 이런 종류의 인코딩을 위치 기반 (locus-based) 인코딩이라 한다. 유전자 재배열에 관한 많은 연구가 시행되어 왔다. 그 중 Bagley는 무작위로 유전자의 위치를 변경하려는 시도를 했다[16]. 또한 Bui와 Moon은 그래프에 존재는 유용한 군집 정보를 추출하려고 노력하여 선형 인코딩을 사용하는 그래프 분할 문제에서 유전자를 재배열함으로써 유전 알고리즘의 성능이 개선될 수 있음을 보였다[17,18]. Kim과 Moon은

유전 알고리즘에서 인공 신경망 표현을 위한 재배열 방법을 제안했다[11]. 그들은 강한 연관성을 가지는 뉴런들을 연결한 간선들은 간선의 집합에서 무작위로 선택된 간선들보다 강한 유형을 가질 거라고 믿었다. 그들은 유전자 사이에 존재하는 연관성을 측정하기 위해 간선이 가지는 가중치의 상대적 크기를 이용하였다. 또한 이 방법이 유전 알고리즘의 성능을 개선함을 보였다.

## 3. 동질 신경망으로의 변환

신경망의 은닉층 뉴런들은 서로 완전 연결되어 있기 때문에 동등한 기능을 가지는 수많은 은닉층 뉴런들의 순서가 존재한다. 이 모든 신경망 구조는 동일한 인공 신경망을 나타내지만 유전 알고리즘에서는 다른 유전자형을 갖게 된다. 이 중 어떤 순서는 유전 알고리즘의 탐색 능력면에서 다른 순서보다 많은 장점을 가진다. 그림 3은 이차원 인코딩을 사용하는 예제를 보이고 있다. 그림에서 세 개의 뉴런  $j$ ,  $i$ ,  $i+1$ 이 서로 강한 연관성을 가지고 있다고 가정하자. 이 뉴런들과 연관된 가중치들은 사각형을 이용하여 표시되었다. 이 사각형들의 집합은 고품질을 가지는 스키마를 생성할 것이다. 지리적 교차 연산자가 이차원 인코딩 환경에서 새로운 스키마를 생성하는 능력이 뛰어나더라도 인코딩 상에서의 유용한 특징들이 군집되어 있지 못하다면(그림 3의 왼쪽 인코딩을 참조하라) 그에 상응하는 스키마는 대부분의 교차 연산 후에 생존하기 어려울 것이다. 그러므로 주어진 신경망의 구조를 유용한 군집 블록을 포함하는 기능적으로 동질인 구조로 변환해야 한다는 것은 중요하다. 여기서 왼쪽의 가중치 행렬에서 뉴런  $j$ 와  $i-1$ 의 위치를 교환한다면 스키마는 생존에 더 적합한 모양을 가진 더 나은 스키마로 변환된다(그림 3의 오른쪽 가중치 행렬을 보라). 이 논문의 목적은 고품질의 스키마를 생존에 유리한 동질의 또 다른 신경망의 스키마로 변환하는 것이다. 강한 연관성을 가지는 뉴런들이 군집화 되어 있을수록 모델링해야 할 복잡한 시스템에 대한 근사 작업이 더 쉽게 이루어진다. 이 연구에서는 뉴런의 군집화를 위해 자기 조직화 신경망의 특성을 이용한다.

### 3.1 신경망 가중치에 기반한 변환

각 은닉층 뉴런들은 가중치의 벡터를 갖고 뉴런에 연결된 가중치들에 따라 서로 다른 뉴런들에 영향을 준다. 가중치가 큰 값을 가지는 간선은 다른 뉴런을 강하게 자극한다. 반대로 아주 작은 음의 값을 가지는 간선은 다른 뉴런에 대해서 강하게 억제한다. 만약 어떤 뉴런이 강하게 다른 뉴런을 자극하거나 억제한다면 이 뉴런들은 서로 강하게 연관성을 갖고 있다고 판단 할 수 있다. 하지만 뉴런들 사이의 상대적인 연관성을 분석하는 일

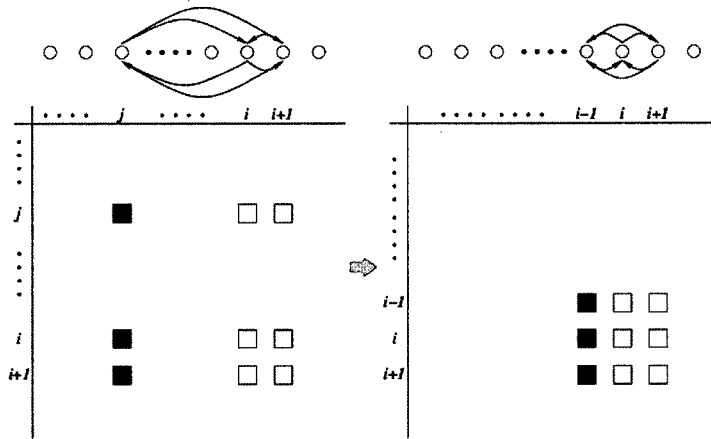


그림 3 변환 후의 스키마의 변화

은 어려운 일이며 또한 그에 상응하는 뉴런들 사이의 공간적 위치를 분석하고 판단하는 일은 상당히 어려운 일이다. 이 연구에서는 자기 조직화 신경망을 이용하여 인공 신경망의 위상을 변환한다. 이 작업은 유전 알고리즘이 진행되는 동안 인공 신경망의 은닉층 뉴런들 사이에 존재하는 연관성을 좀더 효율적으로 보전하기 위한 바램에서 출발한다.

자기 조직화 신경망은 순환 신경망의 은닉층 뉴런들에 연관된 가중치를 이용하여 학습된다. 자기 조직화 신경망은 이 가중치들의 연관성을 반영하는 유사도 그래프를 생성하는 것이다. 자기 조직화 신경망에 입력되는 입력 데이터 쌍의 개수는  $p$ 이다. 즉, 자기 조직화 신경망은 데이터 쌍  $\langle \mathbf{x}_j, d_j \rangle$ 을 학습 시킨다. 여기서  $\mathbf{x}_j$ 는 순환 신경망의 뉴런  $j$ 에 연결된 가중치들의 집합이고  $d_j$ 는 뉴런의 라벨, 즉  $j$  값을 갖는다. 자세한 내용은 다음과 같다.

$$\mathbf{x}_j = \mathbf{w}_j, \quad d_j = j \quad (1 \leq j \leq p)$$

$$\mathbf{w}_j = \langle w_{j1}^{(0)}, \dots, w_{jn}^{(0)}, w_{j1}^{(1)}, \dots, w_{jp}^{(1)}, w_{j1}^{(2)}, \dots, w_{jq}^{(2)} \rangle$$

where

$w_{jk}^{(0)}$ 는 입력층 뉴런  $k$ 와 은닉층 뉴런  $j$ 사이의 가중치

$w_{jk}^{(1)}$ 는 은닉층 뉴런  $k$ 와 은닉층 뉴런  $j$ 사이의 가중치

$w_{kj}^{(2)}$ 는 은닉층 뉴런  $j$ 와 출력층 뉴런  $k$ 사이의 가중치

$n$ 는 순환 신경망 입력층 뉴런의 개수

$p$ 는 순환 신경망 은닉층 뉴런의 개수

$q$ 는 순환 신경망 출력층 뉴런의 개수

다음은 신경망의 가중치에 기반한 변환에 대해서 자세히 설명하려 한다.

입력: 뉴런들 사이의 가중치를 가지고 있는 순환 신경망

출력: 은닉층 뉴런들의 일차원 순서  $\Pi$

$$\Pi = \langle \pi_1, \dots, \pi_p \rangle, \quad \pi_i \in \{1, 2, \dots, p\}$$

1. 순환 신경망의 은닉층을 자기 조직화 신경망의 경쟁층에 대응시켜  $\mathbf{x}_j$ 와  $d_j$ 를 생성한다. 즉, 은닉층 뉴런의 수는 자기 조직화 신경망의 경쟁층 뉴런의 수와 같다. 그리고 순환 신경망의 가중치들은 자기 조직화 신경망의 입력 벡터로 이용된다. 그러면 자기 조직화 신경망은 입력층에  $n+p+q$ 개의 뉴런을 가지게 되고, 경쟁층에  $p$ 개의 뉴런이 존재한다.
2.  $\mathbf{x}_j$ 와  $d_j$ 로부터 집합  $\mathbf{D} = \{ \langle \mathbf{x}_1, d_1 \rangle, \dots, \langle \mathbf{x}_p, d_p \rangle \}$ 를 생성한다.
3. 자기 조직화 신경망 SOM을 초기화 한다.
4. 자기 조직화 신경망은 순환 신경망의 은닉층 뉴런들에 연관된 가중치를 이용하여 학습된다. 즉, 입력 데이터 쌍의 집합  $\mathbf{D}$ 를 이용하여 SOM을 학습 시킨다.
5. 자기 조직화 신경망은 이 가중치들의 연관성을 반영하는 유사도 그래프를 생성한다. 유사도 그래프로부터 새로운 은닉층 뉴런들의 선형 순서  $\Pi = \langle \pi_1, \dots, \pi_p \rangle$ 를 생성할 수 있다.
6. 선형 순서가 생성된 후에 이 순서에 따라 순환 신경망의 은닉층 뉴런들은 재배치된다. 은닉층 뉴런  $i$ 의 라벨은  $\pi_i$ 이다. 때때로 몇몇 경쟁층 뉴런들은 동일한 라벨을 갖기도 한다. 변환 되기 전의 다른 은닉층 뉴런들에 연결된 가중치들이 같은 분포를 나타낼 때 이런 현상이 발생한다. 이 경우, 이 연구에서는 경쟁층 뉴런에 같은 라벨을 할당하였다. 이렇게 함으로써 인공 신경망이 가중치 공유(weight-sharing[19])의 효과를 얻을 수 있게 된다.

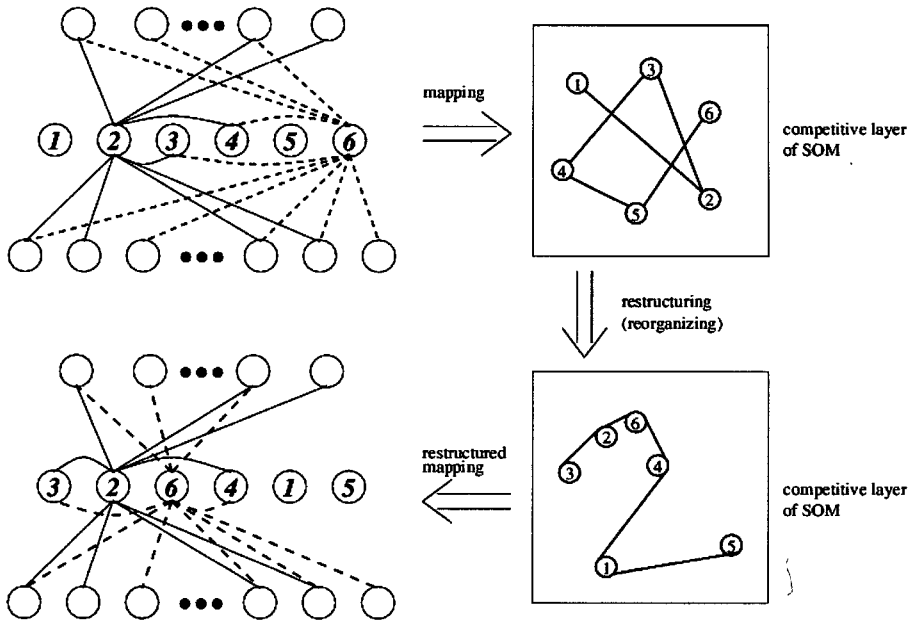


그림 4 순환 신경망과 자기 조직화 신경망 사이의 매핑

그림 4는 가중치에 기반한 변환의 예제를 보이고 있다. 이 그림에서, 실선과 점선은 중요한 역할을 하는 가중치를 나타낸다. 자기 조직화 신경망은 주어진 순환 신경망(그림 4의 왼쪽 위의 인공 신경망)의 가중치들을 학습하고 경쟁층 뉴런들의 순서  $\Pi = (3, 2, 6, 4, 1, 5)$  를 출력한다. 자기 조직화 신경망의 경쟁층에 존재하는 각 뉴런의 위치는 순환 신경망의 은닉층 뉴런들 사이의 상대적인 관계성을 표현한다. 강하게 서로 상관 관계를 유지하고 있던 뉴런 2와 6은 변환되기 전에는 서로 멀리 떨어져 위치했다. 하지만 변환 후에는 서로 인접하도록 위치를 하고 있다.

3.2 신경망 경사도에 기반한 변환

역전파 알고리즘을 이용하여 인공 신경망을 학습시킨 후에 신경망은 학습된 가중치들을 갖게 된다. 각 가중치들은 학습 과정을 통해 경사도를 갖게 된다. 이 경사도에 따라 모든 가중치들은 분류될 수 있다. 경사도 값이 클수록 불안정하다고 볼 수 있다. 안정된 가중치를 가지고 있는 간선은 학습 과정에서 영향의 정도가 작다. 하지만 불안정한 가중치를 가지고 있는 간선들은 학습 과정을 복잡하게 하고 학습의 정확도를 떨어뜨릴 수 있다. 이런 관점에서 출발하여 은닉층 뉴런들을 이 뉴런들과 연관된 가중치의 경사도에 따라 새롭게 재조직하려 한다. 경사도에 기반한 변환에서도 안정 상태인 뉴런들을 군집화하기 위하여 자기 조직화 신경망의 위상 순서화 성질을 이용한다. 결국에 은닉층 뉴런들은 안정도에 따

라 재정렬되고 좀더 안정된 뉴런들끼리 군집화하게 된다. 이렇게 안정화되어 군집된 뉴런들은 교차 연산자 후에도 생존할 확률이 높게 되는 것이다. 인공 신경망을 학습시킨 후에 각 은닉층 뉴런들은 다음과 같은 경사도 정보를 갖게 된다.

- $\delta_j^{(1)} \cdot y_i^{(0)}$  : 은닉층 뉴런  $j$ 와 입력층 뉴런  $i$ 사이의 경사도
- $\delta_j^{(1)} \cdot y_l^{(1)}$  : 은닉층 뉴런  $j$ 와 은닉층 뉴런  $l$ 사이의 경사도
- $\delta_k^{(2)} \cdot y_j^{(1)}$  : 출력층 뉴런  $k$ 와 은닉층 뉴런  $j$ 사이의 경사도

where

$$\delta_k^{(2)} = (t_k - y_k^{(2)}) \cdot \varphi'(\xi_k^{(2)})$$

$$\delta_j^{(1)} = \left( \sum_{k=1}^q \delta_k^{(2)} w_{kj}^{(2)} \right) \cdot \varphi'(\xi_j^{(1)})$$

- $w_{kj}^{(2)}$  : 은닉층 뉴런  $j$ 와 출력층 뉴런  $k$ 사이의 가중치
- $t_k$  : 출력층 뉴런  $k$ 의 목적값
- $\xi_j^{(1)}$  : 은닉층 뉴런  $j$ 의 중간 출력값
- $\xi_k^{(2)}$  : 출력층 뉴런  $k$ 의 중간 출력값
- $\varphi(x)$  : 시그모이드 활성화 함수  $1/(1+e^{-x})$
- $y_i^{(m)} = \varphi(\xi_i^{(m)})$

위에서 언급한 각 은닉층 뉴런의 세가지 경사도 정보는 자기 조직화 신경망을 이용한 변환의 입력 데이터로 이용된다. 그러므로 자기 조직화 신경망에 입력된 입력 벡터의 차원은  $(n + p + q)$ 이다. 경사도를 이용한 변환은 다음과 같은 데이터 쌍  $\langle \mathbf{x}_j, d_j \rangle$ 을 학습한다.

$$\mathbf{x}_j = \delta_j, \quad d_j = j \quad (1 \leq j \leq p)$$

$$\delta_j = \langle \delta_j^{(1)} y_1^{(0)}, \dots, \delta_j^{(1)} y_n^{(0)}, \delta_j^{(1)} y_1^{(1)}, \dots, \delta_j^{(1)} y_p^{(1)}, \delta_j^{(2)} y_j^{(1)}, \dots, \delta_j^{(2)} y_q^{(1)} \rangle$$

은닉층 뉴런과 경쟁층 뉴런 사이의 대응 관계는 가중치에 기반한 변환 방법과 같다. 이 경사도에 기반한 방법에서는 자기 조직화 신경망은 경사도 정보에 기반한 위상 순서화된 뉴런들의 라벨을 출력한다.

#### 4. 인공 신경망 최적화를 위한 유전 알고리즘

##### 4.1 유전 알고리즘 구조

제안된 유전 알고리즘의 구조는 그림 4에 나와 있다. 이 구조는 자기 조직화 신경망을 이용한 변환(그림 5의 SOM-transformation)을 제외하면 전형적인 안정 상태 유전 알고리즘의 구조와 같다. 유전 알고리즘의 각 연산자에 대한 설명은 아래와 같다.

- **초기화:** 해집단에 존재하는 각 해들은 무작위로 설정된다.
- **선택 연산자:** 매 세대마다 각 해가 가지는 적합도에 비례하는 확률로 두 개의 부모해를 선택한다. 여기서 선택되는 확률은 해집단에서 가장 좋은 해가 가장 나쁜 해보다 4배 크도록 설정한다. 이 방법은 전형적인 정규화된 룰렛휠(roulette-wheel) 선택 방법이다.
- **교차 연산자:** 자식해는 이차원 지리적 교차 연산자(그림 5의 g2d\_xover)에 의해 생성된다. 지리적 교차 연산자는 4.3절에서 설명한다.
- **변이 연산자:** 아주 작은 확률로 자식해는 다음과 같은

변이 연산자에 의해 변경된다.  $w_i$ 를  $i$ 번째 유전자라고 할 때 새로운 변이 값은  $w_i = w_i + \lambda_{0.5}$ 으로 정해진다. 여기서  $\lambda_r$ 는  $-r$ 과  $r$ 사이의 값을 리턴 한다.

- **지역 최적화 알고리즘:** 자식해가 변이 연산자에 의해 변경된 후에 자식해는 역전파 알고리즘에 의해 지역 최적화된다. 역전파 알고리즘은 유전 알고리즘이 지역 최적점 근처에서 미세 조정을 할 수 있도록 돕는다. 다른 관점에서 보면, 유전 알고리즘은 역전파 알고리즘에 다양한 초기 해를 만들어주는 역할을 한다. 역전파 알고리즘이 적용된 후에 인공 신경망은 학습 데이터에 대한 평균 제곱 오차(mean square error) 값과 인식률을 갖게 된다. 자식해에 대한 적합도는 이 두 값을 이용하여 다음과 같은 식을 통해 결정된다. 적합도를  $f$ 라 할 때,  $f = \varepsilon + \eta(100 - \mu)$ 이다. 여기서  $\varepsilon$ 는 평균 제곱 오차이고  $\mu$ 는 인식률이다. 그리고  $\eta$ 는 비례 상수이다. 이것은 평균 제곱 오차로는 구별을 할 수 없는 자식해들 사이의 차이를 반영하기 위하여 인식률과 평균 제곱 오차를 결합한 함수로 적합도를 계산한 것이다.

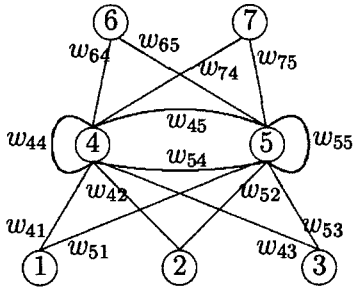
- **대치 연산자:** 해의 품질은 대치 방법에도 상당한 영향을 받는다. 단순한 (예를 들어, 항상 해집단 내의 가장 품질이 나쁜 멤버를 대치하는) 대치 방법을 사용할 경우, 유전 알고리즘은 해집단 내의 다양성을 빨리 잃지만 빨리 수렴할 수 있다. 반면, 엄격한 대치 계획을 사용할 경우, 유전 알고리즘은 오랫동안 높은 다양성을 유지할 수 있지만, 시간이 많이 걸린다. 시간 예산과 합치되는 대치 계획을 결정해야만 한다. Whitley와 Kauth는, Genitor라는 범용 유전 알고리즘의 일부하므로, 해집단 내의 가장 품질이 낮은 멤버를 새로운 자손으로 대치하는 대치 방법을 제안했다[20]. Cavicchio는 preselection이라 불리는 대치 방법을 제안했다[21]. 이 논문에서는 preselection 방법과 Genitor 방법을 결합하였다. 이 방법은 Genitor의 대치 방법에 비해 시간은 더 들지만 성능 면에 있어서는 훨씬 뛰어나다. 생성된 자식해는 다음과 같은 규칙에 의해 해집단의 한 해와 대치된다. 우선 생성된 자식해가 두 부모해 중 자식해와 유사한 부모해보다 적합도 값이 좋다면 이 유사한 부모해와 자식해를 대치한다. 그렇지 않은 경우, 자식해가 나머지 부모해보다 적합도가 좋다면 자식해와 이 부모해를 대치한다. 이 경우도 성립하지 않는다면 해집단의 가장 적합도가 낮은 해와 자식해를 대치한다. 이 정책의 기본 개념은 많은 시간을 소요하지 않으면서 해집단의 다양성을 유지하려는 것이다[22].

- **종료 조건:** 유전 알고리즘의 종료를 위해 고정된 수의 세대 수를 사용한다.

```

Create initial population of fixed size;
do {
    choose parent1 and parent2 from population;
    offspring = g2d_xover(parent1, parent2);
    mutation(offspring);
    backpropagation(offspring);
    SOM-transformation(offspring);
    if suited(offspring)
        then replace(population, offspring);
} until (stopping condition);
Return the best solution;
    
```

그림 5 제안된 유전 알고리즘의 구조



|   | input units |          |          | hidden units |          | output units |          |
|---|-------------|----------|----------|--------------|----------|--------------|----------|
|   | 1           | 2        | 3        | 4            | 5        | 6            | 7        |
| 4 | $w_{41}$    | $w_{42}$ | $w_{43}$ | $w_{44}$     | $w_{45}$ | $w_{64}$     | $w_{74}$ |
| 5 | $w_{51}$    | $w_{52}$ | $w_{53}$ | $w_{54}$     | $w_{55}$ | $w_{65}$     | $w_{75}$ |

그림 6 이차원 인코딩의 예제

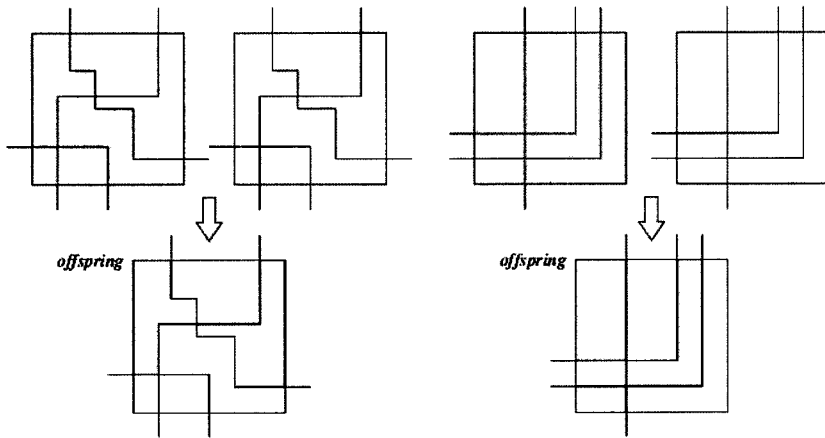


그림 7 이차원 지리적 교차 연산자의 예

4.2 문제 표현

인공 신경망 최적화를 위한 대부분의 직접 인코딩은 선형 문자열을 이용하여 신경망의 가중치 집합을 인코딩 한다[3,4,23]. 이 연구에서는 가중치의 집합을 선형 문자열로 변환하는 대신 인공 신경망을 가중치 행렬로 표현한다. 이 행렬에서, 각 행은 은닉층 뉴런에 대응하고 각 열은 입력층 뉴런, 은닉층 뉴런, 경쟁층 뉴런에 대응된다. 그림 6은 이런 인코딩의 예제를 보여준다. 이 그림에서 원으로 표시된 뉴런 안의 숫자는 뉴런의 인덱스를 표시하고  $w_{ij}$ 는 뉴런  $i$ 와 뉴런  $j$ 사이의 가중치를 표현한다. 신경망 상에 존재하는 간선들 사이의 연관성은 이차원 인코딩 상에서 잘 표현된다. 이것은 [11]의 연구 결과에 의해 뒷받침 된다. 여기서 좀더 생각해보면 특정 뉴런들은 다른 뉴런에 비해 강한 연결 관계성을 가질 수 있다고 생각해 볼 수 있다. 그래서 주어진 신경망을 동질의 신경망으로 변환함으로써 인코딩에 변화를 준다(3절을 참조하라). 요약하면 검색체는  $p \times (n + p + q)$  행렬에 의해 표현이 된다.

4.3 지리적 교차 연산자

이차원 인코딩은 유전자 사이에 존재하는 지리적 연

관성을 좀더 잘 보존한다[9-11]. 그러나 단순한 직선 기반 자름 선을 사용한다면 새로운 스키마를 생성하는 능력이 선형 인코딩 상의 교차 연산자에 의한 스키마 생성 능력보다 떨어지게 된다[10]. 지리적 교차 연산자는 이 문제를 해결하기 위해 제시되었다[10]. 지리적 교차 연산자는 이차원 인코딩에서 여러 개의 단조 증가 자름 선을 선택하여 염색체를 두 개의 동질 영역으로 나누고 두 부모 염색체로부터 각 영역의 유전자를 복사한다. 그림 7은 지리적 교차 연산자의 예를 보이고 있다. 이 연구에서는 지리적 교차 연산자를 사용한다. 이차원 인코딩과 동질 신경망으로의 변환, 이차원 지리적 교차 연산자를 결합하여 사용함으로써 인코딩 상에서 발생하는 정보 손실을 줄이려 하고 새로운 스키마 생성 능력을 증대시키려 한다.

5. 실험 결과

5.1 실험 데이터

제안된 방법의 성능을 측정하기 위하여 잘 알려진 세 가지 실험 데이터를 가지고 실험을 하였다. 우선, 필기체 숫자 인식을 위한 MNIST<sup>1)</sup> 데이터 베이스를 실험에 사용했다. 실험 데이터들은 10개의 숫자들로 구분되어

표 1 실험 결과

|       | BP            | GA            |               |               |               |
|-------|---------------|---------------|---------------|---------------|---------------|
|       |               | Original      | Heuristic     | W-SOM         | D-SOM         |
| MNIST | 90.28 (93.45) | 94.08 (95.20) | 96.52 (97.10) | 96.85 (99.60) | 97.14 (99.60) |
| WBCD  | 95.88 (96.37) | 96.87 (97.21) | 96.91 (97.43) | 97.19 (97.79) | 97.94 (98.00) |
| CHDD  | 80.56 (83.25) | 92.22 (84.26) | 82.75 (86.84) | 82.82 (87.02) | 84.37 (87.50) |

있다. 필기체 문자의 인식 문제는 문서 자동 처리에 있어서 아주 중요한 문제이다. 이런 종류의 응용 프로그램으로는 편지 봉투의 주소를 자동 읽기, 수표의 자동 처리 등을 들 수 있다. 이 실험에서는 학습 데이터로 800개의 숫자를, 검증 데이터(validation data)로 200개를, 테스트 데이터로 2000개를 사용하였다. 인공 신경망은 입력층에 45개, 은닉층에 20개, 출력층에 10개의 뉴런을 가진 구조를 사용하였다.

다음으로 UCI<sup>2)</sup> 데이터베이스에서 가져온 유방암 데이터(WBCD: Wisconsin breast cancer data)와 심장병 데이터(CHDD: Cleveland heart disease data)를 이용해 실험하였다. WBCD 데이터에 대해서 학습 데이터를 위해 80개, 검증 데이터를 위해 20개, 테스트 데이터를 599개로 나누어 사용하였다. 사용된 인공 신경망의 구조는 9-9-2의 신경망 구조를 하고 있다. 마지막으로 CHDD 데이터는 총 303개의 데이터를 포함하고 있다. 그 중 80개는 학습 데이터로 20개는 검증 데이터로, 나머지 203개의 데이터는 테스트 데이터로 사용하였다. 사용된 인공 신경망의 각 층에 존재하는 뉴런의 개수는 13-21-4이다.

이 실험에서는 제안된 방법의 정확한 실험 결과 측정을 위해 5-fold hold out method cross-validation 방법을 사용하였다. 이 실험 방법은 실험 데이터를 5개의 부분 데이터 집합으로 나눈다. 하나의 부분 데이터 집합은 검증에 이용되고 나머지 4개의 부분 데이터 집합을 이용해 학습을 한다. 부분 데이터 집합이 5개가 존재하므로 이 방법은 5가지의 다른 경우가 존재한다. 이 다른 5개의 부분 데이터 집합에 대해서 실험을 하고 평균을 사용하는 것이 5-fold hold out method cross-validation 방법이다.

## 5.2 실험 결과

이 절에서는 자기 조직화 신경망에 기반한 변환 방법이 얼마나 효율적인가를 위해 언급된 실험 데이터를 이용해 설명한다. 인공 신경망의 초기 가중치들은 -0.5와 0.5사이의 무작위 값을 갖는다. 학습 비율은 0.8을 사용한다. 또한 유전 알고리즘의 해집단의 크기는 50으로 설정하였다. 다른 유전 알고리즘 연산자는 4.1절에서 언급

하였다.

표 1은 각 실험 데이터에 대한 인식률을 보이고 있다. 표에서 BP는 역전과 알고리즘만 적용되어 학습시킨 인공 신경망을 나타낸다. 그리고 GA는 신경망 최적화에 유전 알고리즘이 적용된 실험을 나타내며 세부적으로 네 가지 실험 종류가 있다. Original은 유전 알고리즘 과정에 어떠한 변환 방법도 적용되지 않은 실험이고 나머지 실험들은 변환 방법이 포함되어 있다. Heuristic은 [11]에서 사용된 변환 방법을 나타내고 W-SOM은 이 논문에서 언급된 가중치에 기반한 변환이며 W-SOM은 경사도에 기반한 변환 방법이 포함된 유전 알고리즘이다. 각 실험에서 두 값 중 앞의 값은 실험 회수 500번의 평균을 나타내고 괄호 안의 값은 최대값을 나타낸다. 이 실험 결과로 알 수 있는 것은 유전 알고리즘이 병합된 방법이 그렇지 않은 경우 (BP) 보다 성능이 우수하다는 것이다. 특히 변환 방법이 포함된 경우가 더 나은 결과를 보이고 있다. 그 중 경사도에 기반한 변환 방법이 포함된 유전 알고리즘 방법이 가장 주목할만한 결과를 보이고 있다. 이것은 각 가중치들의 안정 상태를 고려하는 것이 신경망 학습에 도움이 된다는 것을 보이는 것이다.

## 6. 결론

이 논문에서는 뉴런 사이의 지리학적 상관성 또는 상대적 기여도를 효율적으로 계산하고 반영하기 위해 인공 신경망의 구조적 변환 방법을 고안하였다. 특히 자기 조직화 신경망의 위상 순서화 성질이 강한 연관성을 가지는 뉴런들을 군집화 시키는데 이용되었다. 제안된 자기 조직화 신경망에 기반한 변환은 유전 알고리즘의 진행 과정 동안 의미 있고 유용한 스킴이 생존 할 확률을 높여준다.

## 참고 문헌

- [1] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999.
- [2] A. F. James and M. S. David. *Neural Networks, Algorithms, Applications, and Programming Techniques*. Addison Wesley, 1994.
- [3] A. Grauel and F. Berk. Mapping of dynamical systems by recurrent neural networks in an

1) <http://yann.lecun.com/exdb/mnist>

2) <http://www.ics.uci.edu/~mllearn/MLRepository.html>



- evolutionary algorithm approach. In *European Congress on Intelligent Techniques and Soft Computing*, volume 1, pages 470-476, 1998.
- [4] R. Jeff and V. B. Ciesielski. An evolutionary approach to training feedforward and recurrent neural networks. In *International Conference on Knowledge-Based Intelligent Electronics Systems*, pages 596-602, 1998.
- [5] V. Patrakis, E. Paterakis, and A. Kehagias. A hybrid neural-genetic multimodel parameter estimation algorithm. *IEEE Transactions on Neural Networks*, 9(5):862-876, 1998.
- [6] J. D. Schaffer, D. Whitley, and L. J. Eshelman. Combinations of genetic algorithms and neural networks: A survey of the state of the art. In *International Workshop on Combinations of Genetic Algorithms and Neural Networks*, pages 1-37, 1992.
- [7] C. A. Anderson, K. F. Jones, and J. Ryan. A two-dimensional genetic algorithm for the Ising problem. *Complex Systems*, 5:327-333, 1991.
- [8] J. P. Cohoon and W. Paris. Genetic placement. In *IEEE International Conference on Computer-Aided Design*, pages 422-425, 1986.
- [9] T. N. Bui and B. R. Moon. On multi-dimensional encoding/crossover. In *International Conference of Genetic Algorithms*, pages 49-55, 1995.
- [10] A. B. Kahng and B. R. Moon. Toward more powerful recombinations. In *International Conference on Genetic Algorithms*, pages 96-103, 1995.
- [11] J. H. Kim and B. R. Moon. Neuron reordering for better neuro-genetic hybrids. In *Genetic and Evolutionary Computation Conference*, pages 407-414, 2002.
- [12] B. R. Moon and H. N. Kim. Effective genetic coding with a two-dimensional embedding heuristic. *International Journal of Knowledge-Based Intelligent Engineering Systems*, 3(2):113-120, 1999.
- [13] B. R. Moon, Y. S. Lee and C. K. Kim GEORG: VLSI circuit partitioner with a new genetic algorithm framework. *Journal of Intelligent Manufacturing*, 9(5):401-412, 1998.
- [14] J. L. Elman. Finding structure in time. *Cognitive Science*, 14:179-211, 1990.
- [15] T. Kohonen, S. Kaski, K. Lagus, J. Salojarvi, J. Honkela, V. Paatero, and A. Saarela. Self organization of a massive document collection. *IEEE Transactions on Neural Networks*, 11(3):574-585, 2000.
- [16] J. Bagley. *The Behavior of Adaptation Systems Which Employ Genetic and Correlation Algorithms*. PhD thesis, University of Michigan, Ann Arbor, MI, 1967.
- [17] T. N. Bui and B. R. Moon. Hyperplane synthesis for genetic algorithms. In *International Conference of Genetic Algorithms*, pages 102-109, 1993.
- [18] T. N. Bui and B. R. Moon. GRCA: A hybrid genetic algorithm for circuit ratio-cut partitioning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17(3):193-204, 1998.
- [19] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation, In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing*, volume 1, chapter 8. MIT Press, Cambridge, MA, 1986.
- [20] D. Whitley and J. Kauth, Genitor: A different genetic algorithm. In *Rocky Mountain Conference on Artificial Intelligence*, pages 118-130, 1988.
- [21] D. Cavicchio, Adaptive search using simulated evolution. PhD thesis, Univ. of Michigan, Ann Arbor, Mich., 1970, Unpublished.
- [22] T. N. Bui and B. R. Moon. Genetic algorithm and graph partitioning. *IEEE Transactions On Computers*, 45(7):841-855, 1996.
- [23] C. T. Lin and C. P. Jou. Controlling chaos by GA-based reinforcement learning neural network. *IEEE Transactions on Neural Networks*, 10(4): 846-869, 1999.



김정환

1998년 서울대학교 계산통계학과 전산과 학전공 학사. 2000년 서울대학교 전산과 학과 석사. 2006년 서울대학교 전기컴퓨터공학부 박사. 관심분야는 인공 신경망, 유전 알고리즘, 엘리베이터 군제어



문병로

1985년 서울대학교 계산통계학과 학사 1987년 KAIST 전산학과 석사. 1994년 펜실바니아 주립대 박사. 1987년~1991년 (주)LG전자 중앙연구소 선임연구원 1994년~1995년 UCLA VLSI CAD Lab 박사후연구원. 1996년~1997년 8월 (주)LG 반도체 DT연구소 책임연구원. 1997년 9월~현재 서울대학교 컴퓨터공학부 부교수. 관심분야는 최적화, 유전 알고리즘, 알고리즘 디자인 및 분석, 스케줄링 등