

# 적합도 공유에 의해 진화한 종분화 신경망

안준현<sup>0</sup> 조성배  
연세대학교 컴퓨터과학과  
{jhahn, sbcho}@candy.yonsei.ac.kr

## Speciated Neural Networks Evolved by Fitness Sharing

Joon-Hyun Ahn<sup>0</sup> and Sung-Bae Cho  
Department of Computer Science, Yonsei University

### 요약

기존 진화 신경망 연구는 마지막 세대에서 최적의 신경망을 찾는 연구가 대부분이었다. 하지만 이 방법은 마지막 세대의 다른 신경망들이 진화와 학습을 통해 얻은 정보를 모두 무시한다. 최근에는 가능한 많은 정보를 이용해서 보다 뛰어난 성능의 시스템을 구축하기 위해, 마지막 세대의 모든 신경망들의 정보를 결합하는 다중 신경망에 관한 연구가 진행되고 있다. 효과적인 다중 신경망을 구축하기 위해서는 다양한 신경망들이 다중 신경망을 구성해서 서로 보완해 주도록 하여야 하는데, 아직까지 효과적인 다중 신경망 구축 방법은 나오지 않고 있다. 본 연구는 유전자 연산에서 다양한 해를 찾기 위해 사용하는 종분화를 이용해서 다양한 신경망들이 생성되도록 하는 다중 신경망 구축방법을 제안하고 실험을 통해 이 방법의 효용성을 보인다.

### 1. 서론

진화 신경망은 진화를 통해 최적의 결과를 내는 신경망을 찾는 방법이다. 진화를 통해 신경망만을 사용했을 때보다 동적으로 환경에 적응해서 자동으로 좋은 성능을 내는 신경망 가중치, 위상구조, 은닉 노드 수 등을 찾아낸다[1][2][3]. 그러나 기존의 진화 신경망 연구는 신경망을 어떻게 효과적으로 진화시켜서 마지막 세대에 최적화된 신경망을 얻을 것인가였다. 이 방법은 최적의 결과를 보이는 하나의 신경망을 사용하기 때문에 진화와 학습 과정에서 다른 신경망들이 얻은 정보를 모두 무시해 버린다. 또한 최적의 신경망이라고 완벽하게 일반화된 해를 제공하는 경우는 거의 없기 때문에 다중 신경망을 이용해서 보다 좋은 성능과 해를 얻고자 하는 연구가 활발히 진행되고 있다 [4][5][6].

다중 신경망이 좋은 성능을 보이기 위해서는 각 신경망이 서로 다른 문제에 대해서 에러를 내면서 자신이 학습한 문제에 대해서는 좋은 성능을 보여야 한다[7]. 기존 연구에서는 신경망의 초기 가중치나 위상 구조를 다양하게 하거나, 각 신경망을 서로 다른 학습 데이터나 학습 알고리즘으로 학습을 시키는 등의 방법을 사용했다. 이런 방법들은 몇몇 문제에 대해서는 좋은 성능을 보이지만 근본적으로 우연히 신경망들이 다양하게 구성되기를 바라거나 문제에 대한 사전 지식을 많이 요구해서 일반적인 방법으로 보기는 어렵다[4].

본 연구에서는 이런 단점을 극복하면서 신경망들이 다양한 형태를 갖도록하기 위해 종분화 방법을 도입했다. 종분화는 진화 알고리즘에서 비슷한 해를 가지는 개체를 모아서 종을 이루도록해서 다양한 해를 찾도록 돕는 방법으로 이를 신경망의 진화에 적용해서 보다 다양한 신경망을 찾도록 한다. 종분화를 이용해 진화된 다중 신경망과 그렇지 않은 다중 신경망을 실험을 통해 비교해서 그 효용성을 알아본다.

### 2. 관련연구

### 2.1 진화 신경망

진화 신경망은 신경망의 가중치, 위상 구조, 은닉 노드 수, 학습 알고리즘 등 신경망 학습시 결정해야 하는 인자들을 여러 세대 진화를 통해 찾아서 최적의 신경망을 결정하는 방법이다[1][2][3]. 이 방법은 마지막 세대에서 최적의 신경망을 찾기 위해 어떻게 진화를 시킬 것인가를 연구하였다. 그렇지만 마지막 세대의 신경망들은 진화와 학습 과정을 거쳐 많은 정보를 가지고 있기 때문에 이를 함께 사용한다면 더 좋은 성능의 시스템을 구축할 수 있다. 최근에는 하나 이상의 신경망들을 이용해서 최적의 결과를 찾아내는 다중 신경망에 대한 연구가 활발히 진행되고 있다 [4][5][6].

### 2.2 다중 신경망의 결합

다양한 해결책을 결합해서 보다 일반화 능력이 뛰어나고 신뢰도 높은 시스템을 구축하려는 연구는 인간의 신체에서 뿐만 아니라 다양한 분야에서 시도되고 있다[4]. 신경망 분야에서 마찬가지로 하나 이상의 신경망을 결합해서 보다 좋은 성능의 시스템을 구축하려는 연구가 많이 시도되었다. 단순히 출력값의 평균을 사용하는 방법에서부터, 투표표를 하는 방법, Dempster-Shafer belief-based 방법, Supra Bayesian 방법, stacked generalisation 방법 등이 그러한 연구들이다[4]. 그렇지만 이런 연구들은 각 신경망들이 이미 구축된 상태에서 이들의 결과를 어떻게 결합할 것인가에 초점이 맞추어져 있기 때문에, 신경망을 구축할 때부터 서로의 오류를 보완해주는 신경망들이 생성되도록 하면 결합시에 더 좋은 성능을 얻을 수 있을 것이다. 본 연구에서는 신경망의 결합과 함께 종분화를 통해 다양한 신경망들을 찾아서 보다 효과적인 다중 신경망이 구축하였다.

### 3. 다중 신경망의 진화

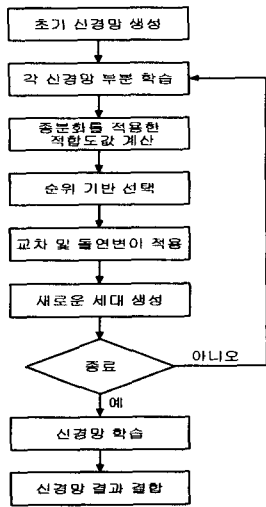


그림 1. 제안한 중분화 신경망 개요

중분화 방식에 의해 진화된 다중 신경망의 개요는 그림 1과 같다. 임의의 가중치 값으로 초기화된 신경망 집단을 생성하고 각 신경망을 부분 학습시켜서 신경망이 주어진 문제에 대해 최적 구조를 찾는 것을 돕는다. 신경망을 검증 데이터로 테스트해서 각 신경망의 적합도를 계산한다. 적합도는 중분화를 적용해서 계산한다. 적합도가 구해지면 가장 인식률이 높은 상위 50% 개체를 복사한다. 그리고 신경망 집단에 교차 연산과 돌연변이 연산을 적용해서 새로운 세대를 생성한다. 위 과정을 반복한 후 마지막 세대의 신경망 집단을 학습 데이터를 이용해서 충분히 학습 시켜서 다중 신경망을 구축한다. 다중 신경망의 결과는 신경망 집단의 결과를 투표, 평균, 최적 결합 방법 등을 이용해서 구한다.

3.1 신경망의 표현

신경망을 진화시키기 위해서는 이를 진화에 적합한 형태로 인코딩할 필요가 있다. 신경망 인코딩은 유전자 알고리즘에서 개체를 표현하는데 일반적으로 사용하는 이진 표현, 트리 형식, 링크드 리스트, 행렬 등의 방법이 있다[1]. 본 연구에서는 간단 하면서 유전 연산자를 적용하기 쉬운 행렬을 사용하였다. N개의 노드를 가진 신경망은 N×N 크기의 정방행렬에 연결 정보와 가중치를 표시해서 인코딩하였다. 행렬의 우상단은 노드간 연결 정보를 1과 0으로 표시하고, 각 연결 정보에 대칭하는 좌하단에는 가중치를 나타낸다. 그림 2는 전체 노드 수가 네 개인 신경망을 표현한 예를 보여준다.

3.2 교차 연산

교차 연산은 두 신경망의 연결 정보와 가중치를 서로 교환하도록 하였다. 우선 교차 연산을 적용할 두 신경망을 선택하고 교차점이 될 노드를 은닉 노드 중에서 하나 선택한다. 그리고 두 신경망에서 이 노드의 연결 정보와 가중치 정보를 서로 교환한다. 그림 3은 어떻게 표시된 노드를 교차점으로 한 교차 연산의 예를 보여준다.

3.3 돌연변이 연산

돌연변이 연산은 연결 삭제와 연결 생성 두 형태로 적용한다. 연결 삭제는 임의의 연결 정보를 선택해서 실제 연결이 있는 경우 이 연결을 삭제하고 반대로 연결 생성은 연결이 없을 경우 새로운 연결을 생성하고 임의의 값으로 연결 강도를 설정한다. 그림 4는 돌연변이 연산의 예를 보여준다.

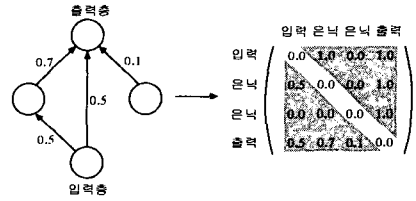


그림 2. 신경망 인코딩

4. 적합도 공유에 의해 중분화된 신경망

신경망 집단의 중분화를 위해 적합도 공유를 사용한다. 적합도 공유는 적합도가 집중된 개체의 적합도 증가치를 줄이고 그 주위의 개체들과 적합도를 공유해서 유전자 알고리즘의 검색 반경을 넓힌다. 결과적으로 검색하지 않았던 영역을 검색해서 다양한 해를 찾으려 한다[8].

각 개체의 적합도를  $f_i$ , 적합도 공유 정도를  $m_i$ 라고 하면, 공유 적합도는  $fs_i = f_i / m_i$ 로 구할 수 있다.  $m_i$ 는 각 개체간의 동질성 차이( $d_{ij}$ )를 이용해서 다음과 같이 계산된다.

$$m_i = \sum_{j=1}^{population\ size} sh(d_{ij})$$

공유 함수  $sh(d_{ij})$ 는 공유의 정도를 나타내는 것으로 서로 다른 개체는 작은 값을 가지고 비슷한 개체들 간에는 큰 값을 가지게 된다. 공유 함수는 다음과 같이 구한다.

$$sh(d_{ij}) = \begin{cases} 1 - \frac{d_{ij}}{\sigma_s} & \text{for } 0 \leq d_{ij} < \sigma_s \\ 0 & \text{for } d_{ij} \geq \sigma_s \end{cases}$$

$\sigma_s$ 는 공유 반경을 나타내는 값으로 적합도 공유를 할 만큼 비슷한 개체인지를 구분하는 기준이 된다.

본 연구에서는 신경망의 인식률을 적합도로 사용하고 동질성은 학습 데이터에 대한 각 신경망의 출력값 평균을 사용해서 비교하였다.

6. 다중 신경망의 결합

신경망 결합은 기존에 많은 연구가 진행되었다. 본 연구에서는 투표 방법, 평균 방법 그리고 최적화된 결합 방법을 사용한다.

투표 방법은 신경망 집합에서 가장 많은 신경망이 선택한 결과를 다중 신경망의 결과로 사용하는 것으로 별도의 계산 과정없이 신경망의 결과를 결합할 수 있다. 평균 방법 역시 별도의 계산 과정없이 결과를 구할 수 있는 방법으로 각 신경망의 출력값을 모아서 평균을 구해서 가장 출력값이 큰 값을 전체 신경망의 결과로 사용하는 방법이다. 최적화된 결합 방법은 전체 신경망 중에서 하나의 신경망이라도 올바른 결과를 보이면 전체 신경망의 결과도 올바른 결과를 내도록하는 이상적인 결합 방법으로 다른 결합 방법과의 비교를 위해 사용한다.

7. 실험결과

실험 데이터는 UCI의 벤치마크 데이터 중에서 Wisconsin 대학병원의 breast cancer 데이터를 사용한다. 이 데이터는 두 개의 클래스를 가지며, 각 데이터는 9가지 속성을 가지고 있다. 학습 데이터, 검증 데이터, 테스트 데이터는 각각 349개, 175개, 175개씩 사용하였다.

각 세대의 신경망 수는 20개이고 총 200세대를 진화시

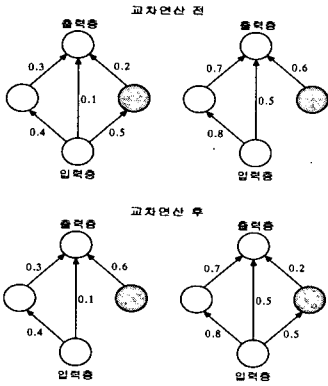


그림 3. 교차 연산

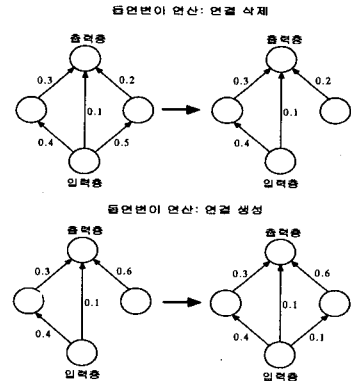


그림 4. 두연변이 연산

졌다. 각 신경망은 9 개의 은닉 노드를 가진 전방향 신경망이고 신경망의 학습은 역전파 알고리즘을 사용하였다. 학습률은 0.1, 학습 데이터는 부분 학습시에는 100 회, 최종 학습시에는 1000 회씩 사용하였다. 교차연산은 0.3 의 비율로, 두연변이 연산은 0.1의 비율로 적용하였다. 신경망의 결합은 마지막 세대의 개체에 대해서 투표 방법, 평균 방법, 최적 결합 방법을 적용하였다.

표1과 표2는 중분화 방식을 사용한 방법과 그렇지 않은 방법에 대해서 총 6 회씩 실험한 결과의 평균이다. 중분화를 사용한 그렇지 않은 경우보다 좋은 성능을 보인다. 결합 방법에서는 투표와 평균 방법이 최적 결합과는 상당한 차이를 보이고 있다.

그림 5는 중분화를 사용한 경우와 그렇지 않은 경우의 네 가지 결과를 비교한 그림이다. Y 축은 인식률을 나타내고 X 축은 마지막 세대에서 가장 좋은 결과를 보이는 신경망(1), 투표 방법(2), 평균 방법(3) 그리고 최적 결합(4)을 나타낸다. 1 과 나머지 경우를 비교해 보면 신경망을 결합했을 경우 더 좋은 결과를 얻었다. 또한 중분화를 사용했을 때 그렇지 않은 경우에 비해 더 나은 성능을 얻었다.

결합방법	인식률	오인식률	기각률
투표	94.19%	5.71%	0.1%
평균	94%	6%	0%
최적결합	99.53%	0.47%	0%

표 1. 중분화를 사용한 다중 신경망

결합방법	인식률	오인식률	기각률
투표	93.62%	6%	0.38%
평균	93.62%	6.38%	0%
최적결합	99.34%	0.66%	0%

표 2. 중분화를 사용하지 않은 다중 신경망

### 5. 결론

기존 다중 신경망 연구는 각 신경망이 서로 다른 문제에 대해서 오류를 내면서 정확한 결과를 낼 때 효과적인 다중 신경망이 구축됨을 보였다. 그러나 아직까지 효과적인 다중 신경망 구축 방법은 나오지 않고 있다. 본 연구에서는 다중 신경망 진화시에 중분화를 이용해서 다양한 신경망들이 생성되도록 해서 효과적인 다중 신경망이 구축되도록 하였다. 실험 결과 상으로는 본 연구에서 제안한 방법이 일반적인 경우보다 결합 방법에 상관없이 인식률에서 더 좋은 성능을 보였다. 향후 연구에서는 신경망들의 상관 관계를 분석하는 실험을 통해 다양한 신경망이 생성되었는지 확인하는 연구가 수행될 것이다.

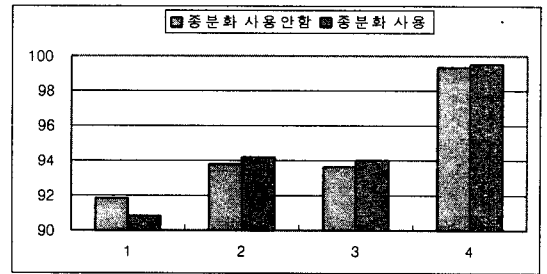


그림 5. 실험 결과 비교

### 8. 참고문헌

- [1] X. Yao, "Evolving Artificial Neural Networks", *Proceedings of the IEEE*, 87(9):1423-1447, September 1999.
- [2] P.A. Castillo, V. Rivas, J.J. Merelo, J. Gonzalez, A. Prieto, and G. Romero, "G-Prop-II: Global Optimization of Multilayer Perceptrons using GAs", *Proceedings of the 1999 Congress on Evolutionary Computation*, vol. 3, pp. 2022-2027, May 1997.
- [3] X. Yao and Y. Liu, "A New Evolutionary System for Evolving Artificial Neural Networks", *IEEE Trans. Neural Networks*, vol 8, pp.694-713, Anchorage, USA, 4-9 May 1998.
- [4] Amanda J. C. Sharkey, "On Combining Artificial Neural Nets", *Connection Science*, Vol. 8, pp.299-313, 1996.
- [5] X. Yao and Y. Liu, "Making Use of Population Information in Evolutionary Artificial Neural Networks", *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, 28 (3):417-425, June 1998.
- [6] David W. Opitz and Jude W. Shavlik, "Actively Searching for an Effective Neural Network Ensemble", *Connection Science*, Vol 8, Nos 3 & 4, pp. 337-353, 1996.
- [7] Y. Liu and X. Yao, "Ensemble Learning via Negative Correlation", *Neural Networks*, Vol. 12, No. 10, December 1999.
- [8] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading Massachusetts, 1989.