

---

# 일반화된 캐스케이드 코릴레이션 알고리즘과 일반화된 순환 캐스케이드 코릴레이션 알고리즘의 결합을 통한 학습 능력 향상

## Improvement of Learning Capability with Combination of the Generalized Cascade Correlation and Generalized Recurrent Cascade Correlation Algorithms

---

이상화, 송해상  
서원대학교 컴퓨터정보통신공학부

Sang-Wha Lee(swl@seowon.ac.kr), Hae-Sang Song(hssong@seowon.ac.kr)

---

### 요약

본 논문에서는 일반화된 캐스케이드 코릴레이션 학습 알고리즘과 일반화된 순환 캐스케이드 코릴레이션 학습 알고리즘의 결합을 통한 새로운 알고리즘을 소개한다. 이 새로운 알고리즘은 패턴분류문제(pattern classification problem)의 신속한 해결을 위하여 비순환 뉴런이 유리한지 순환 뉴런이 유리한지 또는 수직성장이 유리한지 수평성장이 유리한지 고민할 필요 없이 후보뉴런의 학습 중에 네트워크의 구성을 스스로 결정한다. 이 알고리즘의 성능평가를 위하여 학습 알고리즘에서 중요한 기준 문제(benchmark problem) 중의 하나인 콘택트렌즈 문제(Contact lens problem)와 밸런스 스케일 문제(Balance scale problem)에 대하여 실험하였고 기존의 캐스케이드 코릴레이션 알고리즘 및 순환 캐스케이드 코릴레이션 알고리즘과 성능을 비교 하였다. 이 실험에서 활성화 함수는 일반적으로 많이 사용하는 시그모이드 함수(sigmoidal function)와 하이퍼볼릭탄젠트 함수(hyperbolic tangent function)를 사용하였다. 이 새로운 알고리즘은 학습을 통하여 기존의 알고리즘보다 적은 수의 은닉뉴런을 생성하여 보다 빠른 학습 속도를 보여주었다.

■ 중심어 : | (순환)캐스케이드코릴레이션 알고리즘 | 활성화함수 | 시그모이드함수 | 하이퍼볼릭탄젠트함수 |

### Abstract

This paper presents a combination of the generalized Cascade Correlation and generalized Recurrent Cascade Correlation learning algorithms. The new network will be able to grow with vertical or horizontal direction and with recurrent or without recurrent units for the quick solution of the pattern classification problem. The proposed algorithm was tested learning capability with the sigmoidal activation function and hyperbolic tangent activation function on the contact lens and balance scale standard benchmark problems. And results are compared with those obtained with Cascade Correlation and Recurrent Cascade Correlation algorithms. By the learning the new network was composed with the minimal number of the created hidden units and shows quick learning speed. Consequently it will be able to improve a learning capability.

■ keyword : | (Recurrent) Cascade Correlation Algorithm | Activation Function | Sigmoid Function | Hyperbolic Tangent Function |

### 1. 서론

캐스케이드 코릴레이션(Cascade Correlation(CC)) 네트워크는 학습 중에 스스로 네트워크를 형성하므로 문제해결을 위하여 전문가가 미리 네트워크를 설계할 필요가 없는 특징을 갖고 있다[2]. 초기 CC 네트워크는 은닉뉴런과 은닉층 없이 다만 입력뉴런과 출력뉴런의 완전한 연결에 의하여 구성된다. CC 학습알고리즘에서 은닉뉴런은 네트워크에 한 번에 한 개씩 추가되고 선택된 가중치의 값은 변화하지 않는다. 여기에서 추가할 뉴런의 출력과 네트워크의 잔여오차의 상호관계값(correlation value)의 극대화를 시도한다. 새로운 한 은닉뉴런의 생성을 위하여, 후보뉴런(candidate unit)이 학습할 수 있는 시그널은 네트워크의 모든 입력과 이미 존재하고 있는 모든 은닉뉴런들과의 연결들을 통하여 전달된다. 여기에서 이 후보뉴런의 출력은 아직 네트워크와는 연결되지 않은 상태이다. 네트워크는 주어진 패턴들에 대하여 한 번의 학습 후에 후보뉴런들의 입력가중치(input weight)를 수정한다.  $S_i$ 의 극대화를 위한 수정은 다음과 같이 정의 한다.  $S_i$ 는 한 후보뉴런  $i$ 에서 생성된 값  $c$ 와 모든 출력뉴런  $o$ 에서 측정된 잔여출력오차  $E_o$ 의 상호관계(correlation)의 합계로 나타낸다.

$$S_i = \sum_o \left| \sum_p (c_{pi} - \bar{c}_i)(E_{po} - \bar{E}_o) \right| \quad (1)$$

식 (1)에서  $E_{po}$ 는 패턴  $p$ 에 대한 출력뉴런  $o$ 에서의 오차이고  $c_{pi}$ 는 패턴  $p$ 에 대한 후보뉴런  $i$ 의 출력이다.  $\bar{c}_i$ 는 모든 패턴들에 대한 후보뉴런  $i$ 의 평균출력이고  $\bar{E}_o$ 는 모든 패턴들에 대한 출력뉴런  $o$ 에서의 평균오차이다.

후보뉴런  $i$ 의 입력을 위한 연결에서의 가중치  $w_{ij}$ 에 대한  $S_i$ 의 편미분의 유도식은 다음과 같다:

$$\begin{aligned} \frac{\partial S_i}{\partial w_{ij}} &= \sum_o \frac{\partial}{\partial w_{ij}} \left| \sum_p (c_{pi} - \bar{c}_i)(E_{po} - \bar{E}_o) \right| \\ &= \sum_o \left( \text{sign} \left[ \sum_p (c_{pi} - \bar{c}_i)(E_{po} - \bar{E}_o) \right] \sum_p \frac{\partial}{\partial w_{ij}} [(c_{pi} - \bar{c}_i)(E_{po} - \bar{E}_o)] \right) \end{aligned}$$

$$= \sum_o \left( \text{sign} \left( \sum_p (c_{pi} - \bar{c}_i)(E_{po} - \bar{E}_o) \right) \sum_p \frac{\partial c_{pi}}{\partial w_{ij}} (E_{po} - \bar{E}_o) \right) \quad (2)$$

$\sigma_o$ 를 다음과 같이 정의하면

$$\sigma_o = \text{sign} \left( \sum_p (c_{pi} - \bar{c}_i)(E_{po} - \bar{E}_o) \right)$$

식 (2)는 다음과 같이 쓸 수 있다.

$$\frac{\partial S_i}{\partial w_{ij}} = \sum_o \sum_p \sigma_o \frac{\partial}{\partial net_{pi}} f_p(net_{pi}) \frac{\partial net_{pi}}{\partial w_{ij}} (E_{po} - \bar{E}_o)$$

따라서  $S_i$ 의 편미분은 다음과 같다:

$$\frac{\partial S_i}{\partial w_{ij}} = \sum_o \sum_p \sigma_o f'_p I_{pi}(E_{po} - \bar{E}_o) \quad (3)$$

식 (3)에서  $\sigma_o$ 는 후보뉴런  $i$ 에서 패턴  $p$ 에 대한 출력  $c_{pi}$ 와 출력뉴런  $o$ 에서 패턴  $p$ 에 대한 오차  $E_{po}$ 와의 상호관계 값의 부호를 나타낸다.  $f'_p$ 는  $net_{pi}$ 에 대한 후보뉴런  $i$ 의 활성화함수의 미분이고  $I_{pi}$ 는 패턴  $p$ 에 대한 후보뉴런  $i$ 의 입력을 의미한다.  $S_i$ 의 극대화를 위한 후보뉴런의 학습과 오차수정을 위한 출력뉴런의 학습은 quickpropagation 알고리즘을 이용한다[2].

순환 캐스케이드 코릴레이션(Recurrent Cascade Correlation(RCC)) 네트워크에서의 학습은 CC 네트워크에서의 학습과 동일하고, 다만 후보뉴런이 생성될 때마다 [그림 1]과 같이 부가적으로 순환 연결이 이루어져서 출력 값이 다시 뉴런의 입력 가중치로 사용된다 [3][6].

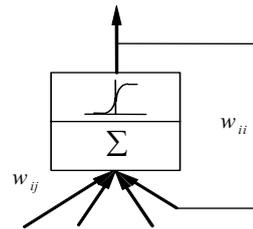


그림 1. 순환 연결 가중치를 갖고 있는 후보뉴런

따라서 후보뉴런  $i$ 의 출력은 다음과 같이 계산되어진다.

$$o_i^{(k)}(t) = \text{act} \left( \sum_j o_j(t) w_{ij}^{(k)} + o_i^{(k)}(t-1) w_{ii}^{(k)} \right)$$

여기에서 가중치  $w_{ij}^{(k)}$  와  $w_{ii}^{(k)}$  에 따른 출력의 미분은 다음과 같다.

$$\frac{\partial}{\partial w_{ii}^{(k)}} o_i^{(k)}(t) = f'_{\text{act}}(\text{net}_i) \left( o_i^{(k)}(t-1) + \frac{\partial}{\partial w_{ii}^{(k)}} o_i^{(k)}(t-1) w_{ii}^{(k)} \right) \quad (i \neq j)$$

$$\frac{\partial}{\partial w_{ij}^{(k)}} o_i^{(k)}(t) = f'_{\text{act}}(\text{net}_i) \left( o_j(t) + \frac{\partial}{\partial w_{ij}^{(k)}} o_i^{(k)}(t-1) w_{ii}^{(k)} \right)$$

$\frac{\partial o_i^{(k)}(t-1)}{\partial w_{ij}^{(k)}}$  의 값은 이전의 학습에서 생성된 것이 현재의 학습에서 다시 사용된다. 따라서 RCC 네트워크에서는 뉴런의 활성화 값  $o_i^{(k)}(t-1)$  과 가중치 값들의 저장이 부가적으로 필요하다.

CC와 RCC는 새로운 뉴런이 각 층에 한 뉴런만 첨가되므로 학습과정 동안 매우 깊은 네트워크를 형성하게 되어 은닉뉴런의 높은 fan-in의 요인이 된다. 따라서 네트워크의 학습에서 풀(pool)에 수평 성장 할 수 있는 후보뉴런을 추가하고 병렬 학습을 시행하여 좀 더 빠른 학습효과를 가져올 수 있는 은닉뉴런이 선택될 수 있는 기회를 제공한다. 이렇게 하여 구성된 일반화된 CC는 참고문헌 [9]에 그리고 일반화된 RCC는 참고문헌 [10]에 더욱 자세히 기술되어 있다. 본 논문에서는 이 두 알고리즘을 결합하여 네트워크가 자동으로 수평 또는 수직 그리고 순환 또는 무순환으로 주어진 조건에서 모든 가능한 방향으로 성장할 수 있도록 구조를 개선하였다. 이렇게 하면 벤치마크 문제의 특성과 여러 모델을 통한 다수의 실험 결과를 비교 할 필요 없이 직접 G-CC-RCC 알고리즘을 이용하면 되는 편리한 이점 있다. 이어서 II장에서 알고리즘의 일반화에 관련한 내용을 간략하게 설명하고, III장에서 이 두 알고리즘을 결합한 새로운 알고리즘의 구성을 제안하며, IV장에서 시뮬레이션을 통하여 새로운 알고리즘의 성능을 평가한 후에 V장에서 결론을 맺는다.

## II. 알고리즘의 일반화

### 1. 캐스케이드 코릴레이션

캐스케이드 코릴레이션 네트워크의 일반화를 위하여 풀(pool)에 있는 후보뉴런들을 두 그룹으로 나눈다[9]. 첫 번째 그룹에 있는 후보뉴런들은 모든 입력뉴런과, 바이어스와 이미 생성된 모든 은닉뉴런과 연결한다. 두 번째 그룹에 있는 후보뉴런들은 모든 입력뉴런과, 바이어스와 그리고 같은 층에 있지 않은 모든 은닉뉴런과 연결한다. 기회균등의 원칙에 의하여 두 그룹은 같은 수의 후보뉴런들로 구성되어야 한다. [그림 2]는 그러한 예로서 각 그룹에 네 개의 후보뉴런들이 나열된 상태를 보여준다. 여기에서  $v$ 로 표시한 후보뉴런은 수직 성장을 위하여 그리고  $h$ 로 표시한 후보뉴런은 수평 성장을 위하여 학습한다.



그림 2. 한 풀에 정렬되어 있는 후보뉴런들

같은 그룹에 있는 모든 은닉뉴런들은 같은 입력시그널을 받으며 각 패턴들로 부터 원인이 된 같은 잔여오차를 갖고 있다. 그들은 서로 아무런 영향을 주고받지 않고 학습하는 동안 네트워크에도 영향을 주지 않기 때문에 풀에서 병렬로 학습한다. 더 이상의 학습이 진행되지 않을 때에는, 즉 뉴런의 학습 횟수가 주어진 파라미터값에 도달 했을 때 가장 큰 상호관계값(correlation score)을 갖고 있는 한 후보뉴런이 은닉뉴런으로 선택되어 네트워크에 첨가되고 다른 뉴런들과 연결된다. 그때마다 네트워크의 성장이 결정된다. 만약 후보뉴런이 첫 번째 그룹에서 선택되었다면 네트워크는 수직 성장을 하고 두 번째 그룹에서 선택되었다면 수평 성장을 한다.

[그림 3]은 수직성장 또는 수평성장의 학습을 위한 후보뉴런  $k$ 의 입력을 보여주기 위한 예이다. 여기에서  $j, i, g$ 는 이미 생성된 은닉뉴런을 나타낸다. 수직 성장을 위해 뉴런  $k$ 는 뉴런  $j, i$  그리고  $g$ 로 부터 입력시그널을 받고 수평 성장을 위해서는 뉴런  $j$ 와  $i$ 로부터 입력시그

널을 받지만 뉴런  $g$ 로 부터는 아무런 영향도 받지 않는다.

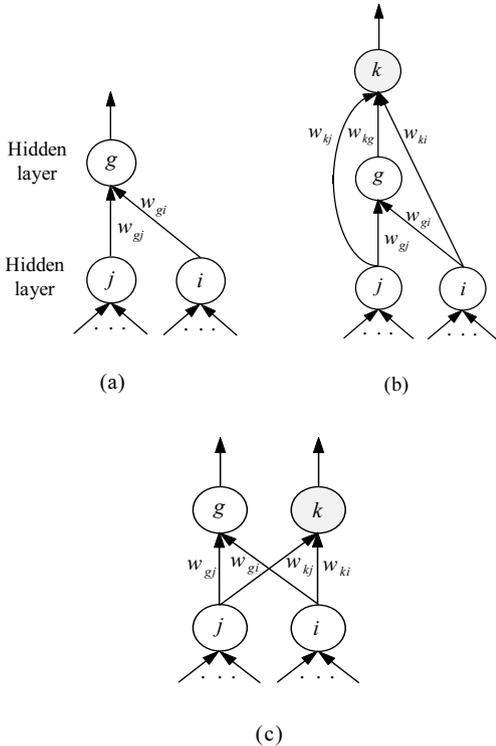


그림 3: (a): 이전의 학습단계에서 이미 생성된 세 은닉뉴런  $j, i, g$ . 현재 학습단계에서 (b): 네트워크의 수직성장을 위한 후보뉴런  $k$ 의 입력 (c): 네트워크의 수평성장을 위한 후보뉴런  $k$ 의 입력

수직성장과 수평성장을 위한 뉴런  $k$ 의 출력은 다음과 같다.

- 수직성장을 위한 후보뉴런  $k$ 의 출력:

$$c_k(t) = f_{act} \left( \sum_{l=j}^g c_l(t) w_{kl} \right)$$

- 수평성장을 위한 후보뉴런  $k$ 의 출력:

$$c_k(t) = f_{act} \left( \sum_{l=j}^i c_l(t) w_{kl} \right)$$

수직 또는 수평 성장할 수 있는 CC를 일반화된 CC(Generalized CC)라 하고 G-CC라고 표기하기로 한다[9].

## 2. 순환 캐스케이드 코릴레이션

순환 캐스케이드 코릴레이션 네트워크의 일반화를 위하여 풀(pool)에 있는 후보뉴런들을 두 그룹으로 나눈다[10]. 첫 번째 그룹에는 후보뉴런들이 모든 입력뉴런과, 바이어스와 그리고 이미 생성된 모든 은닉뉴런과 연결한다. 두 번째 그룹에는 후보뉴런들이 모든 입력뉴런과, 바이어스와 그리고 같은 층에 있지 않은 모든 은닉뉴런과 연결한다. 여기에서 두 그룹은 같은 수의 후보뉴런들로 구성된다. [그림 4]는 한 예로서 각 그룹에 네 개의 후보뉴런들이 나열된 상태를 보여준다. 여기에서도  $v$ 로 표시한 후보뉴런은 수직 성장을, 그리고  $h$ 로 표시한 후보뉴런은 수평성장을 위하여 학습한다.

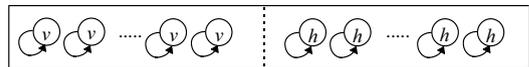
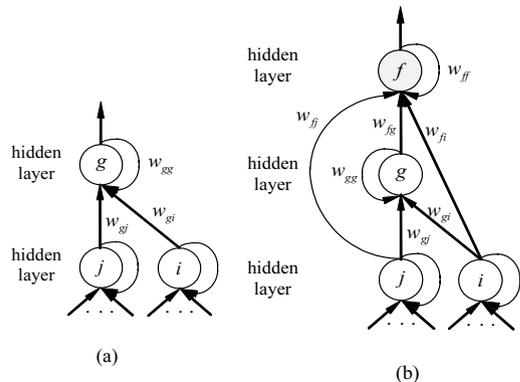


그림 4. 한 풀에 정렬되어 있는 후보뉴런들

순환 캐스케이드 코릴레이션 알고리즘의 일반화에서도 G-CC 알고리즘에서와 같은 방법으로 학습하여 가장 큰 상호관계 값을 갖고 있는 한 후보뉴런이 은닉뉴런으로 선택되어 네트워크의 수직성장과 수평성장이 결정된다. [그림 5]는 수직 또는 수평성장의 학습을 위한 후보뉴런  $k$ 의 입력을 보여주는 예이다. 여기에서  $j, i, g$ 는 이미 생성된 은닉뉴런을 나타낸다. 수직성장을 위해 뉴런  $k$ 는 뉴런  $j, i$  그리고  $g$ 로 부터 입력시그널을 받고 수평성장을 위해서는 뉴런  $j$ 와  $i$ 로부터 입력시그널을 받지만 뉴런  $g$ 로 부터는 아무런 영향도 받지 않는다.



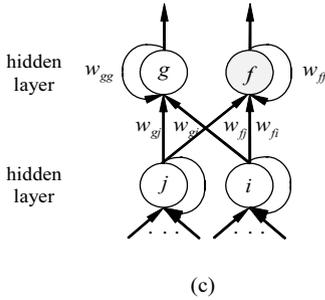


그림 5. (a): 이전의 학습단계에서 이미 생성된 세 은닉뉴런  $j, i, g$ . 현재 학습단계에서 (b): 네트워크의 수직성장을 위한 후보뉴런  $k$ 의 입력 (c): 네트워크의 수평성장을 위한 후보뉴런  $k$ 의 입력

수직성장과 수평성장을 위한 뉴런  $k$ 의 출력은 다음과 같다:

수직성장을 위한 후보뉴런  $k$ 의 출력:

$$o_f^{(k)}(t) = f_{act} \left( \sum_{l=j}^g o_l^{(k)}(t)w_{fl}^{(k)} + o_f^{(k)}(t-1)w_{ff}^{(k)} \right)$$

수평성장을 위한 후보뉴런  $k$ 의 출력:

$$o_f^{(k)}(t) = f_{act} \left( \sum_{l=j}^i o_l^{(k)}(t)w_{fl}^{(k)} + o_f^{(k)}(t-1)w_{ff}^{(k)} \right)$$

수직 또는 수평 성장할 수 있는 RCC를 일반화된 RCC(Generalized RCC)라 하고 G-RCC라고 표기하기로 한다[10].

### III. G-CC와 G-RCC 알고리즘의 결합

본 논문에서 제안하는 G-CC와 G-RCC를 결합하기 위하여 후보뉴런의 풀을 다음과 같이 구성한다.

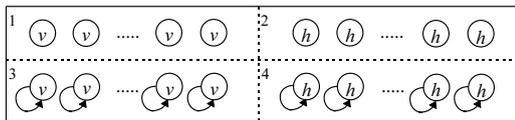


그림 6. 한 풀에 정렬되어 있는 후보뉴런들

이 네트워크의 성장과 학습도 G-CC와 G-RCC에서와 같은 방법과 학습알고리즘으로 진행이 되고 후보뉴

런의 풀에서 수직 또는 수평, 순환 없이 또는 순환 있게 예약된 후보뉴런 중에서 선택되어 네트워크의 성장이 결정 된다. 이러한 네트워크를 G-CC-RCC(Generalized CC and RCC)라고 표기하기로 한다. [그림 7]은 네트워크 구성을 위한 초기 상태를 나타내고 [그림 8]은 네트워크의 첫 번째 성장의 두 가지 상태를 보여준다.

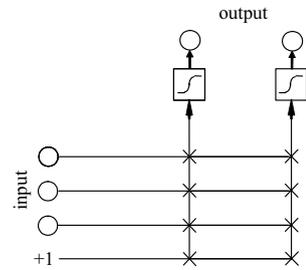
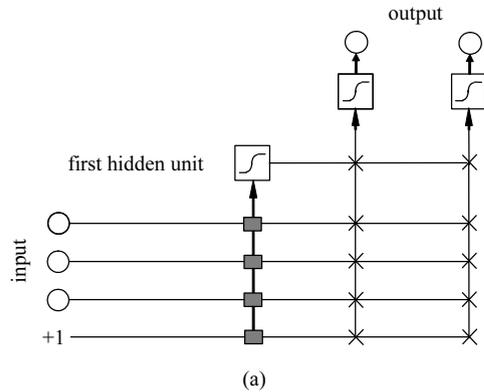
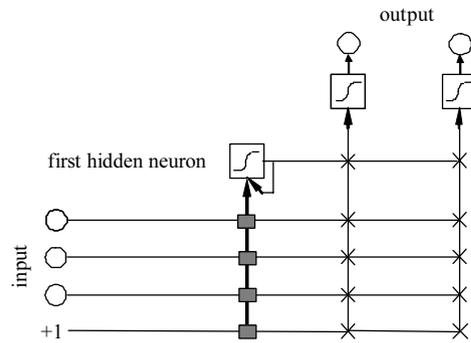


그림 7. G-CC-RCC의 초기 상태



(a)



(b)

그림 8. G-CC-RCC의 첫 번째 성장, (a): 무순환, (b): 유순환

[그림 9]는 첫 번째 성장 이후에 선택된 은닉뉴런을 통하여 가능한 (a)~(d)의 네 가지 형태의 네트워크 구성을 보여준다.

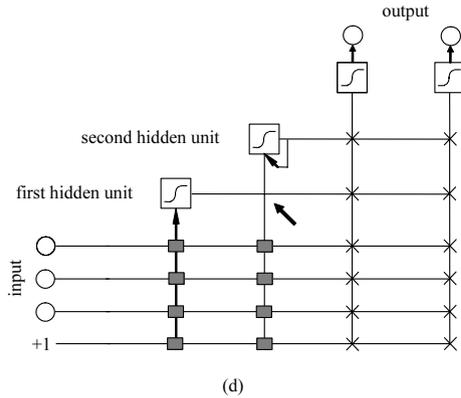
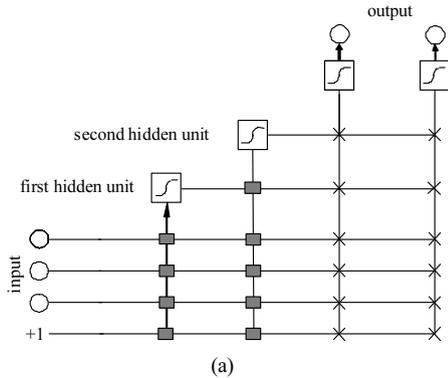
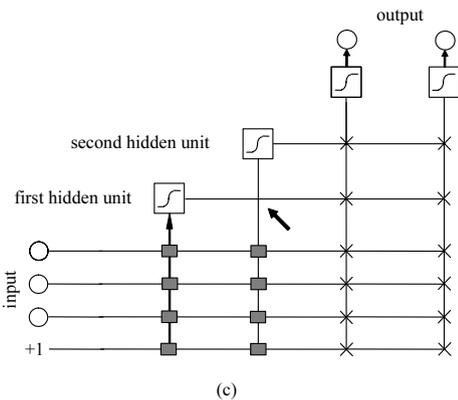
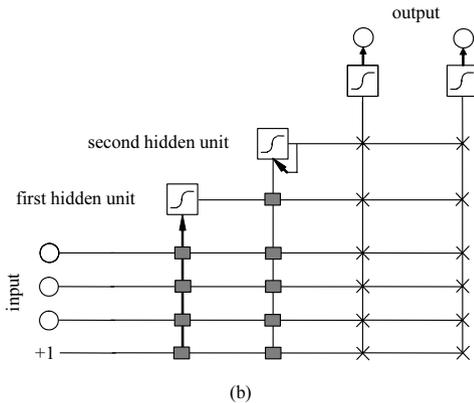


그림 9. G-CC-RCC의 두 번째 성장, (a): 무순환과 수직 성장, (b): 유순환과 수직성장, (c): 무순환과 수평 성장, (d): 유순환과 수평성장



#### IV. 실험 및 성능평가

본 논문에서 제안하는 G-CC-RCC 알고리즘과 CC 및 RCC 알고리즘과 실험을 통하여 비교평가 하기 위하여 벤치마크문제(benchmark-problem)인 콘택트렌즈 문제를 선택했다. 이 문제는 한 환자에게 어떠한 종류의 콘택트렌즈가 알맞은지 구분하는 것이다. 여기에는 9개의 실험 규칙이 있고 실험 정보는 다음과 같다.

- 경우의 수: 24

- 속성의 수: 4

- 속성에 관한 정보:

(1) 환자의 나이: ① 어린이, ② 노안 전 단계, ③ 노안

(2) 안경 착용: ① 근시, ② 원시

(3) 난시: ① 있음, ② 없음

(4) 눈물 생산 비율: ① 줄어 듦 ② 보통

- 3개의 클래스

(1) 환자에게 하드 콘택트렌즈가 적당 함

(2) 환자에게 소프트 콘택트렌즈가 적당 함

(3) 환자에게 콘택트렌즈가 적당하지 않음

- 클래스 분포

(1) 하드 콘택트렌즈: 4

(2) 소프트 콘택트렌즈: 5

(3) 콘택트렌즈가 적당하지 않음: 15

컨택트렌즈 문제의 실험데이터는 참고문헌 [7]에서 얻을 수 있다. 실험을 위한 기본 시뮬레이터(Simulator)는 Stuttgart Neural Network Simulator (SNNS)를 이용하였다[8]. 실험 결과에서 은닉뉴런과 후보뉴런을 위한 활성화함수는 '\*'와 함께 그리고 네트워크의 출력뉴런을 위한 활성화함수는 '.'와 함께 표시했다. 그림에서 원안의 숫자는 은닉뉴런들의 생성된 순서를 나타낸다. 실험에 사용한 활성화함수는 일반적으로 많이 사용하는 시그모이드(sigmoid) 함수와 하이퍼볼릭 탄젠트(hyperbolic tangent) 함수이다.

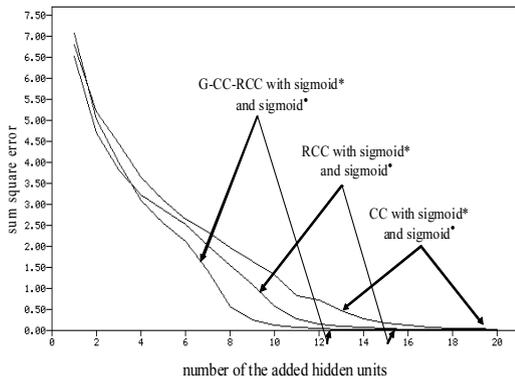


그림 10. 학습오차와 생성된 은닉뉴런 수의 비교

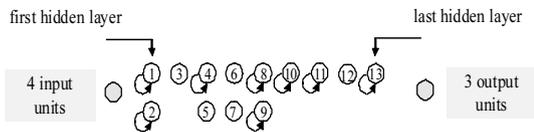


그림 11. 학습을 완료한 후 각 은닉층에 뉴런들이 나열된 상태

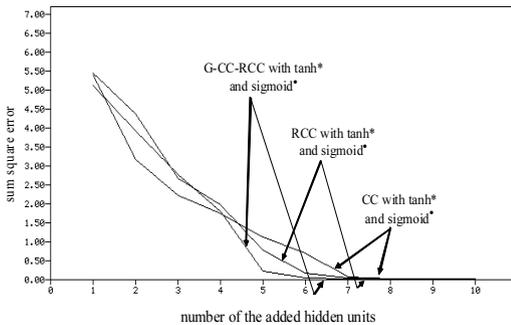


그림 12. 학습오차와 생성된 은닉뉴런 수의 비교

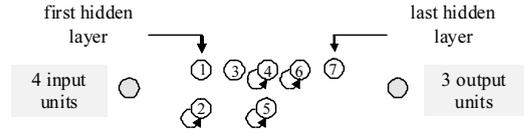


그림 13. 학습을 완료한 후 각 은닉층에 뉴런들이 나열된 상태

실험 결과에 의하면 G-CC-RCC는 시그모이드 함수에서 CC에 비해서 7개, RCC에 비해서 3개의 적은 수의 은닉뉴런이 생성 되었다. 하이퍼볼릭 탄젠트 활성화 함수를 사용한 실험에서는 CC와 RCC에 비해서 1개씩 적은 수의 뉴런이 생성되었다. 특히 후보뉴런에 하이퍼볼릭 탄젠트 함수를 사용한 경우에는 시그모이드 함수에 비해서 약 50% 적은 수의 뉴런이 생성되었다.

표 1. 콘택트렌즈 문제 해결을 위하여 생성된 뉴런의 수, 생성된 은닉층의 수, 순환된 뉴런수의 비교

네트워크	활성화 함수	은닉층의 수	은닉 뉴런수	순환된 뉴런수
CC	sigmoid*, sigmoid*	20	20	0
RCC	sigmoid*, sigmoid*	16	16	16
G-CC-RCC	sigmoid*, sigmoid*	9	13	8
CC	tanh*, sigmoid*	8	8	0
RCC	tanh*, sigmoid*	8	8	8
G-CC-RCC	tanh*, sigmoid*	5	7	4

본 논문에서 두 번째 선택한 벤치마크 문제는 밸런스 스케일 문제(Balance Scale Problem)이다. 실험 데이터는 심리학 실험에서 생성되었고, 막대기의 중앙을 기점으로 거리 값과 무게 값을 곱해서 막대기가 오른쪽 또는 왼쪽으로 기울었는지, 아니면 균형을 이루었는지를 결과로 분류한다. 왼쪽과 오른쪽에 대한 무게 값과 거리 값은 1~5의 값을 가진다. 실험에 사용한 625개의 데이터 중에서 49개는 균형을 이룬 경우이고 288개는 왼쪽으로 기울은 경우이며 288개는 오른쪽으로 기울은 경우이다. 이에 대한 실험 결과는 [그림 14]에서 보여주고 [그림 15]는 학습 후에 만들어진 네트워크의 은닉뉴런 구성을 나타낸다. 또한 [표 2]는 학습 후에 생성된 은닉층의 수, 은닉 뉴런의 수 및 순환된 은닉뉴런의 수를 나타낸다.

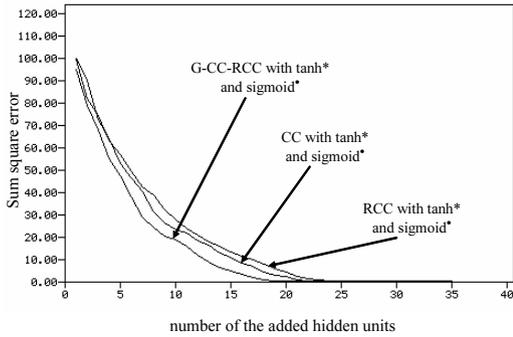


그림 14. 학습오차와 생성된 은닉뉴런 수의 비교

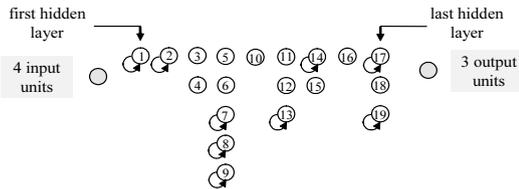


그림 15. 학습을 완료한 후 각 은닉층에 뉴런들이 나열된 상태

표 2. 밸런스 스케일 문제 해결을 위하여 생성된 뉴런의 수, 생성된 은닉층의 수, 순환된 뉴런수의 비교

네트워크	활성화 함수	은닉층의 수	은닉 뉴런수	순환된 뉴런수
CC	tanh*, sigmoid*	22	22	0
RCC	tanh*, sigmoid*	24	24	24
G-CC-RCC	tanh*, sigmoid*	9	19	9

실험결과에서 G-CC-RCC는 CC에 비해서 3개 RCC에 배해서 5개의 적은 은닉뉴런 수를 생성하였다. 이 실험에서도 첫 번째 실험과 비슷한 성능 향상을 보여준다.

### V. 결론 및 향후 과제

일반화된 캐스케이드 코릴레이션과 일반화된 순환 캐스케이드 코릴레이션 알고리즘의 결합을 통하여 네트워크가 자동으로 가능한 모든 방법으로 성장할 수 있도록 구성하였다. 본 논문에서 제안한 알고리즘의 성능 평가를 위하여 콘택트렌즈 문제와 밸런스 스케일 벤치마크문제를 채택하였다. 실험 결과에서 G-CC-RCC는 CC와 RCC에 비해서 적은 수의 은닉뉴런을 생성하여

학습 능력이 향상되었음을 확인 하였다. 향후에는 더욱 더 학습능력을 최적화 시킬 수 있는 좋은 후보뉴런의 구성과 선택을 위하여 유전자 알고리즘(genetic algorithm) 등을 이용한 최적화 시키기 위한 방법의 연구가 필요하다.

### 참고 문헌

- [1] F. Dandurand, V. Berthiaume, and T. R. Shultz, "A systematic comparison of flat and standard cascade-correlation using a student-teacher network approximation task," *Connection Science*, Vol.19, Issue3, pp.223-244, 2007.
- [2] S. E. Fahlman and C. Lebiere, "The cascade-correlation learning architecture," S. Touretzky, Editor, *Advances in Neural Information Processing Systems2*, Morgan Kaufmann, 1990.
- [3] S. E. Fahlman, "The Recurrent Cascade-Correlation Architecture," *Advances in Neural Information Processing Systems3*, Morgan Kaufmann, pp.190-198, 1991.
- [4] X. Z. Gao, X. Wang, and S. J. Ovaska, "A novel hybrid optimization method with application in Cascade-Correlation neural network training," *Proceedings, 8th International Conference on Hybrid Intelligent Systems*, Article, No.4626728, pp.793-800, 2008.
- [5] B. Hammer, A. Micheli, and A. Sperduti, "Universal approximation capability of cascade correlation for structures," *Neural Computation*, Vol.17, Issue5, pp.1109-1159, 2005.
- [6] T. D. Le, T. Komeda, and M. Takagi, "Knowledge-based recurrent neural networks in reinforcement learning," *Proceedings of the 11th IASTED International Conference on Artificial Intelligence and Soft Computing*, pp.169-174, 2007.
- [7] L. Prechelt, *PROBEN1-A Set of Neural*

*Network Benchmark Problems and Benchmarking Rules*, Technical Report, University of Karlsruhe, 1999(9).

- [8] Stuttgart Neural Network Simulator(SNNS), *User Manual, Version 4.0*, Institute for Parallel and Distributed High Performance Systems (IPVR), University of Stuttgart, 1998.
- [9] 이상화, “캐스케이드 코릴레이션 알고리즘의 일반화와 새로운 활성화 함수를 사용한 실험”, 정보과학회논문지, 제25권, 제7호, B, 1998.
- [10] 이상화, 송해상, “순환 캐스케이드 코릴레이션 알고리즘의 일반화와 새로운 활성화함수를 사용한 모스 신호 실험”, 한국지능정보시스템학회논문지, 제10권, 제2호, pp.53-63, 2004(11).

**저 자 소 개**

**이 상 화 (Sang-Wha Lee)**

정회원



- 1985년 10월 ~ 1998년 11월 : 도르트문트대학교 컴퓨터공학과 (공학사, 공학석사, 공학박사)
- 2002년 9월 ~ 현재 : 서원대학교 정보통신공학과 부교수
- 2008년 7월 ~ 현재 : 한국전자통신연구원(ETRI) 초빙연구원

<관심분야> : 인공지능, 지능형 시스템, RFID/USN

**송 해 상 (Hae-Sang Song)**

정회원



- 2000년 8월 : 한국과학기술원 컴퓨터공학박사
- 1999년 1월 ~ 2000년 8월 : 고등기술연구원
- 2000년 9월 ~ 2002년 8월 : (주)스페이스네트 연구소장

▪ 2002년 9월 ~ 현재 : 서원대학교 컴퓨터공학과 부교수

<관심분야> : 소프트웨어공학, 임베디드시스템