

# 이중나선의 패턴 인식 분석과 CosExp와 시그모이드 활성화 함수를 사용한 캐스케이드 코릴레이션 알고리즘의 최적화

이상화\*

<sup>1</sup>서원대학교 정보통신공학과

## Pattern Recognition Analysis of Two Spirals and Optimization of Cascade Correlation Algorithm using CosExp and Sigmoid Activation Functions

Sang-Wha Lee<sup>1\*</sup>

<sup>1</sup>Department of Information & Communication Engineering, Seowon University

**요약** 본 논문에서는 비모노톤함수(non-monotone function)인 CosExp(cosine-modulated symmetric Exponential function) 함수와 모노톤함수(monotone function)인 시그모이드 함수를 캐스케이드 코릴레이션 알고리즘(Cascade Correlation algorithm)의 학습에 병행해서 사용하여 이중나선문제(two spirals problem)의 패턴인식에 어떠한 영향이 있는지 분석하고 이어서 알고리즘의 최적화를 시도한다. 첫 번째 실험에서는 알고리즘의 후보뉴런에 CosExp 함수를 그리고 출력뉴런에는 시그모이드 함수를 사용하여 나온 인식된 패턴을 분석한다. 두 번째 실험에서는 반대로 CosExp 함수를 출력뉴런에서 사용하고 시그모이드 함수를 후보뉴런에 사용하여 실험하고 결과를 분석한다. 세 번째 실험에서는 후보뉴런을 위한 8개의 풀을 구성하여 변형된 다양한 시그모이드 활성화 함수(sigmoidal activation function)를 사용하고 출력뉴런에는 CosExp 함수를 사용하여 얻게 된 입력공간의 인식된 패턴을 분석한다. 네 번째 실험에서는 시그모이드 함수의 변위를 결정하는 세 개의 파라미터 값을 유전자 알고리즘을 이용하여 얻는다. 이 파라미터 값들이 적용된 시그모이드 함수들은 후보뉴런의 활성화를 위해서 사용되고 출력뉴런에는 CosExp 함수를 사용하여 실험한 최적화 된 결과를 분석한다. 이러한 알고리즘의 성능평가를 위하여 각 학습단계 마다 입력패턴공간에서 인식된 이중나선의 형태를 그래픽으로 보여준다. 최적화 과정에서 은닉뉴런(hidden neuron)의 숫자가 28에서 15로 그리고 최종적으로 12개로 줄어서 학습 알고리즘이 최적화되었음을 확인하였다.

**Abstract** This paper presents a pattern recognition analysis of two spirals problem and optimization of Cascade Correlation learning algorithm using in combination with a non-monotone function as CosExp(cosine-modulated symmetric exponential function) and a monotone function as sigmoid function. In addition, the algorithm's optimization is attempted. By using genetic algorithms the optimization of the algorithm will attempt. In the first experiment, by using CosExp activation function for candidate neurons of the learning algorithm is analyzed the recognized pattern in input space of the two spirals problem. In the second experiment, CosExp function for output neurons is used. In the third experiment, the sigmoid activation functions with various parameters for candidate neurons in 8 pools and CosExp function for output neurons are used. In the fourth experiment, the parameters are composed of 8 pools and displacement of the sigmoid function to determine the value of the three parameters is obtained using genetic algorithms. The parameter values applied to the sigmoid activation functions for candidate neurons are used. To evaluate the performance of these algorithms, each step of the training input pattern classification shows the shape of the two spirals. In the optimizing process, the number of hidden neurons was reduced from 28 to 15, and finally the learning algorithm with 12 hidden neurons was optimized.

**Key Words** : Activation function, Cascade correlation algorithm, Cosine-modulated symmetric exponential function, Sigmoid function

\*Corresponding Author : Sang-Wha Lee(Seowon Univ.)

Tel: +82-10-4312-4567 email: swl@seowon.ac.kr

Received January 7, 2014

Revised (1st February 4, 2014, 2nd February 14, 2014)

Accepted March 6, 2014

### 1. 서론

본 논문에서 패턴인식 분야에서 움직이는 물체 인식의 한 해결 방법으로서 Fahlman이 제안한 캐스케이드 코릴레이션(Cascade correlation(CC)) 학습 알고리즘[1]을 CosExp와 시그모이드 활성화 함수를 사용하여 실험하고 분석한 후에 알고리즘의 최적화를 시도한다. 이러한 실험을 위한 기준문제(benchmark problem)로서 이중나선문제(two spirals problem)를 사용하였다[1-3]. 이 알고리즘의 초기 네트워크는 은닉뉴런과 은닉층 없이 입력뉴런과 출력뉴런의 완전한 연결에 의하여 구성된다. 은닉뉴런들은 학습 중에 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. 이 후보뉴런의 출력은 아직 네트워크와는 연결되지 않은 상태이다. 네트워크는 주어진 패턴들에 대하여 한 번의 학습 후에 후보뉴런들의 입력가중치(input weight)를 수정한다.  $S_i$ 는 한 후보뉴런  $i$ 에서 생성된 값  $c$ 와 모든 출력뉴런  $o$ 에서 측정된 잔여출력오차  $E_o$ 의 상호관계(correlation)의 합계로 나타낸다.  $S_i$ 의 극대화를 위한 수정은 다음과 같이 정의한다.

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

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.

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

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

$\sigma_o$ 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.

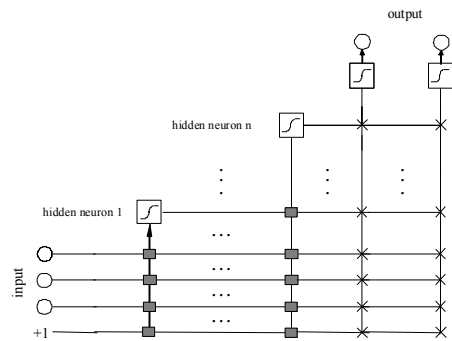
$$\frac{\partial S_i}{\partial w_{ij}} = \sum_o \sum_p \sigma_o \frac{\partial}{\partial net_{pi}} f_p(net_{pi}) \frac{\partial net_{pi}}{\partial w_{ij}} (E_{po} - \bar{E}_o)$$

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[1,3].



[Fig. 1] Cascade Correlation Algorithm

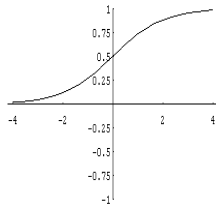
Fig. 1은 학습을 통하여  $n$ 개의 은닉뉴런이 생성되어 구성되는 캐스케이드 코릴레이션 네트워크를 보여준다. 그림에서 사각형으로 표시된 연결들은 고정되고 한번 선택된 가중치는 다음 학습단계에서도 수정 없이 계속 사

용된다. x와 함께 표시한 연결들은 반복해서 학습 한다. 여기에서 각 학습단계마다 가중치는 다시 수정된다.

논문 [5]에서는 활성화 함수로 가장 알려지고 일반적으로 많이 사용하는 모노톤 함수(monotone function)인 시그모이드 함수만을 사용하여 이중나선의 패턴인식 상태를 알아보는 실험을 하였다. 그러나 본 논문에서는 [5]의 확장 버전으로서 비모노톤함수(non-monotone function)인 코사인 모듈화 된 지수함수(CosExp)와 모노톤함수인 시그모이드 함수를 병행해서 사용하여 이중 나선이 어떻게 인식 되는지를 분석하고, 또한 그에 따른 알고리즘의 최적화를 시도하는 것을 목적으로 한다. 이러한 분석 평가를 위해서 첫 번째 주어진 이중나선의 모양과 가급적 똑같은 형태의 인식된 패턴의 모양을 구분해 내야하고 두 번째는 가급적 작은 수의 은닉뉴런을 생성하여 네트워크를 구성하여야한다. 이를 위하여 각 학습단계마다 입력패턴공간에서 인식된 이중나선의 상태를 그림으로 보여줌으로써 단계별 학습 과정의 인식도를 확인할 수 있도록 하였고 또한 생성된 은닉뉴런의 수를 비교하였다.

캐스케이드 코릴레이션에 사용할 활성화 함수인 시그모이드 함수는 다음과 같이 정의된다.

$$f(x) = \frac{1}{1 + e^{-ax+b}} + c$$



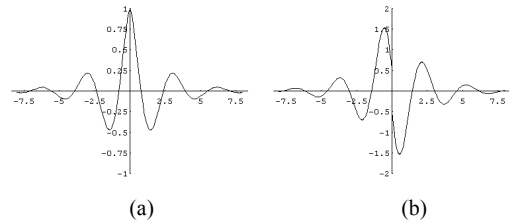
[Fig. 2] Sigmoid function(a=1, b=0, c=0)

또한 CosExp(Cosine-Modulated Symmetric Exponential Function) 함수는 다음과 같이 정의된다.

$$f(x) = e^{-b|x-d|} \cos(c|x-d|)$$

$$f'(x) = \frac{d-x}{|d-x|} e^{-b|x-d|} (b \cos(c|x-d|) + c \sin(c|x-d|)) \quad (d-x \neq 0)$$

Fig. 3의 함수에서 b 는 지수함수의 경사율을 나타내고, d는 초평면(hyperplane)의 위치를 결정한다. CosExp 함수의 최대값은 항상 1이며 c는 주기의 길이를 조절한다.



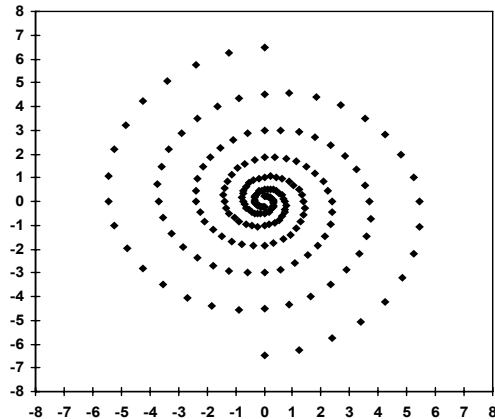
[Fig. 3] (a): CosExp function(b=0.5, c=2, d=0), (b): derivation of the function

CosExp 함수의 특징:

- 0을 제외한 부분에서 연속이고 미분가능이다
- 축대칭(axial symmetry)이다
- f는 한 개 또는 여러 개의 전환점(turning point)을 갖고 있다
- 전환점들은  $input \rightarrow \pm \infty$ 에 대하여 점근적(asymptotically)인 포화값(saturation value) 0으로 수렴 한다.

## 2. 이중나선문제

이중나선의 알고리즘은 연속된 값으로 이루어진 두 개의 입력을 갖고 한 개의 이원값의 출력을 갖는다. 서로 끼워 맞추어진 두 개의 나선은 학습을 통하여 구분하는 것이 목표이다. 각 나선은 97개의 측량점으로 이루어졌으며 3 회전 되었다. 따라서 32개의 측량점들과 함께 각 나선은 한번 회전 한다. 그리고 마지막 한 개의 점이 첨가된다. 각 점은 x축과 y축에 의하여 두 개의 실수값으로 나타낸다. 네트워크의 출력은 한 나선을 위하여 1.0 그리고 다른 나선을 위하여 0.0의 값을 나타낸다. Fig. 4는 이중나선문제를 위한 트레이닝 점들을 표시하였다[4,5].



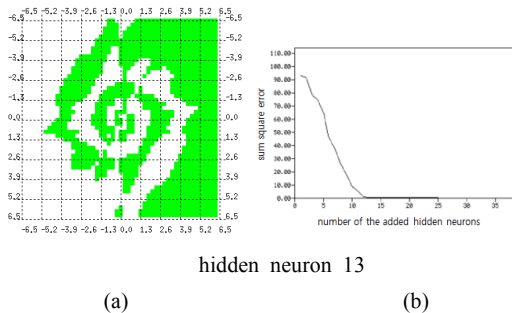
[Fig. 4] training points for two spiral problem

### 3. CosExp 함수를 사용한 실험

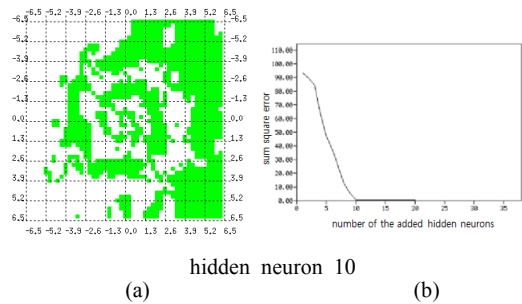
#### 3.1 후보뉴런에서 사용

캐스케이드 코릴레이션 알고리즘은 이중나선문제의 패턴을 인식하기 위하여 하나의 풀(pool)을 구성하고 활성화 함수로는 시그모이드 함수를 사용하였다[1]. 보통 하나의 풀은 4~8개의 후보뉴런으로 구성하는데 같은 활성화 함수를 사용한다. 한 풀에 10개 이상의 많은 후보뉴런을 형성하였다고 해서 반드시 항상 좋은 결과가 나오는 것은 아니다. 따라서 후보뉴런은 4~5개 정도가 적당하고 최대 8개 정도까지 풀을 구성하는 것이 일반적이다. 후보뉴런들은 모든 입력뉴런과, 바이어스와 그리고 이미 생성된 모든 은닉뉴런과 연결한다. 풀에 있는 모든 후보뉴런들은 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. 학습 중에 뉴런의 학습 횟수가 주어진 파라미터값에 도달했을 때 가장 큰 상관관계값을 갖고 있는 한 후보뉴런이 은닉뉴런으로 선택되어 네트워크에 첨가되고 다른 뉴런들과 연결된다.

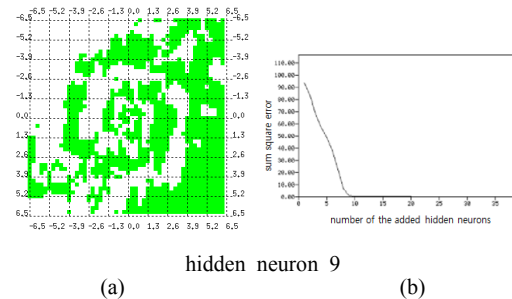
첫 번째 실험에서 Fig. 5는 Fahlman의 실험과 같이 시그모이드 함수만을 사용하여 패턴 인식을 한 결과이다. 이에 비해 Fig. 6 ~ Fig. 8은 후보뉴런에 CosExp 함수에 파라미터 값을 달리하여 실험을 하였는데 시그모이드 함수에 비해 더 빠르게 학습을 하였음을 알 수 있다. 하지만 이중나선의 패턴인식이 깨끗하지 못한 입력공간의 모습을 보여주고 있다. 따라서 후보뉴런의 활성화함수는 시그모이드 함수를 사용하는 것이 더 효율적이라는 판단을 할 수 있다.



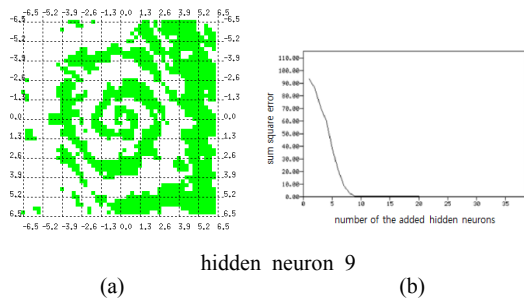
[Fig. 5] (a): two spiral pattern recognition in the experiment by using sigmoid activation function  
(b): comparison between the learning error and the number of produced hidden neurons



[Fig. 6] (a): two spiral pattern recognition in the experiment by using CosExp activation function( $b=0.1$ ,  $c=1$  and  $d=0$ )  
(b): comparison between the learning error and the number of produced hidden neurons



[Fig. 7] (a): two spiral pattern recognition in the experiment by using CosExp activation function( $b=0.1$ ,  $c=1.25$  and  $d=0$ )  
(b): comparison between the learning error and the number of produced hidden neurons

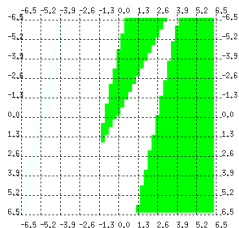


[Fig. 8] (a): two spiral pattern recognition in the experiment by using CosExp activation function( $b=0.01$ ,  $c=1$  and  $d=0$ )  
(b): comparison between the learning error and the number of produced hidden neurons

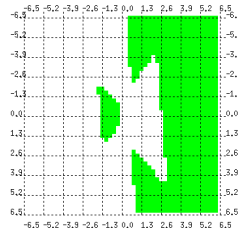
#### 3.2 출력뉴런에서 사용

두 번째 실험에서는 후보뉴런에는 시그모이드 함수를 사용하고 출력뉴런에는 CosExp 함수를 사용하여 진행하였다. 실험의 결과로서 Fig. 9는 입력공간의 이중나선 패

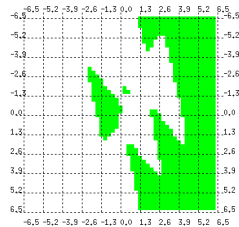
턴 인식 상태를 보여주고, Fig. 10은 학습오차와 생성된 은닉뉴런 수를 비교하였다.



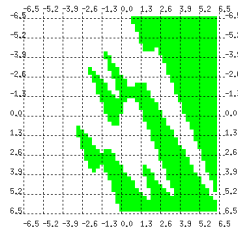
hidden neuron 1



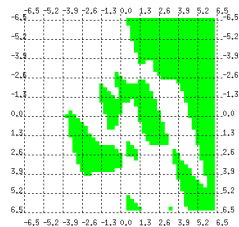
hidden neuron 2



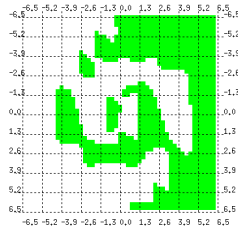
hidden neuron 3



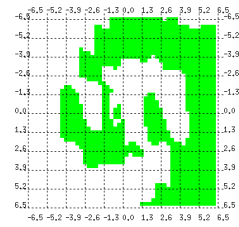
hidden neuron 4



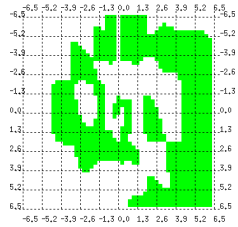
hidden neuron 5



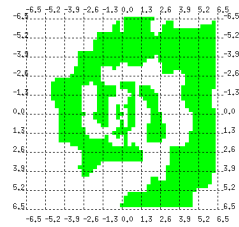
hidden neuron 6



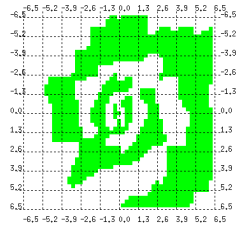
hidden neuron 7



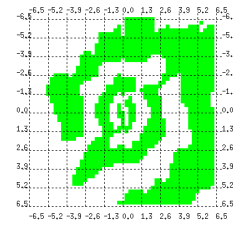
hidden neuron 8



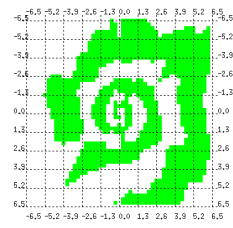
hidden neuron 9



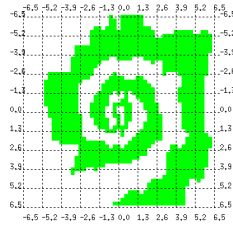
hidden neuron 10



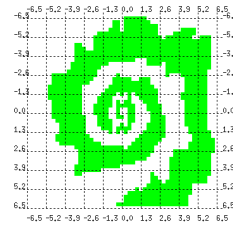
hidden neuron 11



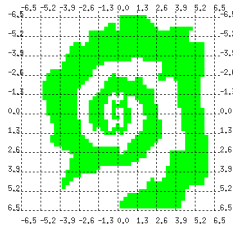
hidden neuron 12



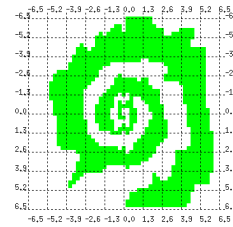
hidden neuron 13



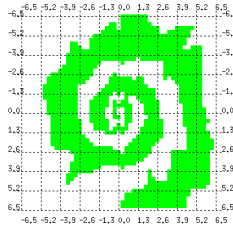
hidden neuron 14



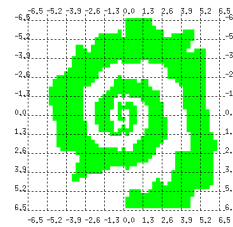
hidden neuron 15



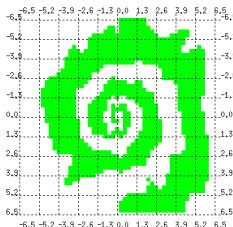
hidden neuron 16



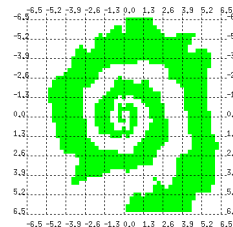
hidden neuron 17



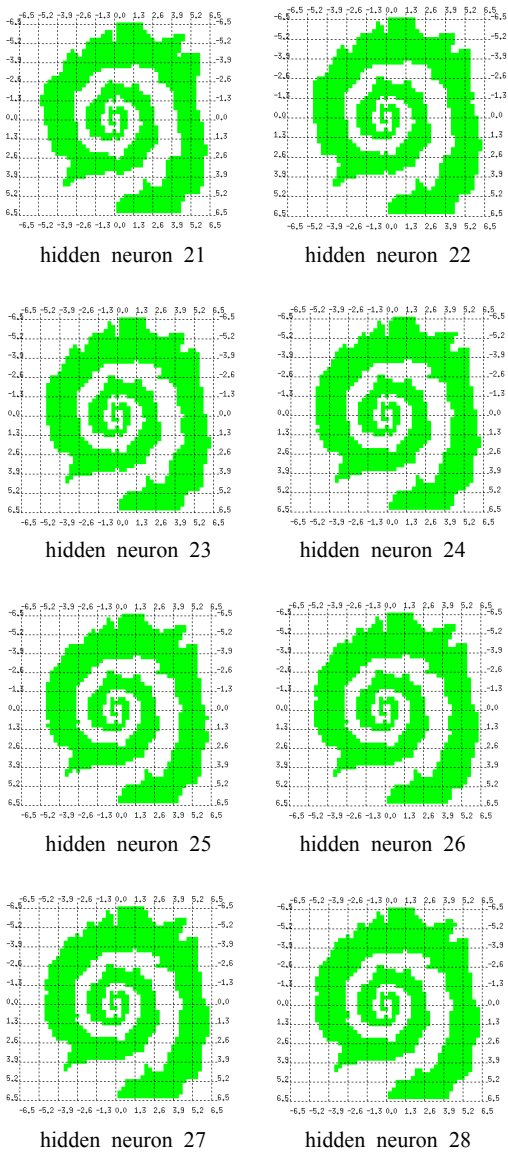
hidden neuron 18



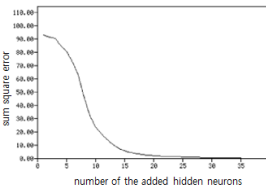
hidden neuron 19



hidden neuron 20



[Fig. 9] two spiral pattern recognition of the input space in the experiment by using CosExp activation function( $b=0.1$ ,  $c=5$  and  $d=0$ )



[Fig. 10] comparison between the learning error and the number of produced hidden neurons in the experiment by using CosExp activation function( $b=0.1$ ,  $c=5$  and  $d=0$ )

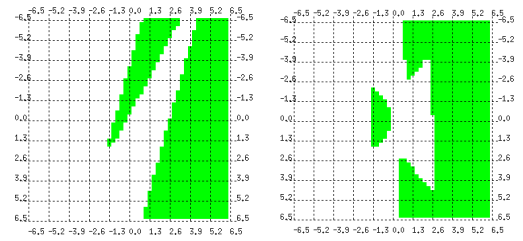
Fig. 9에서와 같이 입력공간에서의 이중나선 패턴 인식은 아주 잘 되었으나 28개의 은닉뉴런을 생성하여 학습 속도의 저하로 큰 네트워크를 형성하였다. 따라서 이의 개선을 위하여 다중풀의 구성과 여러 형태의 변형된 시그모이드 함수를 사용하여 실험하였다.

#### 4. 알고리즘의 최적화를 위한 다중 풀의 구성과 다양한 시그모이드 함수의 사용

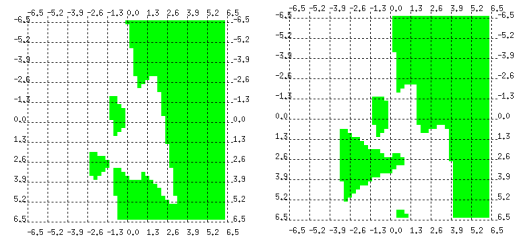
세 번째 실험에서는 캐스케이드 코릴레이션 학습 알고리즘의 성능 향상을 위하여 후보뉴런들의 활성화함수인 시그모이드 함수의 파라미터 값에 따라서 8개의 풀을 구성한다. 각 풀에는 4개의 후보뉴런들로 이루어지고, 풀에 따라서 Table 1에 있는 시그모이드 함수의 파라미터 a, b, c의 값이 부여된다.

[Table 1] Value of parameter given to candidate neurons in the pool

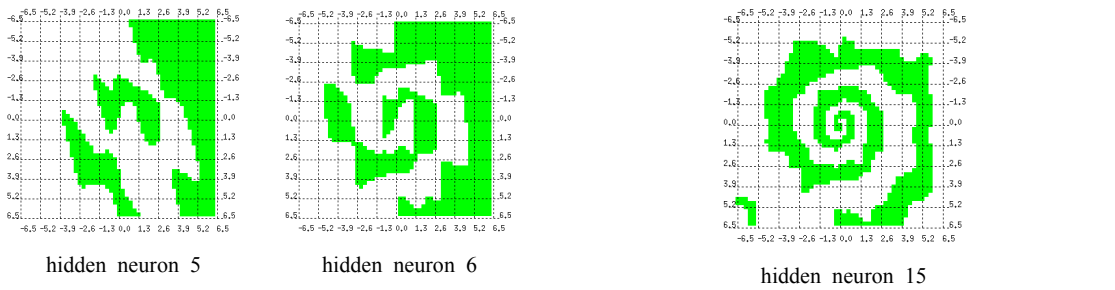
pool name	value a	value b	value c
pool a	1	0	0
pool b	1	1	0
pool c	1	2	0
pool d	1	-1	0
pool e	1	-2	0
pool f	1	0	0.2
pool g	1	0	-0.2
pool h	1	0	-0.4



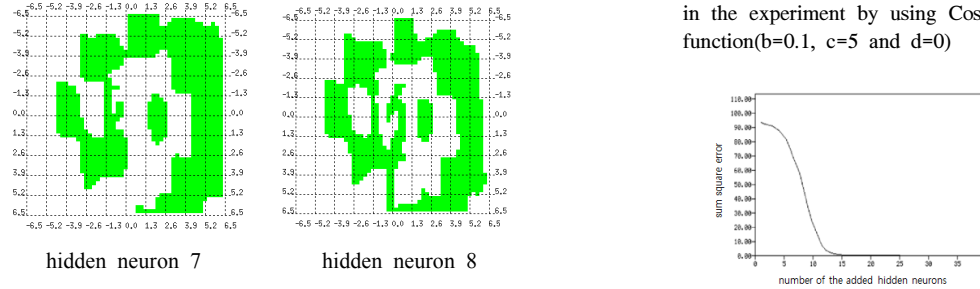
hidden neuron 1                      hidden neuron 2



hidden neuron 3                      hidden neuron 4



[Fig. 11] two spiral pattern recognition of the input space in the experiment by using CosExp activation function( $b=0.1$ ,  $c=5$  and  $d=0$ )

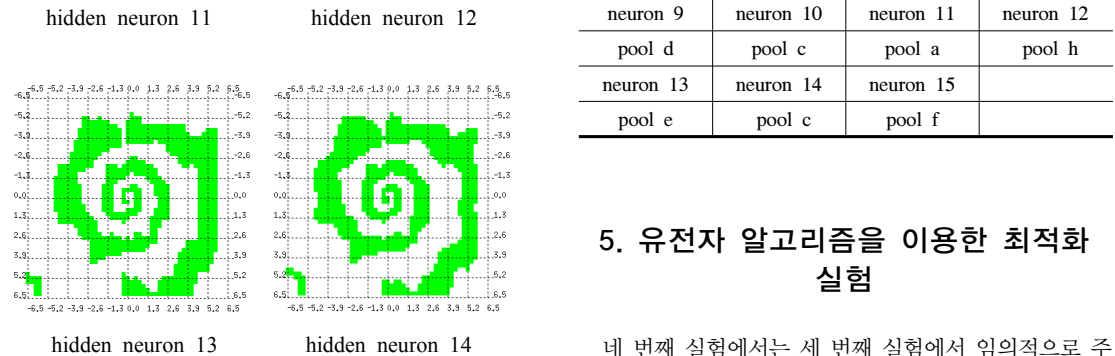


[Fig. 12] comparison between the learning error and the number of produced hidden neurons in the experiment by using CosExp activation function( $b=0.1$ ,  $c=5$  and  $d=0$ )

Table 2에서는 학습 중에 선택되어진 후보뉴런이 속한 풀을 나타내었다. 이 실험에서는 28개의 은닉뉴런이 필요했던 학습에서 개수가 15개 까지 줄었고 패턴 인식 또한 아주 잘 된 것을 볼 수 있다.

[Table 2] Selected activation functions during learning

Selected Activation Functions			
neuron 1	neuron 2	neuron 3	neuron 4
pool g	pool b	pool e	pool e
neuron 5	neuron 6	neuron 7	neuron 8
pool e	pool d	pool g	pool g
neuron 9	neuron 10	neuron 11	neuron 12
pool d	pool c	pool a	pool h
neuron 13	neuron 14	neuron 15	
pool e	pool c	pool f	



### 5. 유전자 알고리즘을 이용한 최적화 실험

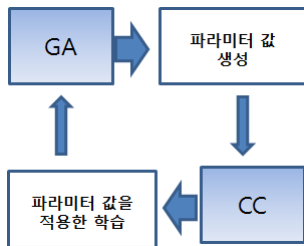
네 번째 실험에서는 세 번째 실험에서 임의적으로 주었던 시그모이드 함수의 파라미터 값을 매 트레이닝

(training)마다 유전자 알고리즘(Genetic Algorithm(GA))을 이용하여 구하고 이를 CC 알고리즘의 패턴 인식 학습에 적용한다. 이 실험에서 사용한 유전자알고리즘의 의사 코드는 다음과 같다[5].

```

Algorithm GA is
t:=0;
initialized Population P(t);
evaluate P(t);
while not finished (P(t)) do
    t:=t+1;
    P':=select Parents from P(t);
    recombine P'(t)
    with Probability Pc;
    mutate P'(t)
    with Probability Pm;
    evaluate P'(t);
    P:=P';
endwhile
end GA.
    
```

Fig. 13은 GA와 CC의 두 프로세스의 상호 관계를 나타내는데 GA를 통해서 파라미터의 값이 생성되고 이를 CC에 적용하여 이중나선의 패턴을 인식하는 학습을 하게 된다. 이를 Fig. 13에 나타내었다. 여기에서 GA와 CC의 프로세스는 병렬로 진행된다[5].



[Fig. 13] Optimization process of algorithm

유전자 알고리즘의 실행을 위하여 주어진 파라미터 값들은 다음과 같다.

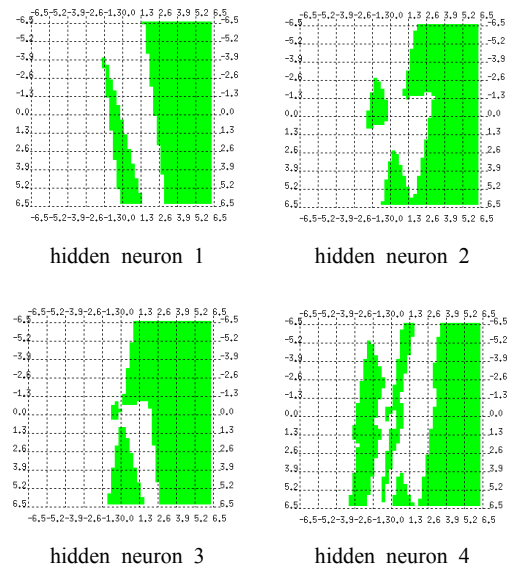
- 크로모솜(염색체)의 길이: 32
- 모집단크기: 50
- 돌연변이비율: 0.001
- 크로스오버 포인트의 수: 2
- 크로스오버의 사용 비율: 0.6
- 세대 수: 100

후보뉴런을 위하여 사용할 시그모이드 활성화함수의 파라미터 a, b 와 c의 값을 유전자알고리즘을 이용하여 구할 때 이중나선문제의 입력공간 구간인 -6.5에서 6.5까지의 전 구간에서 파라미터값을 선택하면 알고리즘의 트레이닝이 불안하고 학습에 많은 은닉뉴런을 필요로 한다. 이는 활성화 함수의 분배가 입력패턴의 전 공간에서 이루어지기 때문에 좋은 패턴인식 결과를 얻을 수 없다. 따라서 시그모이드 함수에서 이중나선문제의 입력패턴공간을 고려하여 함수의 파라미터 a의 값은  $0.5 \leq a \leq 1.5$ , 파라미터 b의 값은  $-2.5 \leq b \leq 2.5$ , 그리고 파라미터 c의 값은  $-2.5 \leq c \leq 2.5$ 의 구간으로 한정한다.

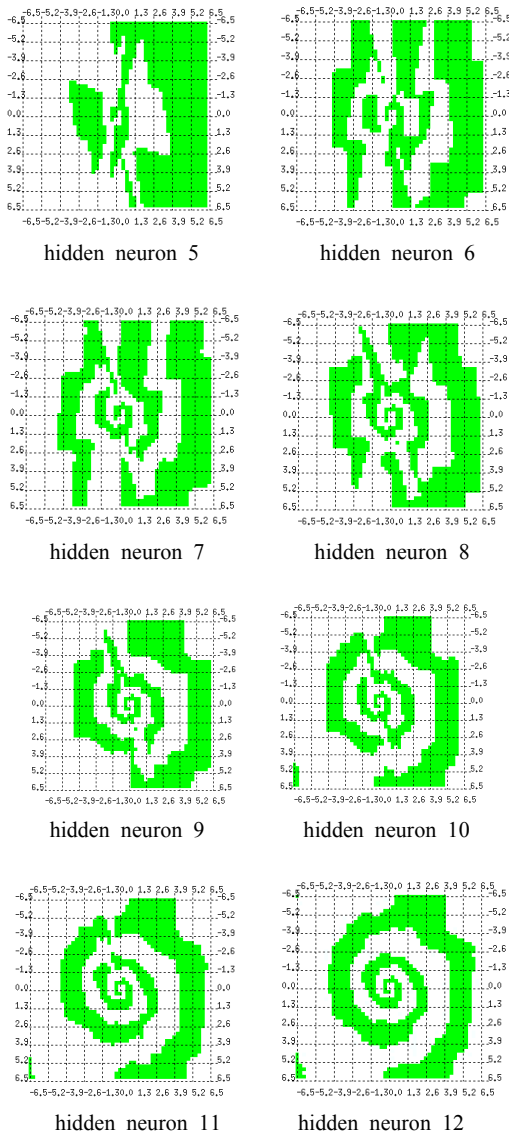
Fig. 14는 입력공간의 이중나선 패턴 인식 상태를 보여주고 Fig. 15는 학습오차와 생성된 은닉뉴런 수의 비교를 나타내었다. Table 3은 학습 중에 선택된 시그모이드 활성화 함수의 파라미터 값을 기록하였다.

[Table 3] Selected activation functions during learning

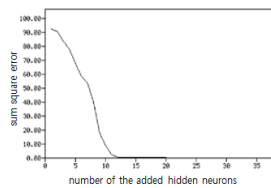
Parameter values of selected activation functions	
hidden neuron 1 (0.9, -1.0, 1.2)	hidden neuron 2 (1.1, -2.0, 1.6)
hidden neuron 3 (0.9, -1.1, 1.5)	hidden neuron 4 (1.3, 0.1, -0.4)
hidden neuron 5 (1.5, -1.9, 0.2)	hidden neuron 6 (0.8, 1.2, 1.7)
hidden neuron 7 (1.1, 1.6, -0.7)	hidden neuron 8 (0.7, -2.1, 1.3)
hidden neuron 9 (1.2, 0.5, 1.3)	hidden neuron 10 (1.3, -1.2, 0.9)
hidden neuron 11 (1.0, -0.3, -1.5)	hidden neuron 12 (0.8, -1.3, 2.0)







[Fig. 14] two spiral pattern recognition of the input space in the experiment by using CosExp activation function( $b=0.1$ ,  $c=5$  and  $d=0$ )



[Fig. 15] comparison between the learning error and the number of produced hidden neurons in the experiment by using CosExp activation function ( $b=0.1$ ,  $c=5$  and  $d=0$ )

캐스케이드 코릴레이션 알고리즘의 최적화를 위한 네 번째 실험을 분석해보면 임의로 파라미터 값을 주었던 세 번째 실험에 비해서 은닉뉴런의 수가 3개나 줄었고 또한 이중나선의 입력공간 패턴인식도 더 좋게 되었다. 이를 통하여 알고리즘이 최적화 되었다는 것을 확인 할 수 있었다.

## 6. 결론

CosExp 함수를 캐스케이드 코릴레이션 알고리즘에 사용하는 것은 후보뉴런보다 출력뉴런에서 더 효율적이라는 것을 보여주었다. 또한 이중나선 문제 해결의 최적화를 위하여 기본 시그모이드 함수만 사용하는 것 보다 여러 개의 다양하게 변형된 시그모이드 활성화 함수를 사용하는 것이 알고리즘의 성능 향상과 이중나선의 패턴 인식에 영향을 주는 것을 보여주었다. 이를 각 학습과정에서 패턴 분류(classification)를 통하여 패턴을 인식하는 과정을 단계별로 보여줌으로써 확인할 수 있었다. 은닉뉴런의 수를 줄이기 위한 최적화를 위하여 유전자 알고리즘을 이용하여 파라미터 값을 구하여 학습에 사용하면 더 좋은 결과를 얻을 수 있었다. 이러한 실험을 통하여 패턴 인식 능력을 최적화 시킬 수 있음을 본 논문에서 보여 주었다.

## References

- [1] Fahlman, S. E. and Lebiere, C., "he Cascade-Correlation Learning Architecture", Advances in Neural Information Processing Systems 2, D. S. Touretzky (ed.), Morgan-Kaufmann, Los Altos CA., 1990.
- [2] Prechelt, L., "PROBEN1-A Set of Neural Network Benchmark Problems and Benchmarking Rules", Technical Report 21/94, Department of Computer Science, University of Karlsruhe, September 30, 1999.
- [3] Thomas R. Shultz, Scott E. Fahlman, "Cascade-Correlation", Encyclopedia of Machine Learning, pp.139-147, 2010
- [4] Ikuta, C, Uwate Y. and Nishio Y., "Chaos Glial Network connected to Multi-Layer Perceptron for Solving Two-Spiral Problem", Proc. ISCAS'10, May, 2010.
- [5] Sang-Wha Lee, "Optimization of sigmoid activation Function parameters using genetic algorithms and pattern recognition analysis in input space of two spirals problem", The Korea contents association, v.10, No.4,

pp.10-18, 4. 2010.

DOI: <http://dx.doi.org/10.5392/JKCA.2010.10.4.010>

---

이 상 화(Sang-Wha Lee)

[정회원]



- 1994년 10월 : 도르트문트대학교 (컴퓨터공학석사)
- 1998년 11월 : 도르트문트대학교 (컴퓨터공학박사)
- 2000년 2월 ~ 2002년 8월 : 한국전자통신연구원 선임연구원
- 2002년 9월 ~ 현재 : 서원대학교 정보통신공학과 교수
- 2008년 7월 ~ 2009년 6월 : 한국전자통신연구원 초빙연구원

<관심분야>

정보통신, 인공지능, 컴퓨터 네트워크, RFID/USN