

論文96-33B-10-10

다중컴퓨터망에서 SOFM 신경회로망의 병렬구현 및 성능평가

(Parallel Implementations and their Performance Evaluations of a SOFM Neural Network on the Multicomputer)

金宣鍾*, 崔興文**

(Seon Jong Kim and Heung Moon Choi)

요 약

본 논문에서는 SOFM신경회로망의 학습시스템을 그 네트워크 병렬성 및 학습집합 병렬성에 따라 분할하여 각각을 다중컴퓨터망에서 병렬구현하고, 신경회로망의 크기(연결세기의 수) N , 학습집합의 크기(학습패턴의 수) L 및 프로세서의 수 p 의 증가에 따른 시스템의 병렬처리 성능변화를 조사하였다. 각 분할방법별 병렬구현에 대해 성능평가 모델을 구하여 해석적으로 성능평가하고, 실험을 통하여 각 구현방식의 병렬처리 성능을 구하여 이들을 비교 고찰하였다. 링구조에서의 해석적 성능평가 결과, 망 분할과 학습집합 분할을 이용할 때 최고의 성능을 얻기 위한 프로세서의 수는 각각 \sqrt{N} 및 \sqrt{L} 에 비례하여 증가하였다. 특히, 두 분할 방법에 따른 병렬 SOFM의 성능은 N 과 L 의 크기에 따라 좌우되었다.

Abstract

This paper presents an efficient parallel implementation and its performance evaluations of a SOFM neural network on the multicomputer. We investigate the parallel performances as the size of a neural network N , the number of the patterns L , and the number of the processors p increase. We propose an analytical performance evaluation model for each of the parallel implementations and verified the validity of the model through experiments. Analytical results show that the number of processors for a maximum speedup of the network decomposition and the training-set decomposition increases in proportion to \sqrt{N} and \sqrt{L} , respectively. The performances of the both decompositions depend on the number of training patterns L and the size of the neural network N and, if $L \geq 0.423N$, the performance of the training-set decomposition is proved to be better than that of the network decomposition.

I. 서 론

Kohonen의 SOFM(self-organizing feature ma

* 正會員, 順天工業專門大學 制御計測科

(Dept. of Control and Instrumentation, Suncheon Technical Junior College)

** 正會員, 慶北大學校 電子電氣工學部

(School of Electronic and Electrical Engineering, Kyungpook National Univ.)

接受日字:1996年1月9日, 수정완료일:1996年10月2日

ps) 신경회로망^[1]은 입력층과 출력층으로 구성되며, 입력뉴런과 출력뉴런들 사이에 완전결합(fully connected)형태의 연결세기망을 갖고 있다. 학습되어야 할 패턴 수가 많거나 신경회로망의 크기가 커져 학습할 연결세기의 수가 많아지면, 그에 따라 학습시간이 대단히 길어진다. 따라서 병렬화 등 효과적인 고속화 방안이 강구되어야 한다. SOFM 신경회로망은 VLSI, 시스틀릭 어레이 프로세서, 그리고 다중프로세서나 다중컴퓨터망을 이용하여 병렬구현되고 있다^[2-5]. VLSI 또는 시스틀릭 어레이를 이용한 병렬처리 시스템은 실

시간 응용에 적합하지만, 일단 하드웨어 구현후에는 구조가 고정되므로 그 융통성 및 대규모 시스템으로의 확장성이 떨어진다. 반면 다중프로세서나 다중컴퓨터망을 이용하여 구현하면 망의 크기나 학습패턴의 수 등 매개변수가 변화할 경우에도 적응적으로 대처할 수 있으며, 대규모 시스템으로의 확장도 용이하므로, 다중컴퓨터망은 SOFM 신경회로망의 병렬 시뮬레이션 및 병렬구현에 효과적으로 이용되고 있다.

SOFM 신경회로망의 학습시스템은 그 내재된 병렬성에 따라 망 분할(network decomposition)^[2,3] 또는 학습집합 분할(training-set decomposition)^[4,5]하여 병렬구현할 수 있다. 망 분할방법은 SOFM 신경회로망 자체를 몇개의 부회로망(sub-network)으로 분할하여 각각을 각 프로세서에 할당구현하는 방법으로, 뉴런과 뉴런사이의 완전연결을 뒷받침하기 위한 프로세서간의 복잡한 통신부담을 감당해야 하므로, 신경회로망의 규모가 커져 분할된 부회로망 수가 증가하면 프로세서의 수가 많아지고, 이 프로세서간의 통신량이 급격히 증가한다. 특히, 승리뉴런을 결정하는데 프로세서간의 순서적인 통신이 요구되어 속도제고율(speedup factor) S_p 의 증가에 한계가 있다. 따라서 프로세서의 수가 많아지면 이러한 통신량이 급증하게 되고, 전체적인 학습속도가 늦어져 병렬처리 성능이 오히려 포화될 수 있다^[2,4].

학습집합 분할방법은 학습시킬 패턴의 전체학습집합(total training-set)을 몇개의 부학습집합(partial training-set)으로 나누고 각 부집합을 온전한 SOFM 신경회로망을 갖는 각 프로세서에 할당하여 병렬처리하는 방법으로서 전체집합에 대한 학습을 위해서는 네트워크 전체의 정보를 모든 프로세서간에 통신해야 하므로 그에 따른 메모리와 통신시간이 많이 요구된다^[6,7]. 온전한 SOFM 신경회로망이 프로세서마다 할당되어 전체적인 신경회로망의 규모가 증가할 뿐만 아니라 학습집합 전체를 학습하기에 필요한 정보를 얻기 위하여 각 프로세서가 갖고있는 신경회로망의 크기만큼의 상호 통신량이 필요하다. 따라서 통신량은 각 프로세서에 할당된 신경회로망 크기에 좌우되며, 그 크기가 커지면 통신량도 그만큼 증가하게 된다. 또한 전체 학습패턴에 대한 연결세기의 갱신(update)은 각 패턴에 대해서 갱신된 연결세기들을 선형적으로 합하기 때문에, 프로세서의 수가 증가할수록 합해진 연결세기의 크기가 커질 수 있고 따라서 수렴에 문제가 발생할 수

있다. Paugam 등^[9]은 역전파 신경회로망(backpropagation)에 학습집합 분할방법을 적용하였는데, 학습패턴의 수가 많아지면 그 학습률의 크기를 줄였고 따라서 갱신된 연결세기의 크기를 줄여 수렴이 보장될 수 있도록 하였다.

본 논문에서는 SOFM신경회로망의 학습시스템을 망 분할 및 학습집합 분할하여 각각을 다중컴퓨터망에 병렬구현하고 성능평가하였다. 먼저 각 분할방법에 따른 성능평가 모델을 구하고, 링구조에 대하여 해석적 방법과 실험을 통하여 각 구현방식의 병렬처리 성능을 구하여 비교 고찰하였다. 또한 신경회로망의 크기, 학습집합의 크기 및 프로세서의 수의 증가에 따른 시스템의 병렬처리 성능변화를 조사비교하고, 병렬시스템 설계에서의 분할방법 선택을 위한 기준을 마련하였다. 그리고 학습집합 분할시의 효과적인 학습을 위한 학습률의 크기도 조사하였다.

II. SOFM 신경회로망의 병렬분할

SOFM 신경회로망은 그림 1과 같이 입력뉴런들과 출력뉴런들을 갖는 단층구조로서, 입력뉴런 수를 n_i 와 출력뉴런 수를 n_o 라 할 때, 망의 크기는 연결세기의 수 N

$$N = n_o n_i \quad (1)$$

로 결정된다. 또한 전체 학습집합은 L 개의 패턴으로 구성되며, 각 패턴은 입력뉴런의 수 n_i 개의 입력요소를 갖는다. 망의 크기가 N 이고 학습집합의 크기가 L 인 SOFM 신경회로망의 학습시스템에서, k 번째 입력패턴에 대한 i 번째 출력뉴런의 값 V_i 는 j 번째 입력요소 x_j^k 와 연결세기 벡터 W_{ij} ($i=1, \dots, n_o, j=1, \dots, n_i$)와의 유클리드 거리에 따라 결정된다. 즉,

$$V_i(t) = \sum_j^{n_i} \|x_j^k(t) - W_{ij}(t)\| \quad (2)$$

이다. 따라서 전체 출력뉴런의 값을 결정하는데는 $t_o = n_o(2n_i - 1)\tau_A$ 의 시간이 필요하다. 이 때 τ_A 는 두 실수를 덧셈하는 시간이다. 전체 출력뉴런의 값이 결정되면, 승리뉴런이 결정되어야 한다. 승리뉴런은 출력뉴런 값중에서 가장 적은 값을 갖는 뉴런으로, 이를 결정하는데는 $t_w = (n_o - 1)\tau_B$ 의 수행시간이 요구된다. 이 때 τ_B 는 두 실수를 비교하는 시간이다. 마지막으로,

모든 연결세기들은 승리뉴런을 중심으로 다음과 같이 갱신된다. 즉,

$$\Delta W_{ij}(t+1) = \alpha (x_j^k(t) - W_{ij}(t)) \quad (3)$$

$$W_{ij}(t+1) = W_{ij}(t) + \Delta W_{ij}(t+1) \quad (4)$$

이다. 이 때 α 는 학습률을 나타낸다. 식 (3)에서는 $t_d = N(\tau_A + \tau_M)$ 의 시간이, 식 (4)에서는 $t_i = N\tau_A$ 의 시간이 필요하다. 여기서 τ_M 은 두 실수를 곱하는 시간이다. SOFM 신경회로망의 학습 시스템은 전술한 과정을 학습집합의 크기 L 만큼 계속하여 하나의 epoch를 끝낸다. 이러한 학습과정을 순서처리형 컴퓨터에서 처리하면 망의 크기 N 또는 학습집합의 크기 L 이 커짐에 따라 그 처리시간이 무척 길어진다. 따라서 병렬화 등 고속화 방안이 요구된다.

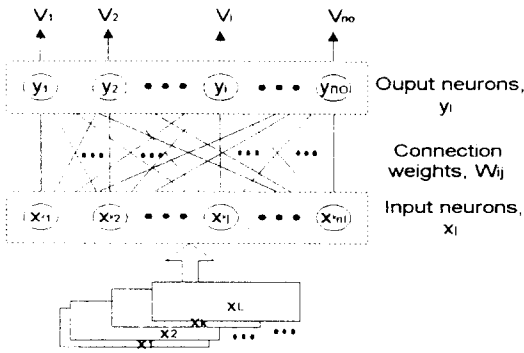


그림 1. 출력뉴런이 일차원적으로 구성된 SOFM 신경회로망

Fig. 1. SOFM neural network with one dimensional output neurons.

1) 망 분할에 의한 병렬처리

망 분할 방법은 그림 2와 같이 SOFM 신경회로망을 부회로망으로 분할하고, 각 부회로망의 뉴런과 그것에 연결된 연결세기들을 병렬처리 시스템의 각 프로세서에 할당하여 처리하는 분할방법이다. 따라서 p 개의 프로세서를 사용하면 크기 N 인 망은 $N_i = N/p, i = 1, 2, \dots, p$ 인 p 개의 부회로망으로 분할된다. 일반적인 SOFM 신경회로망의 입력뉴런 수는 출력뉴런 수보다 많이 적어서 프로세서에서 메모리를 별로 차지하지 않는다. 따라서 입력층은 분할하지 않고 모든 프로세서에 공통적으로 할당시키면, 입력뉴런간 통신시간을 줄일 수 있어 병렬처리 성능은 좀 더 높아진다¹⁶⁾. 따라서 출력뉴런의 값을 계산하는데는 t_o/p 의 시간이 걸린

다. 이 때, 각 프로세서에 할당된 뉴런들에서 국소적 승리뉴런(local winner)을 구한 다음, 각 승리뉴런 i 의 위치와 그것의 출력 V_i 가 프로세서간에 상호 통신되어야 전역적 승리뉴런(global winner)이 결정될 수 있다. 예를 들어, 링 구조인 경우에는 프로세서 수만큼의 single node gather의 통신이 요구된다. 이 경우 각 프로세서는 할당된 뉴런의 출력을 순서적으로 통신하며, 전역적 승리뉴런을 알아야만 학습을 할 수 있기 때문에, 다른 프로세서가 통신을 하는 시간에도 더 이상 다른 일을 진행시킬 수 없다. 이와 같이 p 개의 프로세서에 분산되어 있는 V_i 값으로부터 순서적으로 승리뉴런이 결정되면, 그 결과를 다시 single node broadcast를 통하여 전체 프로세서에 통신되어야 한다. 그런 후 각 프로세서는 자신에게 할당된 연결세기들을 학습하며, 이 때 $(t_d + t_i)/p$ 의 시간이 요구된다. 망 분할을 이용한 병렬학습에서는 매 패턴에 대하여 승리뉴런을 순서처리로 결정하기 때문에 속도제고율(speedup) 증가에 한계가 있다¹²⁾.

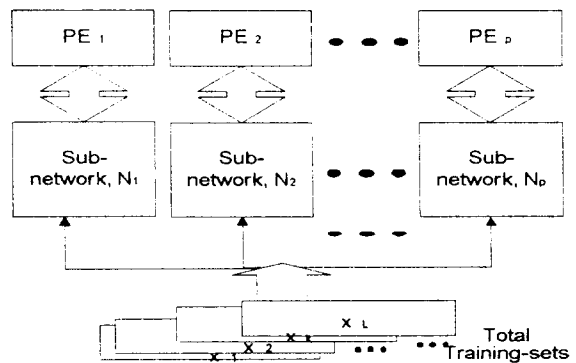


그림 2. 망 분할에 의한 병렬구현

Fig. 2. Parallel implementation using network decomposition.

2) 학습집합 분할에 의한 병렬처리

SOFM 신경회로망을 학습집합 분할에 의해 병렬구현할 경우에는 그림 3과 같이 L 개로 구성된 전체 학습집합을 p 개의 부학습집합으로 나누고, 이들을 p 개의 각 프로세서에 할당시켜 병렬학습하도록 한다. 이 때 망 분할과 달리 학습집합을 분할할 뿐 각 프로세서는 각각 크기 N 인 온전한 SOFM 신경회로망을 하나씩 할당받는다. 각 프로세서는 자기에게 할당된 L/p

개의 패턴으로 구성된 부학습집합에 대하여 학습을 수행한다. 따라서 m번째 프로세서에서의 학습은 다음과 같이 식 (3)과 (4)를 변형하여야 한다. 즉,

$$\Delta W_{ij}^m(t+1) = \alpha (x_j^m(t) - W_{ij}(t)) \quad (3')$$

$$W_{ij}(t+1) = W_{ij}(t) + \sum_{m=1}^p \Delta W_{ij}^m(t+1) \quad (4')$$

이다. 식 (4)'의 우변 두번째 항은 프로세서간 single node gather 후에 얻어지며, 모든 프로세서가 일치된 연결세기를 갖기 위해서는 얻어진 연결세기를 다시 single node broadcast 하여야 한다. 또한 전체 집합에 대한 학습을 위하여 궁극적으로는 할당받은 신경회로망의 모든 연결세기를 프로세서간에 상호 통신해야 하므로 통신부담이 가중된다. 그러나 망 분할에서 승리 뉴런을 결정할 때와 같은 순서적인 수행은 없다. 또한 학습집합 분할에서는 통신부담을 줄이기 위하여 각 프로세서는 ΔW_{ij} 를 매 패턴마다 통신하지 않고 update cycle 만큼 축적하였다가 통신을 한다. 본 논문에서는 update cycle이 L/p 인 표준적인 epoch update 방법을 이용하여 성능평가하였다.

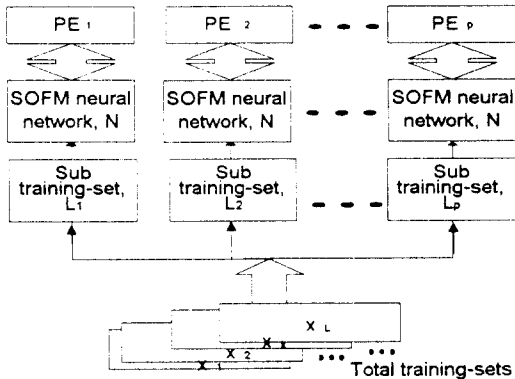


그림 3. 학습집합 분할에 의한 병렬구현
Fig. 3. Parallel implementation using training-set decomposition.

3) 병렬처리 성능해석

먼저, 순서처리형 컴퓨터에 의한 SOFM 신경회로망의 수행시간 T_s 는

$$T_s = e_s L [t_o + t_w + t_d + t_l] \quad (5)$$

이 된다. 여기서 e_s 는 반복수행 횟수를 나타내며, t_o 는 식 (2)를, t_w 는 승리뉴런을 구하는 시간을, t_d 는 식 (3)을, t_l 은 식 (4)를 각각 구하는 시간이다. 그리고

p 개의 프로세서를 사용하여 망 분할로 SOFM 신경회로망을 병렬구현하였을 때의 수행시간 T_n 은

$$T_n = e_n L \left[\frac{t_o}{p} + t_{wl} + \frac{(t_d + t_l)}{p} + t_{wg} + t_{cn} \right] \quad (6)$$

이 된다. 이 때 e_n 는 망 분할 시 반복횟수를 나타낸다. 식에서 t_{wl} 은 각 프로세서 내에서의 국소적 승리뉴런을 결정하기 위한 시간인 반면, t_{wg} 는 전체 프로세서로부터의 정보를 가지고 전역 승리뉴런을 결정하는데 걸리는 시간이다. 이 t_{wg} 는 그림 4에서와 같이 각 프로세서에서 순서적으로 계산되며, 나머지 프로세서들은 결정된 전역 승리뉴런에 대하여 통보될 때까지 기다리면 된다. 그리고 마지막 항 t_{cn} 은 각 프로세서에서 국소적 승리뉴런의 위치정보와 그 출력을 교신하는 시간과 전역 승리뉴런의 위치정보 전송을 위한 통신시간이다. 즉, 식 (6)에서 t_{wl} , t_{wg} 및 t_{cn} 은 각각

$$\begin{aligned} t_{wl} &= \left(\frac{n_o}{p} - 1 \right) \tau_B \\ t_{wg} &= (p-1) \tau_B \\ t_{cn} &= dn_t c + 6dt_c \end{aligned} \quad (7)$$

이 된다. t_c 는 실수값의 통신시간을, d 는 프로세서망 또는 컴퓨터망의 직경(diameter)을 각각 나타낸다. 링구조에서 $d=p-1$ 이고, single node broadcast와 single node gather일 경우에 d 번의 링크통과횟수가 요구된다. 승리뉴런은 하나의 프로세서에서 순서적으로 처리되므로 프로세서 수 p 만큼 통신이 되어야 하고, 따라서 t_{wg} 는 p 에 비례하여 증가한다. 또한 전체통신시간은 프로세서 수 및 학습 패턴의 수 L 에 비례하지만 망의 크기 N 과는 무관하다. 망 분할을 이용한 SOFM 신경회로망의 병렬구현에서는 프로세서의 수 p 가 어느정도 이상이 되면 통신시간이 계산시간보다 커져 전체 수행시간 T_n 이 증가한다.

그리고 망 분할을 링구조인 다중컴퓨터 망에서 구현하였을 때, 식 (6)으로부터 최대의 성능을 얻기 위한 프로세서의 수 \bar{p}_n 는 $\partial T_n / \partial p = 0$ 을 만족시키므로 다음과 같은 결과를 얻을 수 있다. 즉,

$$\bar{p}_n = c_n \sqrt{N} \quad (8)$$

이다. 여기에서 c_n 는 τ_A, τ_B, τ_M 및 t_c 의 함수이고, 이는 대상 다중컴퓨터망 시스템이 주어지면 고정되는 비례상수이다.

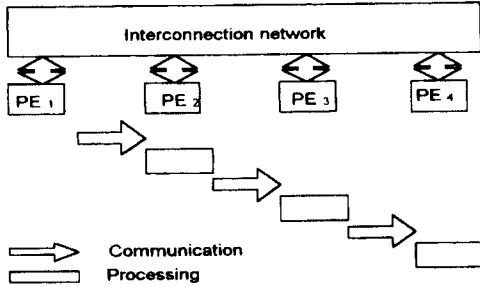


그림 4. SOFM 신경회로망의 학습시스템에서 전역적 승리뉴런을 얻기위한 통신
Fig. 4. Communication for the determination of the global winner.

다음으로 학습집합 분할에 의해 SOFM 신경회로망을 병렬구현 하였을 때의 수행시간 T_t 는

$$T_t = e_t \left[\frac{L}{p} (t_o + t_w + t_d) + \left(\frac{L}{p} - 1 \right) t_l + (p-1) t_l + t_{cl} \right] \quad (9)$$

이다. 이 때 e_t 는 학습집합 분할시의 반복 수행횟수이다. 두 번째 항은 국부연결세기의 합을 계산하기 위한 시간이고, 세번째항은 국부연결세기를 합하여 전역연결세기를 계산하는 시간이다. 그리고 t_{cl} 는 이들을 전송하기 위한 통신시간이다. 여기서 t_{cl} 는

$$t_{cl} = 2dNt_c + dn_t c \quad (10)$$

이 된다. 따라서 학습집합 분할시에는 전체가 병렬처리되어, 전체통신시간은 망의 크기 N 과 p 에만 비례하여 증가할 뿐 학습집합의 크기 L 과는 관계없다. 따라서 식 (9)로부터 링구조의 다중컴퓨터망에서 학습집합 분할에 의해 구현하였을 때, 최대의 성능을 얻기 위한 프로세서의 수 \bar{p}_i 는 $\partial T_t / \partial p = 0$ 을 만족시키므로 다음과 같은 결과를 얻을 수 있다. 즉,

$$\bar{p}_i = c_n \sqrt{L} \quad (11)$$

이다. 여기에서 c_i 는 다중컴퓨터 시스템에 따른 비례상수이다.

III. 다중컴퓨터망에서의 병렬 구현

전술한 SOFM신경회로망의 병렬성능을 그림 5와 같이 환형구조의 Transputer 시스템^[10]을 대상으로 성능평가하고 병렬 구현하여 비교 평가하였다. 병렬언어

인 OCCAM을 사용하였으며, master와 slave로 구성된 processor farm모델을 이용하였다. master는 PC로부터의 필요한 데이터를 slave프로세서에 전달하고, slave는 학습을 전달한다. 따라서 Transputer 시스템에서 SOFM신경회로망을 병렬학습한다면 식 (7)과 (10)에서의 통신시간 t_c 는

$$t_c = \beta + \gamma b \quad (12)$$

으로 모델링되며, 여기서 β 는 통신을 위한 초기화 시간이고, γ 는 b바이트의 통신시간이다. 32 비트 프로세서인 T414 프로세서를 갖는 트랜스퓨터 시스템에서는 $\beta=8.4 \mu s$, $\gamma=1.15 \mu s$ 로 측정되었다.

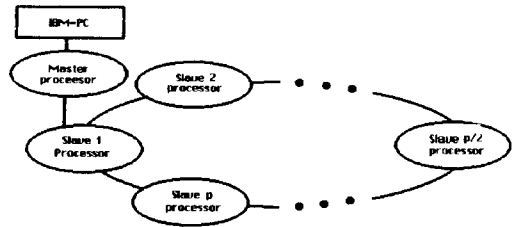


그림 5. 링 구조의 트랜스퓨터
Fig. 5. Ring connected Transputer system.

또한 c_n 과 c_i 는 각각 1.11 및 1.71로 계산되었으며, 식 (8) 및 (11)로부터 최대의 성능을 얻기 위한 프로세서의 수 $\bar{p}_i = \bar{p}_n$ 인 해석적인 경계는 다음과 같다. 즉,

$$L = 0.423N \quad (13)$$

이다. 식에서 최대의 성능을 얻기 위한 프로세서 수에서 두 분할 방법의 성능을 판단할수 있는 경계지수는 0.423이다. 즉, 주어진 SOFM 신경회로망의 L 이 N 의 0.423배보다 크다면, 최대의 성능을 얻을 수 있는 프로세서의 수는 망 분할 방법보다 학습집합 분할 방법에서 더 많아서 확장성이 크다. 그 반대인 경우에는 망 분할에서 더 많다.

표 1. 망 분할을 이용하였을 때의 수행시간과 학습집합 분할을 이용하였을 때의 수행시간이 같아질 때의 학습패턴 수 L

Table 1. L when the processing time in network decomposition equals that in training-set decomposition.

p	2	4	8	16	...	∞
L	0.5927N	0.4962N	0.4570N	0.4395N	...	0.4232N

그리고 식 (6)과 (9)로부터 두 분할 방법에 따른 처리 시간을 비교하여, 그 경계지수를 구하면 표 1과 같다. 이 때, n_i 는 2로 하였다. 프로세서의 수 p 가 증가할수록 경계의 값은 줄어들며, $p \rightarrow \infty$ 일 때 0.4232이다.

IV. 실험 및 결과 고찰

그림 5와 같이 Transputer 시스템을 구성하고 두 분할 방법에 따라 SOFM 신경회로망을 각각 병렬 구현하였다. 실험을 위한 학습패턴은 32비트 길이를 갖는 2 차원상의 임의적인 실수로 하였으며, slave 프로세서의 수를 변화시키면서 두 분할방법에 따른 학습시간을 조사하였다.

표 2는 망 분할방법에 대한 수행시간과 속도제고율을 나타낸 것이다. 이 때 A는 성능평가에 의한 이론값을, B는 구현하였을 때의 수행시간이다. Transputer 시스템에서 구현하였을 때의 수행시간에는 부가된 모든 시간이 포함되어 있다. 망 분할에서는 학습집합의 크기보다 망의 크기가 큰 경우에 좋은 성능을 가져 해석적인 결과와 같다. 표 3은 학습집합 분할에서의 수행시간과 속도제고율을 나타낸 것이다. 학습집합 분할에서도 해석적인 결과 마찬가지로 망의 크기보다 학습집합의 크기가 큰 경우에 좋은 성능을 가졌다. 표 2와 3 으로부터 망의 크기 N 이 학습집합의 크기 L 보다 2배 큰 문제인 경우, 2개의 프로세서에서는 망 분할에서 더 좋은 성능을 가졌지만, 그 이상의 프로세서에서는 학습 집합 분할에서 더 좋은 성능을 가졌다. 이것은 표 1에서 $p=2$ 일 때의 경계가 $L=0.5927N$ 이고, $p=4$ 일 때의 그것은 $L=0.4962N$ 이기 때문이다.

표 2. 망 분할 병렬구현하였을 때의 수행시간 T_n 과 속도제고율 Sp

Table 2. The processing time, T_n , and speedup, Sp in the network decomposition.

p		N=512L=64	N=256L=64	N=128L=64	N=128L=128	N=128L=256
1	A	3.52	1.76	0.88	1.76	3.52
	B	4.78	2.40	1.20	2.40	4.79
2	A	1.77(1.98)	0.89(1.98)	0.44(1.98)	0.89(1.99)	1.76(1.99)
	B	2.92(1.63)	1.46(1.64)	0.73(1.64)	1.46(1.64)	2.91(1.64)
4	A	0.94(3.76)	0.47(3.76)	0.23(3.74)	0.46(3.86)	0.90(3.91)
	B	1.52(3.13)	0.76(3.14)	0.38(3.15)	0.74(3.22)	1.47(3.25)
8	A	0.57(6.16)	0.29(6.14)	0.14(6.10)	0.25(6.87)	0.48(7.34)
	B	0.88(5.42)	0.44(5.42)	0.22(5.40)	0.40(5.92)	0.77(6.22)

표 3. 학습집합 분할 병렬구현하였을 때의 수행시간 T_s 과 속도제고율 Sp

Table 3. The processing time, T_s , and speedup, Sp in the training-set decomposition.

p		N=512L=64	N=256L=64	N=128L=64	N=128L=128	N=128L=256
1	A	3.52	1.76	0.88	1.76	3.52
	B	4.78	2.40	1.20	2.40	4.79
2	A	1.77(1.98)	0.89(1.98)	0.44(1.98)	0.89(1.99)	1.76(1.99)
	B	2.92(1.63)	1.46(1.64)	0.73(1.64)	1.46(1.64)	2.91(1.64)
4	A	0.94(3.76)	0.47(3.76)	0.23(3.74)	0.46(3.86)	0.90(3.91)
	B	1.52(3.13)	0.76(3.14)	0.38(3.15)	0.74(3.22)	1.47(3.25)
8	A	0.57(6.16)	0.29(6.14)	0.14(6.10)	0.25(6.87)	0.48(7.34)
	B	0.88(5.42)	0.44(5.42)	0.22(5.40)	0.40(5.92)	0.77(6.22)

식 (3)과 (4)에 따르는 망 분할된 학습 시스템은 수렴이 보장되지만, 식 (3)'와 (4)'에 의한 학습집합 분할된 학습 시스템은 식 (3)'에서 학습률 α 가 커지면 각 프로세서에서 계산되어진 연결세기의 크기가 커지고, 따라서 식 (4)'의 연결세기는 더욱 커지게 되어수렴을 보장할 수 없다. 그림 6은 학습패턴 수 $L=128$ 일 때, 학습률 α 에 따른 수렴을 알아보기 위하여 반복횟수에 대한 학습오차를 나타낸 것이다. 학습패턴수 L 에 따라 수렴이 되기 위하여는 효과적인 학습률 α 를 선정하여야 하는데, 이 값이 너무 크면, 갱신된 연결세기의 값이 커져 발산을 할 수 있다. 그림에서 만약 학습률 α 가 0.03 이상의 값으로 주어지면, SOFM 신경회로망은 더 이상 학습이 되지 않고 발산을 한다. 또한 학습률 α 값이 0.001로 너무 적게 주어진다면 수렴은 보장되지만 학습오차를 줄이지 못하게 되어 효과적이지 못하다. 따라서 가장 효과적인 $\alpha=0.02$ 의 학습률을 가지고 학습을 하여야만 학습오차를 빠르고 최소로 줄일 수 있다.

그림 7은 학습 패턴 수 L 에 따른 학습률 α 를 조사한 결과이다. $L=64$ 및 256일 때, α 는 각각 0.06, 0.015로 선정되어야만 효과적인 학습이 이루어진다. 그리고 L 의 크기가 커질수록 α 를 적은 값으로 주어야 한다.

그림 8은 L 에 따른 그림 7의 α 를 가지고 학습하였을 때의 평균적인 학습오차를 나타낸 것이다. 그림에서 학습패턴 수 L 이 증가하더라도 α 가 적당하게 주어진다면 학습오차는 반복횟수 e_i 가 늘어나지 않아도 줄어

들 수 있다.

V. 결 론

본 논문에서는 SOFM신경회로망의 학습시스템을 그 네트워크 병렬성 및 학습집합 병렬성에 따라 분할하여 각각을 다중컴퓨터망에서 병렬구현하고, 신경회로망의 크기 N , 학습집합의 크기 L 및 프로세서의 수 p 의 증가에 따른 시스템의 병렬처리 성능변화를 조사하였다. 링구조에서 해석적 성능평가 결과, 망 분할방법은 학습 집합의 크기보다는 신경회로망의 크기에, 학습집합 분할방법은 신경회로망의 크기보다는 학습집합 크기에 더 의존하여 성능이 변화함을 확인하였다. Transputer 시스템에서 병렬구현하여 실험한 결과도 이에 일치하였다. Transputer시스템의 경우, 망 분할에서 최고의 속도제고율을 갖는 프로세서의 수는 $1.11\sqrt{N}$ 이었고, 학습집합 분할에서의 그것은 $1.71\sqrt{L}$ 이었다. 학습집합의 크기가 망의 크기보다 2배 이상 크면, 학습집합 분할을 이용하여 SOFM 신경회로망을 병렬구현하는 것이 망 분할을 이용하는 것보다 더 나은 성능을 얻을 수 있다. 그리고 학습패턴의 수 $L=64, 128$ 및 256 을 가진 SOFM 신경회로망을 학습집합 분할에 의해 구현하고자 할 때, 수렴이 보장되면서도 효과적인 학습이 되기 위한 학습률 α 는 각각 $0.06, 0.02$ 및 0.015 이었다. 따라서 SOFM 신경회로망을 이용한 응용문제를 고속처리하기 위해 사용한 다중컴퓨터 시스템이 주어지면 최대성능을 얻기위한 병렬처리 시스템의 설계와 구현에 효과적으로 이용될 수 있다.

참 고 문 헌

- [1] T. Kohonen, *Self-Organization and Associative Memory*, Third edition, Springer-Verlag Berlin, Heidelberg, 1989.
- [2] R. Togneri, "Parallel implementation of the Kohonen algorithm on Transputer," *IJC-NN*, Singapore, vol. 2, pp. 1717-1722, 1991.
- [3] H. Kaiyo et al, "A parallel neurocomputer architecture towards billion connection updates per second," *IJCNN*, Washington, vol. 2, pp. 47-50, 1990.
- [4] R. Mann and S. Haykin, "A parallel implementation of Kohonen feature maps on the Warp systolic computer," *IJCNN*, Washington, vol 2, pp. 84-87, 1990.

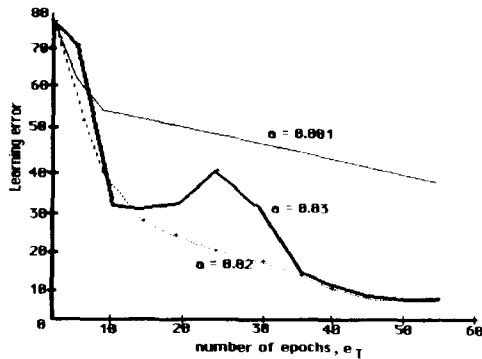


그림 6. 학습률에 따른 수렴오차
Fig. 6. Convergence error vs. learning rate.

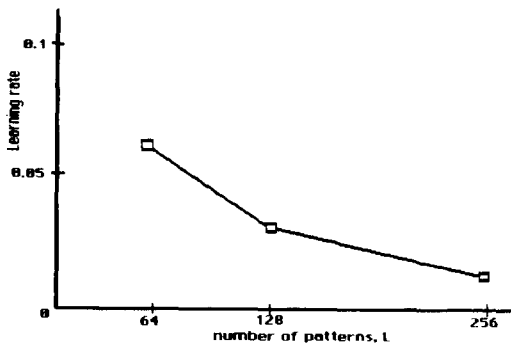


그림 7. 학습패턴 수 L이 변할 때, 학습을 위한 학습률
Fig. 7. Variation of the learning rate for convergence when the number of patterns, L increases.

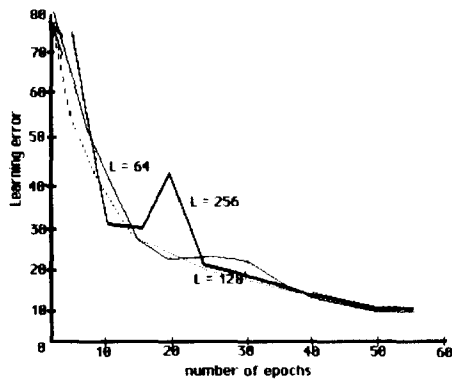


그림 8. 학습패턴 수 L에 따른 평균 수렴오차
Fig. 8. Average convergence error vs. the number of patterns, L.

- [5] W. Pan, P. K. Sharpe, "A simulation of MLP exploiting training-set parallelism," *IJCNN*, Beijing, vol. 1, pp. 183-188, Nov. 1992.
- [6] H. Kaito, H. Yoshizawa, H. Iciki, and K. Asakawa, "Parallelizing the self-organizing feature map," *IJCNN*, Washington, vol. 2, pp. 283-286, 1990.
- [7] Kai Hang, *Advanced Computer Architecture : Parallelism, Scalability, Programmability*, McGraw-Hill, Singapore, 1993.
- [8] Jacob M. J. Murre, "Transputers and neural networks : An analysis of implementation constraints and performance," *IEEE Transaction on Neural Networks*, vol. 4, no. 2, Mar., 1993.
- [9] H. Paugam and Moisy, "Parallel neural computing based on network duplicating," *Parallel Algorithms for Digital Image Processing, Computer Vision and Neural Network*, John Wiley & Sons, Singapore, pp. 305-340, 1993.
- [10] R. S. Cok, *Parallel Programs for the Transputer*, Prentice Hall, 1991.

 저 자 소 개



金 宣 鍾(正會員)

1989년 2월 경북대학교 전자공학과 졸업(학사). 1991년 2월 경북대학교 대학원 전자공학과 졸업(석사). 1996년 8월 경북대학교 대학원 전자공학과 졸업(박사). 1995년 2월 ~ 현재 순천공업전

문대학 전임강사. 주관심분야는 병렬처리, 신경회로망, 패턴인식 등임

崔 興 文(正會員) 第 33卷 B編 第 2 號 參照

현재 경북대학교 전자전기공학부 교수