

다중컴퓨터망에서 신경회로망 설계를 위한 고속병렬처리시스템의 구현

(An Implementation of High-Speed Parallel Processing System for Neural Network Design by Using the Multicomputer Network)

金 鎮 浩*, 崔 興 文**

(Jin Ho Kim and Heung Moon Choi)

要 約

본 논문에서는 다중컴퓨터망에서 신경회로망 설계를 위해 높은 선형고속화 확장성을 갖는 고속 병렬 처리시스템을 구현하였다. 동기부담 및 통신부담을 줄여 선형고속화 확장성을 높였으며, 또한 신경회로망의 병렬화 방법에 따른 병렬처리 모델들과 그 성능평가모델들도 마련하였다. 제안한 시스템을 숫자 및 문자의 기억학습을 위한 신경회로망에 적용하여 본 결과, 기존의 연구 결과들 보다 선형 고속화 확장성이 좀 더 높았으며 제안한 성능평가모델의 타당성도 확인할 수 있었다.

Abstract

In this paper, an implementation of high-speed parallel processing system for neural network design on the multicomputer network is presented. Linear speedup expandability is increased by reducing the synchronization penalty and the communication overhead. Also, we presented the parallel processing models and their performance evaluation models for each of the parallization methods of the neural network. The results of the experiments for the character recognition of the neural network based on the proposed system show that the proposed approach has the higher linear speedup expandability than the other systems. The proposed parallel processing models and the performance evaluation models could be used effectively for the design and the performance estimation of the neural network on the multicomputer network.

* 正會員, 慶北産業大學校 電子工學科
(Dept. of Elec. Eng., Kyungpook Sanup Univ.)

**正會員, 慶北産業大學校 電子工學科
(Dept of Elec, Kyungpook Nat'l Univ.)

(*이 논문은 1991년도 교육부지원 한국학술진흥재단의 자유공모(지방대학육성) 과제 학술연구조성비에 의하여 연구되었음.)

接受日字: 1992年 9月 14日

I. 서론

최근 신경회로망을 이용한 패턴인식, 영상처리, 자연어처리 및 최적화 등에 관한 연구들이 활발히 진행되고 있다.^[1-12] 그러나 신경회로망의 가변시냅스 연결세기(synapse interconnection strength)를 하드웨어로 구현하기가 어렵고, 특히 확률론적 신경회로망의 경우 온도 스케줄링과정 등을 신경회로망의 하드웨어에 반영시키는 것이 매우 힘든 실정이다.^[7-8] 따라서, 대부분의 신경회로망 관련연구들에 있어서 실제 하드웨어 구현을 위한 기본 연구로서 컴퓨터 시뮬레이션도 필수적으로 요구되고 있다.^[10]

신경회로망 시뮬레이션은 매우 계산집약적(computation intensive)이고 많은 양의 컴퓨터 자원(computer resource)들을 필요로 한다. 따라서 현재의 순서처리형 컴퓨터로는 비교적 작은 규모의 신경회로망의 설계 및 시뮬레이션도 어려운 경우가 많고, 특히 수천개의 뉴런들이 상호 연결된 대규모 신경회로망의 설계 및 시뮬레이션은 거의 불가능한 경우가 많다.^[1-3]

순서처리형 컴퓨터의 처리능력 한계로 인한 신경회로망 관련연구의 제약을 극복하기 위해 신경회로망의 설계 및 응용을 위한 고속병렬처리시스템^[14-16]에 관한 연구들이 활발히 진행되고 있다.

기존의 신경회로망 병렬처리시스템에서는 신경회로망의 규모에 따라 노드컴퓨터의 수가 증가할 경우 속도제고율(speedup)이 오히려 감소되는 성능포화 현상에 대한 명확한 분석 및 대비를 명확하게 밝히고 있지 않을 뿐만 아니라 다양한 규모와 여러 종류의 신경회로망들을 각각 효과적으로 시뮬레이터에 할당하는 방법도 명확하게 제시되지 않고 있는 실정이다. 따라서 대규모 신경회로망의 고속 시뮬레이션 및 그 설계와 응용에서 신경회로망 병렬처리시스템이 최대의 선형 속도제고율 증가와 확장성을 갖도록 하는 방안을 마련할 필요가 있으며, 그 성능평가 모델 및 방안 등 신경회로망의 해석과 설계를 위한 도구가 절실히 요구되고 있다.

본 논문에서는 다중컴퓨터망에서 신경회로망 설계를 위해 높은 선형 고속화 확장성을 갖는 고속 병렬처리시스템을 구현하였다. 동기부담 및 통신부담을 줄여 선형고속화 확장성을 높였으며, 또한 신경회로망의 각 병렬화 방법에 따른 병렬처리 모델들과 이를 이용한 성능평가 방안들도 마련 하였다. 실제 구현한 신경회로망 병렬처리 시스템을 숫자 및 영문자의 기억학습문제에 적용하여 실험하고 그 성능을 기존의 연구결과와 비교검토 하였다.

II. 신경회로망의 병렬처리모델

본 논문에서는 먼저 역전파와 같은 학습용 신경회로망의 신호전송 모델을 마련하고 이를 이용하여 병렬처리 모델을 제안하였다. 특히 네트워크분할 병렬처리 모델과 패턴분할 병렬처리 모델을 각각 제안함으로써 신경회로망의 구조적특성 및 학습패턴의 크기와 개수에 따라 선택적으로 이용할 수 있도록 하였다.

1. 역전파 신경회로망의 신호전송모델

역전파 신경회로망의 신호전송과정은 입력층에서 출력층으로의 순방향 신호전송과 출력층에서 입력층으로의 역방향 신호전송으로 구분된다. 3 층구조의 역전파 신경회로망을 그림 1에 도시하였다.

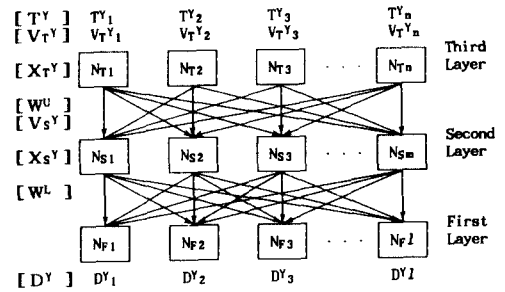


그림 1. 3 층 구조 역전파 신경회로망의 예
Fig. 1. An example of a three-layer back-propagation neural network.

그림에서 첫째, 둘째 및 셋째 층은 각각 입력층, 은닉층 및 출력층이다. 그리고 N_{Ti} , $i=1,2,\dots,l$ 는 첫째 층의 i 번째 뉴런이고 N_{Sj} , $j=1,2,\dots,m$ 는 둘째 층의 j 번째 뉴런이며 N_{Tk} , $k=1,2,\dots,n$ 는 셋째 층의 k 번째 뉴런이다. D^Y 는 Y 번째 입력패턴이고, T^Y 는 Y 번째 입력패턴에 대한 목적출력(desired output)이다. 그리고 X^S 및 X^T 는 각각 Y 번째 패턴에 대한 둘째 층 및 셋째 층 뉴런들의 입력신호이고, V^S 및 V^T 는 각각의 출력이다. 또한 W^L 및 W^U 는 각각 하위층 및 상위층의 시냅스 가중행렬들이다.

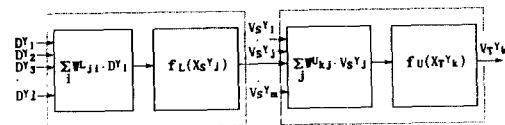


그림 2. 역전파 신경회로망의 순방향 신호전송모델
Fig. 2. Forward signal flow model for the backpropagation neural network.

역전파와 같은 인식용 신경회로망에서 V_s^Y 및 V_T^Y 의 계산과정을 전달함수 f_L 및 f_U 로 표현하면 뉴런의 순방향 신호 전송과정을 그림 2와 같이 모델링 할 수 있다.

그림 2에서 f_L 및 f_U 는 각각 역전파 신경회로망의 경우 S형(sigmoid)함수가 되지만 임의의 학습용 신경회로망 일 경우 해당 신경회로망의 전달함수가 된다.

한편 역전파의 가중치 갱신방법은 패턴간 독립학습과 패턴간 종속학습 등 두가지 방법으로 구분된다. 두 경우의 가중치 갱신과정을 그림 3과 같이 하나의 역방향 신호전송과정으로 모델링 할 수 있다.

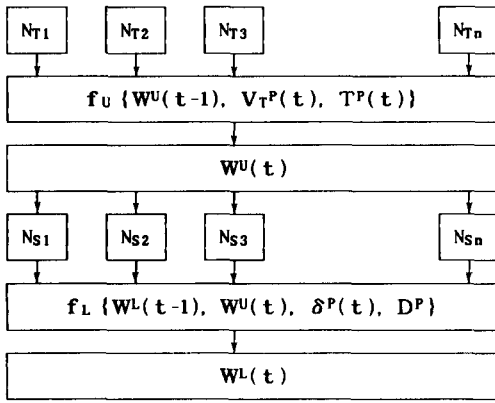


그림 3. 역전파 신경회로망의 역방향 신호전송 모델
Fig. 3. Backward signal flow model for the backpropagation neural network.

그림에서는 둘째 층 및 셋째 층의 가중치 갱신과정을 각각 가중치 갱신함수 f_L 및 f_U 로 표현하였다. 따라서 패턴종속학습의 경우에는 각 패턴들이 제시될 때 마다 가중치 갱신함수 f_U 에 따라 패턴들간의 의존성에 따라 학습시키면 되고, 패턴독립학습의 경우에는 각 패턴들에 대해 독립적으로 가중치 갱신함수 f_U 에 따라 국소 가중치를 계산한 다음 이들을 합하면 된다. 패턴간 종속학습에서는 신경망을 뉴런간 병렬성에 따라 분할하고, 이때 입력패턴들은 패턴내의 병렬성(intrapattern parallelism)에 따라 분할하면 된다. 한편 패턴독립 학습의 경우에는 학습패턴 집합내에 패턴간 병렬성(interpattern parallelism)을 이용하여 패턴들을 분할하면 된다. 본 논문에서는 인식용 신경회로망을 뉴런간 병렬성에 따른 네트워크 분할방식 및 패턴간 병렬성에 따른 패턴분할방식의 두 가지로 병렬 구현하였다.

2. 네트워크 분할 병렬처리모델

역전파의 네트워크 분할 병렬구현을 위해 그림 2 및 3으로 표현되는 각 신호전송모델에서 그 고유의 뉴런간 공간적 병렬성(spatial parallelism) 및 시간적 병렬성(temporal parallelism)을 추출한 다음 그 병렬성에 따라 신경회로망을 병렬분할하고 병렬처리 모델을 제안 하였다. 즉, 신경회로망의 뉴런간 공간적 병렬성에 따라 뉴런 및 시냅스 가중행렬을 병렬분할하고, 시간적 병렬성을 반영시켜 이를 좀 더 세부 모듈들로 분할하였다.

Y 번째 패턴의 둘째 층의 뉴런들에 대한 입력 X_S^Y 및 그 출력들 V_S^Y 를 다음과 같이 분할하여 병렬로 계산한다.

$$\sum_{j=0}^{m-1} X_{S^p_j} = \sum_{j=0}^{q-1} X_{S^p_j} + \sum_{j=q}^{2xq-1} X_{S^p_j} + \dots + \sum_{j=(n^*-1)xq}^{n^*xq-1} X_{S^p_j} \quad (1)$$

$$\sum_{j=0}^{m-1} V_{S^y_j} = \sum_{j=0}^{q-1} V_{S^y_j} + \sum_{j=q}^{2xq-1} V_{S^y_j} + \dots + \sum_{j=(n^*-1)xq}^{n^*xq-1} V_{S^y_j} \quad (2)$$

둘째 층 뉴런들의 순방향 신호전송과정을 식 (1) 및 (2)와 같이 분할하면 각 뉴런들의 입력신호 X_S^Y 및 출력값 V_S^Y 를 계산하기 위해 다른 영역에 속한 하위층 뉴런들의 출력을 전송받아야 한다. 따라서 하위층 뉴런의 데이터 의존성이 존재한다. 한편, Y 번째 패턴에 대한 셋째 층의 뉴런들에 대한 입력신호 X_T^Y 및 그 출력들 V_T^Y 도 다음과 같이 분할하여 병렬로 계산 할 수 있다. 즉,

$$\sum_{k=0}^{n^*-1} X_{T^y_k} = \sum_{k=0}^{r-1} X_{T^y_k} + \sum_{k=r}^{2xr-1} X_{T^y_k} + \dots + \sum_{k=(n^*-1)xr}^{n^*xr-1} X_{T^y_k} \quad (3)$$

$$\sum_{k=0}^{n^*-1} V_{T^y_k} = \sum_{k=0}^{r-1} V_{T^y_k} + \sum_{k=r}^{2xr-1} V_{T^y_k} + \dots + \sum_{k=(n^*-1)xr}^{n^*xr-1} V_{T^y_k} \quad (4)$$

이다. 셋째 층의 뉴런들의 순방향 신호전송과정을 식 (3) 및 (4)와 같이 병렬분할 했을 경우에도 한 영역의 뉴런들은 다른 영역에 속한 둘째 층 뉴런들의 출력 V_S^Y 에 대한 데이터 의존성을 갖는다.

한편 상반부 가중치는 다음과 같이 병렬로 갱신할 수 있다.

$$\sum_{k=0}^{n^*-1} \sum_{j=1}^m W_{kj}^u(t) = \sum_{k=0}^{r-1} \sum_{j=1}^m W_{kj}^u(t) + \sum_{k=r}^{2xr-1} \sum_{j=1}^m W_{kj}^u(t) + \dots + \sum_{k=(n^*-1)xr}^{n^*xr-1} \sum_{j=1}^m W_{kj}^u(t) \quad (5)$$

$$W^U(t) = f_U\{W^U(t-1), V^Y(t), T^Y(t)\} \quad (6)$$

식 (5)와 같이 상반부 가중치를 갱신 할 때는 다른 영역의 뉴런들과 데이터 의존성이 존재하지 않는다. 하반부 가중치도 다음과 같이 병렬로 갱신할 수 있다.

$$\begin{aligned} \bigcup_{j=0}^{n-1} w_{ji}^L(t) &= \bigcup_{j=0}^{q-1} w_{ji}^L(t) + \bigcup_{j=r}^{2q-1} w_{ji}^L(t) + \\ &\dots + \bigcup_{j=(n'-1)sq}^{n'sq-1} w_{ji}^L(t) \end{aligned} \quad (7)$$

$$W^L(t) = f_i \{ W^L(t-1), W^U(t), \delta^y(t), D^y \} \quad (8)$$

식 (7)과 같은 하반부 가중치에서는 상위층의 오차값 및 가중치들이 이용되므로 각 영역간의 데이터의존 관계가 존재 한다.

영역간의 데이터의존 관계에 따라 통신이 필요한 부분을 의미한다. 각 통신모듈들을 더욱 세분화하여 전산처리 및 통신과정을 중규모의 부태스크로 분할 시킴으로써 노드간의 국소적 동기의 파이프라인 처리가 가능하도록 하였다. 따라서 다중컴퓨터망의 동기부담 및 통신부담을 가급적 줄일 수 있도록 하였다.

3. 패턴분할 병렬처리모델

역전파의 패턴간 병렬성을 수평축에 반영하고 시간적 병렬성을 각 패턴들에 대한 순차처리과정인 수직축에 반영하여 패턴분할 병렬처리모델을 작성하였다. 학습패턴들을 p개의 그룹으로 분할하여 각 부태스크들에 할당하면, 각 부태스크들은 할당받은 패턴들에 대해 t 회째 국소오차값 $E^p(t)$ 및 국소가중치들 $W^{ph}(t)$, $W^{lh}(t)$ 를 계산한다. 계산된 국소오차값 및 국소가중치들을 상호통신하여 전역가중치 $W^{GU}(t)$, $W^{GL}(t)$ 들을 계산하여 다음 t+1 번째 갱신에서의 시냅스 가중행렬로 한다. 따라서 패턴분할 병렬학습에서는 국소가중치 및 국소오차값들의 상호통신이 필요하게 된다.

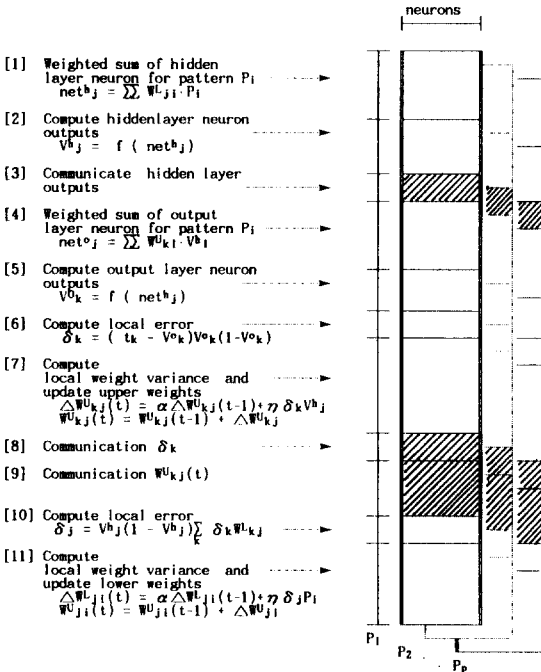


그림 4. 3층 역전파 신경회로망의 네트워크 분할 병렬학습에 대한 병렬처리모델

Fig. 4. Parallel processing model for the network-decomposed parallel learning of 3-layer backpropagation neural network.

전술한 방식에 따라 역전파를 p 개의 영역으로 네트워크 분할 병렬처리할 경우의 병렬처리모델을 그림 4에 도시하였다. 그림에서 빗금친 부분은 전술한 각

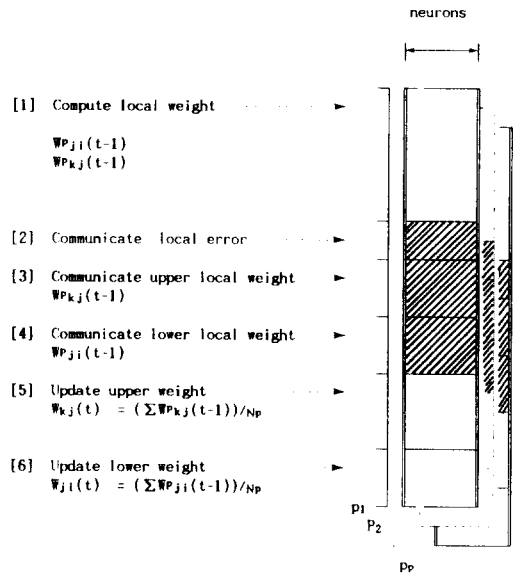


그림 5. 3층 역전파 신경회로망의 패턴분할 병렬학습에 대한 병렬처리모델

Fig. 5. Parallel processing model for the pattern-decomposed parallel learning of 3-layer backpropagation neural network.

패턴분할 병렬학습의 병렬처리 모델을 도시하면 그

림 5와 같다. 각 노드간의 데이터 통신과정은 시간적 병렬성에 따라 더욱 세분화 하여 국소적 동기가 가능하도록 하였다. 패턴분할 병렬학습방법을 이용하면 네트워크 분할과는 달리 각 노드에 신경회로망의 순차 학습알고리즘이 할당되므로 임의의 구조의 신경회로망을 쉽게 병렬구현 할 수 있다.

Ⅲ. 성능평가모델

N 개의 뉴런을 갖는 신경회로망을 p 개의 부태스크로 네트워크 병렬 분할하여 다중컴퓨터망에 할당했을 경우의 성능평가 모델을 그림 6 에 도시하였다.

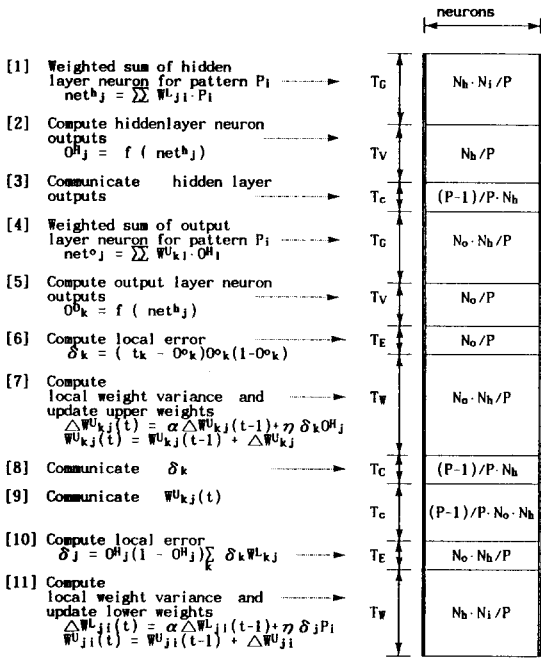


그림 6. 역전파를 p 개의 노드에서 네트워크 분할 병렬처리할 경우의 성능평가 모델

Fig. 6. Performance evaluation model for the backpropagation algorithm which is network-decomposed on the p nodes.

그림 6 에서 T_G , T_V 및 T_C 는 각각 단위 가중합계산, 각 뉴런의 출력계산 및 각 뉴런의 출력전송에 사용되는 시간이고, T_E 및 T_W 는 각각 한 뉴런의 오차값계산 및 하나의 가중치값을 갱신하는 데 소요되는 시간이다.

역전파 신경회로망 알고리즘을 하나의 순서처리형

컴퓨터에서 처리 할 때 한번의 가중치갱신에 소요되는 시간 T_{sing} 은 다음과 같다.

$$T_{sing} = N^2 (2T_G + 2T_W + T_E) + N (2T_V + T_E) \quad (9)$$

식에서 N 은 각 층별 N_i , N_h 및 N_o 를 의미한다. 한편 역전파를 네트워크 분할병렬구현 했을 때 한번의 가중치갱신을 하는데 소요되는 처리시간 T_{PEI} 는

$$T_{PEI} = \frac{N^2}{P}(2T_G + 2T_W + T_E) + \frac{N}{P}(2T_V + T_E) + \frac{P-1}{P}(2N + N^2)T_C \quad (10)$$

가 된다. 따라서, 역전파 신경회로망 알고리즘을 P개의 노드컴퓨터로 병렬처리할 때 그 고속화율 S_p 는

$$S_p(N, P) = \frac{P}{1 + \frac{Ca}{Pa}} \quad (11)$$

이 된다. 여기서 Pa 및 Ca 는 각각

$$Pa = N^2(2T_G + 2T_W + T_E) + N(T_V + T_E) \quad (12)$$

$$Ca = (p-1)(2N + N^2)T_C \quad (13)$$

이 된다. 따라서 N이 ∞가 될 때와 P가 ∞가 될 때 S_p 는 각각 다음과 같이 된다. 즉,

$$\lim_{N \rightarrow \infty} S_p(N, P) \cong P \frac{1}{1 + \frac{2(p-1)T_C}{2T_G + 2T_W + T_E}} \quad (14)$$

$$\lim_{P \rightarrow \infty} S_p(N, P) \cong \frac{P_s}{(2N^2 + 2N)T_C} \quad (15)$$

이다. 식에서 $2(p-1)T_C < 2T_G + 2T_W + T_E$ 일때는 $S_p \cong P$ 가 되지만 그렇지 않을 경우에는 $S_p < P$ 가 된다. 따라서, 네트워크분할 병렬 처리의 경우 뉴런 개수 N 을 증가시켜도 S_p 는 선형적으로 증가 되지 않는 경우도 생기게 된다. 또한 뉴런 개수 N을 고정시키고 노드개수 P를 증가시키면 S_p 는 N^2 에 반비례함을 알 수 있다. 따라서, 네트워크분할 병렬처리의 경우 식 (14) 및 (15)를 이용하면 최대의 고속화율을 얻을 수 있는 뉴런개수 및 노드수를 설정할 수 있다.

패턴분할의 경우에도 동일한 방법으로 고속화율을 구하고 패턴 개수 S를 ∞로 하면 그 고속화율은 다시

다음과 같이 된다.

$$\lim_{P \rightarrow \infty} S_p \approx P \frac{L_s}{L_p} \quad (16)$$

식에서 L_s 및 L_p 는 각각 순서처리 할 때의 학습횟수 및 패턴분할 병렬 처리에서의 반복학습 횟수이다. 가중치 갱신방법에 따라 L_s 와 L_p 는 서로 다를 수가 있다. 즉, $L_p \gg L_s$ 가 될 수도 있으므로 패턴 분할 병렬학습에서는 가중치 갱신방법에 따라 고속화율이 크게 변할 수 있다. 만일 $L_p = L_s$ 라 가정한다면 패턴의 갯수 S 가 증가 될 수록 $P\alpha \gg C\alpha$ 가 되어 $S_p \approx P$ 에 가깝게 된다.

한편 네트워크분할 병렬처리에서의 병렬화 부담 (parallelization overhead) O_n 과 패턴분할 병렬처리에서의 병렬화 부담 O_p 의 비 $O\alpha$ (N, P, S)는 다음과 같다.

$$O\alpha(N \cdot P, S) = \frac{O_n}{O_p} = \frac{\left\{ \frac{P-1}{P} \cdot (2N + N^2) \cdot T_c \right\} \cdot S}{\left\{ (P-1)(N \cdot T_E + 2N^2 \cdot T_C) + 2N^2 \cdot T_w \right\}} \quad (17)$$

따라서, 뉴런의 갯수 N , 노드갯수 P 및 패턴의 갯수 S 에 따라 O 는 달라질수 있으며 $O\alpha < 1$ 일때는 네트워크 분할 병렬처리가 유리하고 $O\alpha > 1$ 일때는 패턴분할 병렬처리가 유리함을 판단할 수 있다. 그러므로 식 (17)에 따라 두 방법중 유리한 방법을 선택할 수 있다.

IV. 응용 예 및 결과 고찰

역전파알고리즘을 갖는 신경회로망을 네트워크분할 및 패턴분할 방식으로 각각 구현하여 숫자 및 영어 알파벳의 연상기억 학습에 적용실험하고 그 결과를 타 시스템과 비교 검토하였다. 먼저 그림 7과 같이 트랜스퓨터(Transputer) 시스템을 환형구조로 접속한 다음 occam 언어를 이용하여 제안된 병렬처리 모델들을 구현하였다.

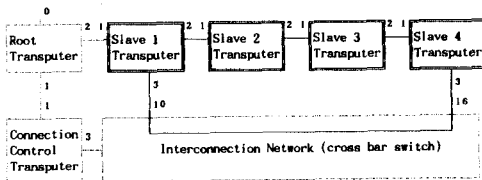


그림 7. 환형구조로 접속된 트랜스퓨터시스템
Fig. 7. Transputer system with ring interconnection.

표 1과 같이 4 가지의 신경회로망들을 트랜스퓨터 시스템에 네트워크분할 병렬구현하고 숫자 및 영문자들을 기억학습시켰다. 입력 패턴들은 32비트 및 64비트로 구성되며 숫자는 0 에서 9 까지의 10개 종류 그리고 영문자는 A 에서 Z 까지의 26 개 종류가 이용되었다. 실험에서는 숫자 또는 영문자 전체를 기억시킨 후 한 비트의 오차도 없이 원래 패턴을 모두 복구할 수 있을 때까지 신경회로망을 학습 시켰다.

표 1. 실험에 이용된 신경회로망들의 구조
Table 1. Architecture of the neural networks used for the experiments.

Data	10 Numeric characters		26 English alphabets	
	32-bit	64-bit	32-bit	64-bit
N_i	32	64	32	64
N_h	16	16	16	16
N_o	32	64	32	64
N_s	16, 384	65, 536	16, 384	65, 536

N_s : the number of synaptic interconnections

표 2 는 트랜스퓨터 상에 네트워크분할방식으로 병렬 구현된 역전파 알고리즘을 이용하여 표 1에서 제시된 패턴들을 각각 기억학습시킬 때의 고속화율과 병렬처리효율을 도사한 것이다. 표 2를 보면 성능평가모델 식 (14)에서 예측한 바와 같이 역전파를 네트워크분할 병렬구현 했을 경우, 뉴런의 갯수 N 이 커지더라도 고속화율이 선형적으로 증가되지 않음을 알 수 있다.

표 2. 네트워크분할 방식의 역전파 신경회로망을 이용하여 기억학습시킬 때의 고속화율과 효율

Table 2. Speedup and efficiency of the network-decomposed backpropagation algorithm.

Data	[Sp/Err]			
	10 Numeric characters		26 English alphabets	
	32-bit	64-bit	32-bit	64-bit
1	1.0/100	1.0/100	1.0/100	1.0/100
2	1.9/95	1.9/95	1.9/95	1.9/95
4	3.6/90	3.6/90	3.6/90	3.6/90
8	6.4/80	6.4/80	6.4/80	6.4/80
16	10.2/64	10.4/65	10.1/64	10.4/65

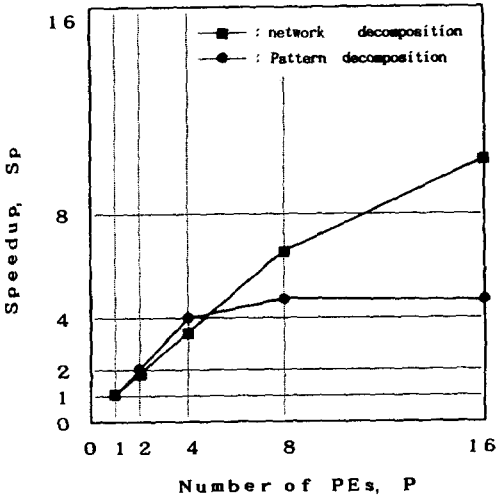


그림 8. 네트워크분할 및 패턴분할에 의한 병렬학습에 있어서 각각의 노드수 증가에 대한 고속화율의 변화

Fig. 8. Speedup change of the network-decomposed and the pattern decomposed parallel learnings as a function of the node increment.

한편 동일한 입력 패턴에 대해서 네트워크분할 및 패턴분할에 의한 병렬학습에 있어서 각각의 노드수 증가에 대한 고속화율의 변화를 그림 8에 도시하였다.

그림에서 노드수가 적을 경우에는 패턴분할 병렬처리방법이 네트워크 분할 병렬처리에 비해 좀 더 높은 고속화율의 증가를 얻을 수 있지만 노드수가 계속 증가되어 갈수록 네트워크분할 병렬처리방법이 좀 더 유리함을 알 수 있다.

이는 네트워크분할 병렬처리에서는 식 (11)과 같이 고속화율이 패턴의 갯수와는 무관하게 정의되지만 패턴분할 병렬처리에서는 식 (16)에서 보논바와 같이 한 노드당 패턴의 갯수가 상당히 많을 경우 그 고속화율이 노드 갯수에 비례하여 증가하기 때문이다. 따라서, 뉴런의 갯수가 많을수록 네트워크 분할 병렬처리가 유리하고, 뉴런의 갯수가 적고 패턴의 갯수가 많을 수록 패턴분할 병렬처리가 유리함을 확인할 수 있다. 그림 9는 그림 8에서 도시한 네트워크분할 병렬처리에서의 학습속도와 기존 발표된 타 시스템들^[3]의 학습속도를 비교한 것이다. 그림 9에서 제안된 시스템이 대규모의 신경회로망이나 소규모의 신경회로망에서 공히 Intel iPSC/2 Hypercube 시스템에

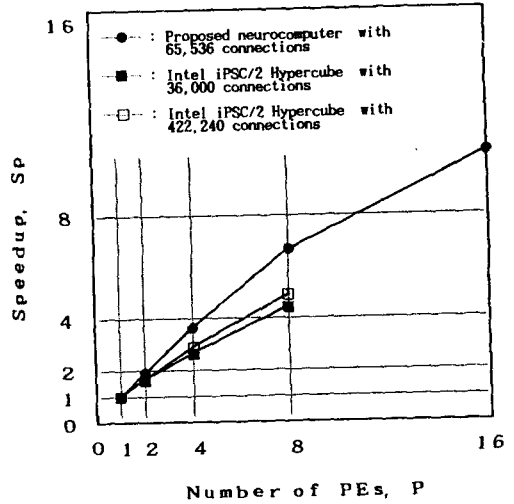


그림 9. 역전파 학습에 대한 고속화율 Sp의 비교
Fig. 9. Speedup comparison of neurocomputers for the backpropagation neural networks.

비해 좀 더 높은 고속화율을 갖는다는 것을 확인할 수 있다. 이는 신경회로망의 구조적 특징인 다중노드 방송(multinode broadcast or multicast) 능력에서는 제안된 시스템이나 iPSC/2 가 비슷하지만 제안된 시스템의 경우 처리와 통신을 가능한 시간중첩(time overlapping) 시키도록 함으로써 통신부담을 최소화한 결과로 볼 수 있다. 비록 제한된 하드웨어 환경하에서 비교적 작은 규모의 신경회로망에 대해서 실험 하였지만 현재의 실험결과와 성능평가모델로 해석해 볼 때, 제안된 시스템은 대규모의 노드를 갖는 신경컴퓨터로 확장시킬 때도 좀 더 선형고속 확장이 가능함을 확인할 수 있다.

V. 결론

본 논문에서는 다중컴퓨터망에서 신경회로망 설계를 위해 높은 선형 고속화 확장성을 갖는 고속 병렬처리시스템을 구현하였다. 동기부담 및 통신부담을 줄여 선형 고속화 확장성을 높였으며, 또한 신경회로망의 병렬화 방법에 따른 병렬처리 모델들과 각각의 성능평가 모델들도 마련하였다. 설계 구현한 신경회로망 병렬처리 시스템을 문자 기억학습문제에 적용해보고 그 성능을 기존의 연구결과와 비교 검토하였다.

제안한 시스템을 숫자 및 문자의 기억학습에 적용하여 본 결과 기존의 연구 결과들 보다 선형 고속화

확장성이 좀 더 높았으며 제한한 성능평가모델의 타당성도 확인되었다. 특히 패턴의 갯수를 $N_p = 26$ 으로 고정시켰을 때, 뉴런수 대 노드수가 $N/P > 10$ 정도일 경우에는 네트워크 분할 병렬처리가 유리하고, $N/P < 10$ 정도일 경우에는 패턴분할 병렬처리가 유리함도 확인 할 수 있었다. 따라서 본 방식을 이용하면 다중컴퓨터망의 노드 갯수 및 노드의 메모리를 증가시킬 경우에 선형 고속화율의 증가하에서도 대규모의 신경회로망을 쉽게 병렬구현 할 수 있으리라 기대된다.

參 考 文 獻

- [1] A. Singer, "Exploiting the inherent parallelism of artificial neural networks to achieve 100 million interconnects per second," *International Neural Network Conference*, Paris, France, vol. 2, pp. 656-660, July 1990.
- [2] D. Ercoskun and K. Oflazer, "Experiments with parallel backpropagation on a hypercube parallel processor system," *Artificial Neural Networks*, North-Holland, vol. 2, pp. 1465-1468, 1991.
- [3] J. Ghosh and K. Hwang, "Mapping neural networks onto message passing multicomputers," *Journal of PDC*, vol. 6, pp. 291-330, April 1989.
- [4] J. Oglesby and J. Mason, "Dynamic scheduling for feedforward neural nets using Transputer," *International Conference on Artificial Neural Network*, vol. 3, pp. 257-260, Oct. 1989.
- [5] J. Mure, "Transputer implementations of neural networks: an analysis," *Artificial Neural Networks*, North-Holland, vol. 2, pp. 1537-1540, 1991.
- [6] K. Parker, "Parallelized backpropagation training and it's effectiveness," *International Joint Conference on Neural Networks*, Washington D.C., vol. 2, pp. 179-182, Jan. 1990.
- [7] M. Miksa, "A development tool for neural networks simulations on Transputers," *Parallel Processing in Neural Systems and Computers*, North-Holland, pp. 295-298, 1990.
- [8] M. Migliore, G. Ayala, and S. Fornili, "Modeling of neural systems on Transputer networks," *Parallel Processing in Neural Systems and Computers*, North-Holland, pp. 291-294, 1990.
- [9] N. Bagherzadeh, T. Kerola, B. Leddy, and R. Brice, "On parallel execution of the traveling salesman problem on a neural network model," *International Joint Conference on Neural Networks*, San Diego, vol. 3, pp. 317-324, June 1987.
- [10] H. Ernst, B. Mokry and Z. Schreter, "A Transputer based general simulator for connectionist models," *Parallel Processing in Neural Systems and Computers*, North-Holland, pp. 283-286, 1990.
- [11] N. Fukuda, Y. Fujimoto, and T. Akabane, "A Transputer implementation of toroidal lattice architecture for parallel neurocomputing," *International Joint Conference on Neural Networks*, Washington D.C., vol. 2, pp. 43-46, Jan. 1990.
- [12] P. Koikkalainen and E. Oja, "Specification and implementation environment for neural networks using communicating sequential processes," *International Joint Conference on Neural Networks*, San Diego, California, vol. 1, pp. 533-540, July 1988.
- [13] D. Ercoskun and K. Oflazer, "Experiments parallel backpropagation on a hypercube parallel processing system," *Artificial Neural Networks*, North-Holland, pp. 1465-1468, 1991.
- [14] A. Shaw, "Deterministic timing schema for parallel programs," *5th International Parallel Processing Symposium*, California, pp. 56-63, May 1991.
- [15] A. Basu, S. Srinivas, K. Kumar and

A. Paularaj, "Performance analysis of algorithms on a message passing multiprocessor," *5th International Parallel Processing Symposium*, California, pp. 43-50, May 1991.

[16] C. A. R. Hoare, "Communicating sequential processes," *Communications of ACM*, vol. 21, pp. 666-677, Aug. 1978.

著者紹介



金 鎮 浩(正會員)

1960年 6月 20日生. 1985年 8月 경북대학교 공과대학 전자공학과 졸업. 1988년 2월 경북대학교 대학원 전자공학과 공학석사학위 취득. 1992년 2월 경북대학교 대학원 전자공학과 공학박사학위 취득. 1992

年 3월 ~ 1993년 4월 현재 경북산업대학교 공과대학 전자공학과 전임강사. 주관심분야는 병렬처리, 신경회로망 및 컴퓨터그래픽 등임.



崔 興 文(正會員)

1944年 5月 17日生. 1966年 2월 인하대학교 공과대학 전기공학과(전자전공) 졸업. 1974년 2월 인하대학교 대학원 전기공학과(전자전공) 공학석사학위 취득. 1987년 8월 인

하대학교 대학원 전자공학과 공학박사학위 취득. 1982년-1983년 뉴욕주립대학교 객원교수. 1974년 11월 ~ 1993년 4월 현재 경북대학교 전자공학과 교수. 주관심분야는 논리회로설계, 병렬처리, 신경회로망 및 그 응용 등임.