

MLP 신경망을 위한 시공간 병렬처리모델

김성완*

A Spatiotemporal Parallel Processing Model for the MLP Neural Network

Sung-Oan Kim *

요약

본 논문에서는 MLP 신경망의 패턴 학습과정을 위하여 시공간 병렬성을 고려한 병렬처리모델을 제시한다. 시간 병렬성을 위한 학습집합 분할과 공간 병렬성을 위한 네트워크 분할을 동시 적용하여 융통성있는 병렬처리모델을 설계하고자 하였다. 성능평가모델로부터 해석적으로 구한 결과, 대규모 과제라고 해도 패턴 크기와 패턴 갯수 중 어느 쪽이 지배적이냐에 따라 분할병렬처리 방법이 절충되어야 할 것으로 본다.

Abstract

A parallel processing model by considering a spatiotemporal parallelism is presented for the training procedure of the MLP neural network. We tried to design the flexible parallel processing model by simultaneously applying both of the training-set decomposition for a temporal parallelism and the network decomposition for a spatial parallelism. The analytical performance evaluation model shows that when the problem size is extremely large, the speedup of each implementation depends, in the extreme, on whether the problem size is pattern-size intensive or pattern-quantity intensive.

▶ Keyword : MLP neural network, Spatiotemporal parallelism, Parallel processing model, Performance evaluation

• 제1저자 : 김성완
• 접수일 : 2005.10.14, 심사완료일 : 2005.10.26
* 수원과학대학 인터넷정보과 조교수

I. 서론

패턴의 인식이나 복구 능력이 우수하여 널리 적용되고 있는 MLP(Multi-layer Perceptron) 신경망은 대규모의 패턴에 대해서는 그 학습에 지나치게 많은 시간이 요구되는 문제가 있다. 또한, 기존의 순서처리형 컴퓨터는 패턴인식이나 최적화 등과 같은 대규모의 과제에 적용되는 신경망의 내재된 병렬성을 효과적으로 구현할 수 없을 뿐만 아니라 만족할 만한 해를 구하는데 대단히 긴 시간을 필요로 한다. 이러한 문제점을 극복하기 위해 여러 가지 신경망을 병렬구현할 수 있는 융통성과 대규모 시스템으로의 확장성을 가진 다중프로세서나 다중컴퓨터시스템 등을 이용하여 병렬구현하기 위한 많은 연구가 발표된 바 있다 [1-4].

MLP 신경망의 병렬처리는 뉴런들간의 공간 병렬성을 이용하는 네트워크 분할(network decomposition) 병렬처리 [1-2]와 입력패턴들간의 시간 병렬성을 이용하는 학습집합 분할(training-set decomposition) 병렬처리 [3-4]로 분류된다. 네트워크 분할 병렬처리는 신경망의 입력층, 은닉층 및 출력층 뉴런들을 균등하게 분할시키는 병렬 학습방법이고, 학습집합 분할 병렬처리는 전체 입력패턴을 균등하게 학습부집합들로 분할시키는 병렬 학습방법이다. 네트워크 분할과 학습집합 분할은 각각 단일방식으로 적용되어 많은 병렬처리시스템에서 구현되었다. 그러나 학습과정의 시공간 병렬성에 비추어 학습집합 분할과 네트워크 분할을 동시 적용한다면 좀더 융통성있는 병렬구현이 될 수 있을 것이다.

본 논문에서는 MLP 신경망의 패턴 학습과정을 위하여 시공간 병렬성을 고려한 병렬처리모델을 제시하였다. 시간 병렬성을 위한 학습집합 분할과 공간 병렬성을 위한 네트워크 분할을 동시 적용하여 융통성있는 병렬처리모델을 설계하고자 하였다. 또한 성능평가모델을 통하여 과제의 크기에 따라 병렬처리시스템의 학습속도를 높일 수 있는 방안도 해석적으로 모색하였다. 본 논문의 구성은 다음과 같다. II장에서는 관련 연구로부터 정리된 MLP 신경망과 병렬처리에 대하여 소개하고, III장에서는 MLP 신경망을 위한 시공간 병렬처리모델의 설계에 대하여 설명한다. 그리고 IV장에서는 해석적 성능평가를 통한 시공간 병렬처리모델의 타당성에 대하여 검토하고, V장에서는 마지막으로 본 논문에 대하여 결론을 맺는다.

II. MLP 신경망과 병렬처리

2.1 MLP 신경망

MLP 신경망은 입력층(input layer), 은닉층(hidden layer) 및 출력층(output layer)의 다층 구조를 가지며 Backpropagation 알고리즘(5)을 이용하여 효과적으로 학습시킬 수 있다. (그림 1)은 삼층 구조를 가지는 MLP 신경망을 예시한 것이다.

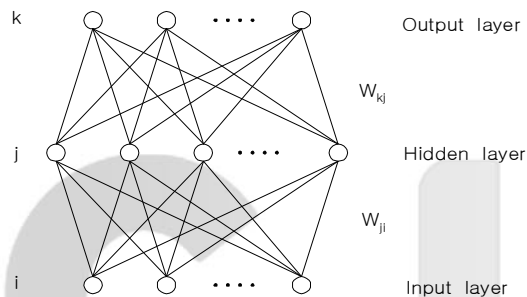


그림 1. 삼층 구조의 MLP 신경망
Fig. 1 MLP neural network with three layers

입력의 입력패턴 P_i 에 대해서 은닉층의 j 번째 뉴런과 출력층의 k 번째 뉴런의 입력값 net_j 와 출력값 O 는 다음과 같이 각각 나타낼 수 있다. 즉,

$$net_j = \sum_{i=0}^{N_i-1} W_{ji} \cdot O_i \quad (0 \leq j \leq N_h - 1) \dots\dots\dots (1)$$

$$O_j = f(net_j - \theta_j) = \frac{1}{1 + EXP(-(net_j - \theta_j))} \dots\dots (2)$$

$$net_k = \sum_{j=0}^{N_h-1} W_{kj} \cdot O_j \quad (0 \leq k \leq N_o - 1) \dots\dots\dots (3)$$

$$O_k = f(net_k - \theta_k) = \frac{1}{1 + EXP(-(net_k - \theta_k))} \dots\dots (4)$$

이 된다. (식 1, 3)에서 N_i , N_h 및 N_o 는 각각 입력층, 은닉층 및 출력층의 뉴런 수를 나타내고, (식 2, 4)에서 θ_j 와 θ_k 는 은닉층의 j 번째 뉴런과 출력층의 k 번째 뉴런의 바이어스값을, $f(\cdot)$ 는 활성화함수를 나타낸다.

출력층에서는 입력패턴 P_i 로부터 계산된 실제 출력값 (actual output) O_k 와 목적 출력값 (target output) T_k 사이의 차로부터 오차값 E_k 를

$$E_k = \frac{(T_k - O_k)^2}{2} \dots\dots\dots (5)$$

로 구할 수 있으며, 이렇게 구하여진 각 뉴런의 오차값으로부터 층 사이의 뉴런들간의 가중치의 변화분을 다음과 같이 계산할 수 있다. 즉, 출력층과 은닉층 사이의 가중치의 변화분 ΔW_{kj} 및 은닉층과 입력층 사이의 가중치의 변화분 ΔW_{ji} 는

$$\delta_k = O_k \cdot (1 - O_k) \cdot (T_k - O_k) \dots\dots\dots (6)$$

$$\Delta W_{kj}(t + 1) = \eta \cdot \delta_k \cdot O_j + \alpha \cdot \Delta W_{kj}(t) \dots\dots\dots (7)$$

$$\delta_j = O_j \cdot (1 - O_j) \sum_{k=0}^{N_o-1} W_{kj} \cdot \delta_k \dots\dots\dots (8)$$

$$\Delta W_{ji}(t + 1) = \eta \cdot \delta_j \cdot O_i + \alpha \cdot \Delta W_{ji}(t) \dots\dots\dots (9)$$

로 구해질 수 있다. 여기서 t 는 반복 학습횟수, η 는 학습률 그리고 α 는 모멘트값을 나타낸다. 전체 입력패턴에 대한 전역오차값(global error) E_t 는 다음과 같이 각 패턴 P 에 대해 계산된 오차값 E^P_k 의 합으로 주어진다.

$$E_t = \sum_{P=0}^{N_p-1} \sum_{k=0}^{N_o-1} E^P_k \dots\dots\dots (10)$$

여기서 N_p 는 입력패턴의 수를 나타낸다.

2.2 병렬처리

신경망의 병렬처리에는 학습집합 분할 병렬처리와 네트워크 분할 병렬처리가 있으며, (그림 2)는 이러한 두 가지의 병렬처리를 예시한 것이다. (그림 2(a))는 학습시킬 신

경망을 나타낸 것이고, (그림 2(b))는 학습집합 분할 병렬 처리로서 전체 입력패턴을 균등하게 학습부집합들로 분할하며 분할된 각각을 동일한 신경망이 할당된 각각의 노드에 나누어 병렬학습시키는 방법이다. 각 노드는 국소오차값과 국소가중치를 계산한 후 전역오차값과 전역가중치를 얻기 위해 다른 노드들과 통신한다. 이렇게 계산된 전역가중치를 새로운 가중치로 갱신해서 다시 국소가중치를 계산하게 된다. 이러한 과정을 전역오차값이 원하는 범위 이내로 될 때까지 반복하게 된다. (그림 2(c))는 네트워크 분할 병렬 처리로서 입력층, 은닉층 및 출력층을 균등하게 수직분할하며 각 노드가 그 분할된 각 층의 뉴런의 일부분을 책임지고 순서적으로 입력되는 각 패턴별로 병렬학습시키는 방법이다. 여기서 만약 전체 패턴 수가 많아질 경우에는 많은 노드간의 통신이 필요하게 된다. 그리고 (그림 2(d))는 이러한 통신의 양을 줄이기 위한 변형된 네트워크 분할 병렬처리[2]이다. 여기서는 각 노드가 입력층과 출력층의 뉴런들을 모두 가지게 하고 은닉층에서만 수직분할 함으로써 통신의 양을 감소시킨다.

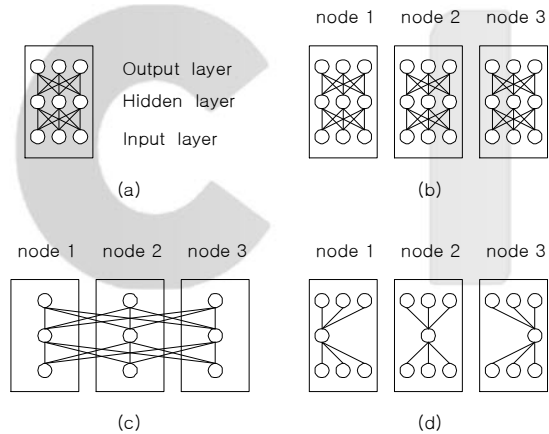


그림 2. 3개 노드를 이용한 신경망의 병렬처리
Fig. 2 Parallel processing of neural network using three nodes

본 논문에서는 (그림 3)에 예시된 바와 같이 전체적으로는 학습집합 분할을 적용함으로써 학습시킬 패턴 수가 많은 경우에도 전체 학습속도를 높이고, 분할된 각 패턴에 대해서는 개선된 네트워크 분할을 적용함으로써 하나의 패턴을 학습시키는 시간도 줄일 수 있도록 하였다.

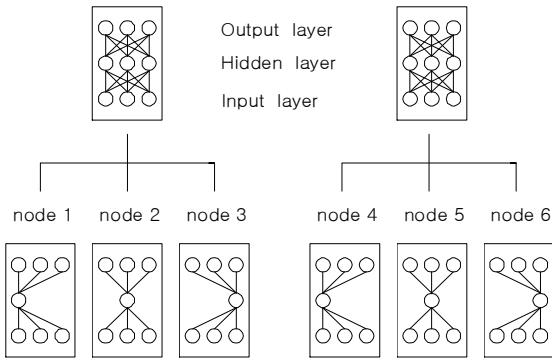


그림 3. 학습집합 분할과 네트워크 분할을 동시에 적용한 병렬처리
Fig. 3 Parallel processing by simultaneously applying both of the training-set decomposition and the network decomposition

III. 시공간 병렬처리모델

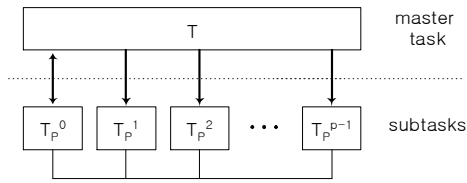
MLP 신경망을 병렬로 처리하기 위한 모델은 분할된 부태스크의 제어 및 데이터의 종속관계를 표시해주는 태스크 그래프로 나타낼 수 있다. (그림 4(a))는 한 개의 주태스크 T 와 p 개의 부태스크 T_p^i 를 갖는 시공간 병렬처리모델의 태스크 그래프이다. 주태스크는 각 부태스크로 학습시킬 패턴을 전송하고, 학습과정을 감시 (monitoring)하는 한편 부태스크들로부터 최종 학습결과인 가중치를 전송받는다. (그림 4(b))와 같이 구성되는 각 부태스크는 주태스크로부터 균등하게 분할된 학습패턴들을 받아서 네트워크 분할 병렬처리를 적용하여 가중치의 변화분을 구한 다음, 학습집합 분할 병렬처리로 국소가중치 W^U_t, W^L_t 를 각각 계산하여 다른 부태스크들과 통신한다. 이때 전역오차값 E_t 와 상위층 전역가중치 W^{GU}_t 및 하위층 전역가중치 W^{GL}_t 는 각각 다음과 같이 구할 수 있다.

$$E_t = \sum_{i=1}^q E_i + \sum_{i=q+1}^{2 \cdot q} E_i + \dots + \sum_{i=(p-1) \cdot q+1}^{p \cdot q} E_i \dots \dots \dots (11)$$

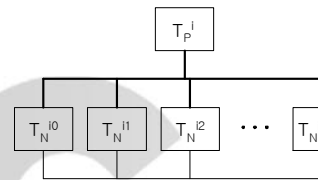
$$W^{GU}_t = \frac{\sum_{i=1}^q W^U_i + \sum_{i=q+1}^{2 \cdot q} W^U_i + \dots + \sum_{i=(p-1) \cdot q+1}^{p \cdot q} W^U_i}{p} (12)$$

$$W^{GL}_t = \frac{\sum_{i=1}^q W^L_i + \sum_{i=q+1}^{2 \cdot q} W^L_i + \dots + \sum_{i=(p-1) \cdot q+1}^{p \cdot q} W^L_i}{p} (13)$$

여기서, p 는 전체 부태스크의 수를, q 는 i 번째의 부태스크에 할당된 입력패턴의 수를 나타낸다. (식 12, 13)에서 전역가중치는 각 부태스크에서 전송된 국소가중치를 계속 더할 경우 발산할 수도 있으므로, 전체 부태스크 갯수만큼 나누어 값으로 계산한다.



(a)



(b)

그림 4. 시공간 병렬처리의 태스크 그래프
Fig. 4 Task graph of spatiotemporal parallel processing

패턴간의 병렬성을 수평축에 반영하고 각 패턴들에 대한 순차처리 과정의 시간적 병렬성을 수직축에 반영하여 MLP 신경망을 위한 학습집합 분할과 네트워크 분할을 동시에 적용한 병렬처리의 데이터존 그래프는 (그림 5)와 같다. 그림에서 전체 학습패턴들은 p 개의 그룹으로 분할되어 각 부태스크들로 입력되고, 각 부태스크들은 할당받은 패턴들에 대해 네트워크 분할을 적용하여 t 회째 국소오차값 $E^{U}_{TN^p}(t)$ 및 국소가중치 $W^U_{TN^p}(t), W^L_{TN^p}(t)$ 를 계산한다. 계산된 국소오차값 및 국소가중치를 상호 통신하여 전역가중치 $W^{GU}_{TN^p}(t), W^{GL}_{TN^p}(t)$ 를 계산하여 이를 다음 $t+1$ 번째 갱신에서의 가중치로 한다. 궁극적으로 학습은 국소오차값 및 국소가중치로부터 전역오차값 및 전역가중치를 구하여 전역오차값이 원하는 범위 이내의 값으로 될 때까지 전역가중치를 조정해가는 과정으로 볼 수 있다.

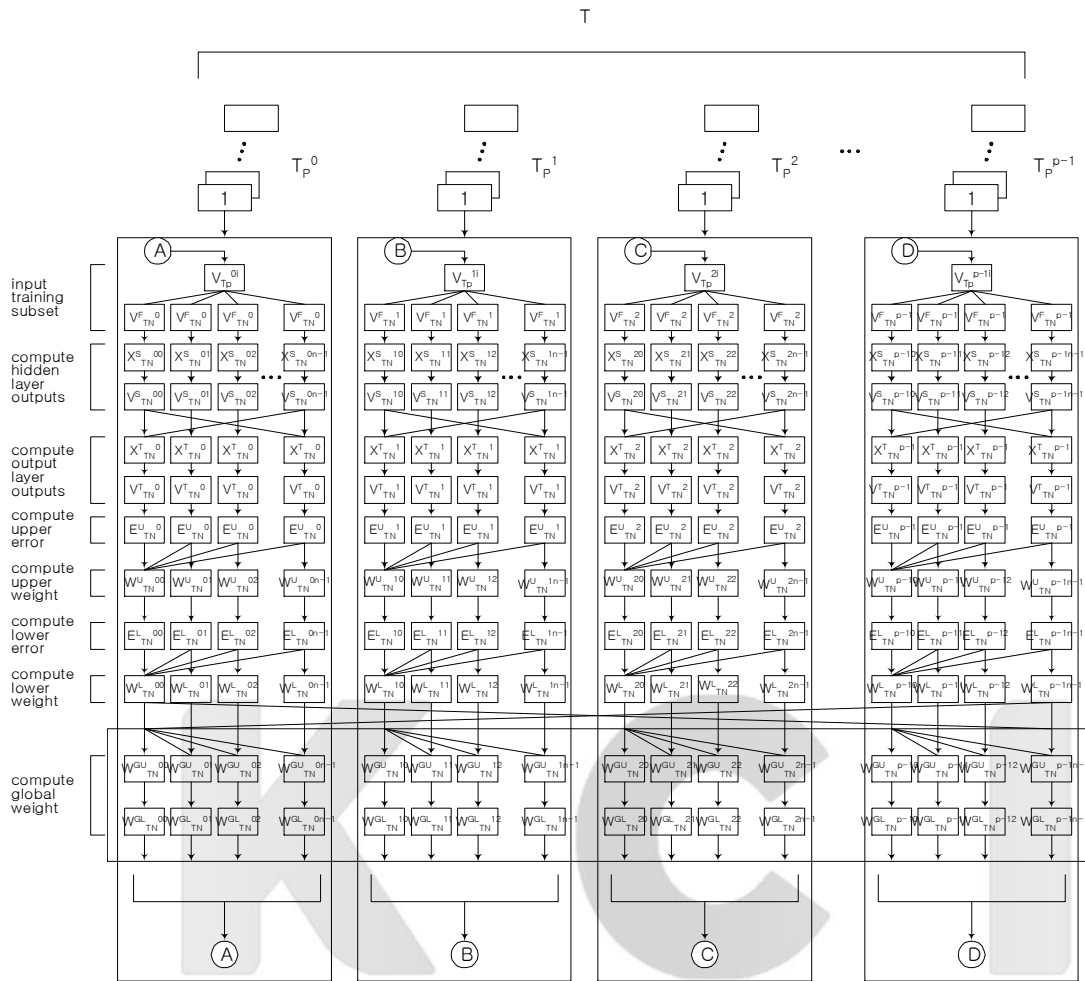


그림 5. 시공간 병렬처리의 데이터존 그래프
Fig. 5 Data dependence graph of spatiotemporal parallel processing

(main-ring) 구조를 나타낸 것이다. 한편 국소 네트워크 분할 병렬처리를 수행하는 노드컴퓨터들은 그림의 각 노드 컴퓨터와 부환형(sub-ring) 구조로 접속된다. 주처리컴퓨터와 노드컴퓨터들 사이 또는 노드컴퓨터들 상호간의 통신은 인접한 노드 컴퓨터를 통해 이루어질 수 있도록 하였다.

IV. 성능평가

본 논문에서는 시공간 병렬처리모델의 성능평가를 위하여 (그림 6)과 같이 병렬처리시스템을 구성하였다. 그림은 주처리컴퓨터 PE_H 와 p 개의 노드컴퓨터 PE_i 가 전역 학습 집합 분할 병렬처리를 위해 선형적으로 접속된 환형

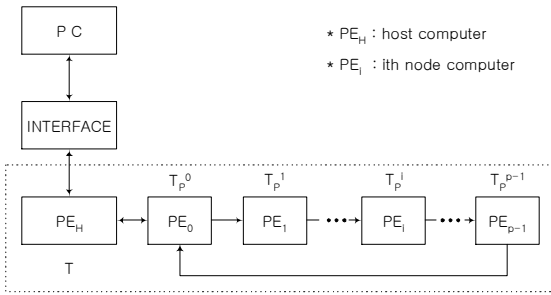


그림 6. 성능평가를 위한 병렬처리시스템의 구조
Fig. 6 Parallel processing system for performance evaluation

전술한 방법과 같이 MLP 신경망에 학습집합 분할과 네트워크 분할을 동시 적용하여 병렬처리할 경우의 시공간도표를 도시하면 (그림 7)과 같다. 그림에서 N_i , N_h 및 N_o 는 각각 입력층, 은닉층 및 출력층의 뉴런 갯수를, T_G 는 덧셈 및 곱셈에 소요되는 시간을, T_v 는 sigmoid 함수를 계산하는 시간을, 그리고 T_c 는 뉴런 한 개에 대한 가중치를 인접한 노드컴퓨터로 전송하는데 걸리는 시간을 나타낸다. 여기

서 수평축은 하나의 부태스크내에서 사용된 노드컴퓨터의 수를 나타내고, 수직축은 각 노드컴퓨터에 제시된 q 개의 입력패턴에 대한 한 번의 학습시간을 나타낸다.

전체 입력패턴의 수 N_p 를 사용된 전체 노드컴퓨터 수 $p \cdot n$ 에 분할하여 학습시킬 경우, T_{p1} 을 네트워크 분할에 소요되는 시간, T_{p2} 를 학습집합 분할에 소요되는 시간이라고 할 때 병렬처리시간 T_p 는

$$\begin{aligned}
 T_p &= T_{p1} + T_{p2} \\
 &= \frac{N_p}{p \cdot n} \cdot \{ (7 \cdot N^2 + 2 \cdot n \cdot N + 2 \cdot \frac{p \cdot n}{N_p} \cdot N^2) \cdot T_G \\
 &\quad + (n+1) \cdot N \cdot T_v \\
 &\quad + (n-1) \cdot N - 2 \cdot \frac{p}{N_p} \cdot N^2 + 2 \cdot \frac{p^2 \cdot n}{N_p} \cdot N^2 \} \cdot T_c \} \quad (14)
 \end{aligned}$$

여기서, $N = N_i = N_h = N_o$

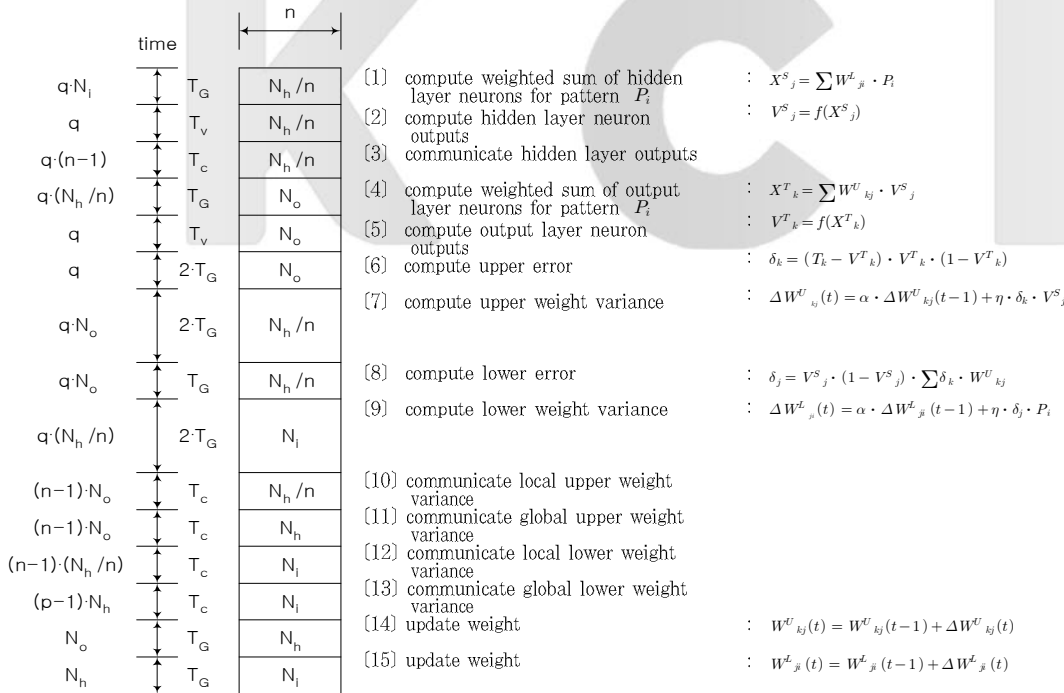


그림 7. 시공간 병렬처리의 시공간도표
Fig. 7 Time-space chart of spatiotemporal parallel processing

이다. (식 14)로부터 N_p 개의 패턴을 $p \cdot n$ 개의 노드컴퓨터에 분할하여 학습시킬 경우의 속도제고율 S_p 는

$$S_p = \frac{T_s}{T_p} = p \cdot n \cdot \frac{1}{\frac{(7 \cdot N^2 + 2 \cdot n \cdot N + 2 \cdot \frac{p \cdot n}{N_p} \cdot N^2) \cdot T_G}{(2 \cdot N + 9 \cdot N^2) \cdot T_G + 2 \cdot N \cdot T_v} + \frac{(n+1) \cdot N \cdot T_v}{(2 \cdot N + 9 \cdot N^2) \cdot T_G + 2 \cdot N \cdot T_v} + \frac{((n-1) \cdot N - 2 \cdot \frac{p}{N_p} \cdot N^2 + 2 \cdot \frac{p^2 \cdot n}{N_p} \cdot N^2) \cdot T_c}{(2 \cdot N + 9 \cdot N^2) \cdot T_G + 2 \cdot N \cdot T_v}} \quad (15)$$

로 계산될 수 있다. 여기서 T_s 는 N_p 개의 입력패턴을 하나의 컴퓨터가 순서적으로 학습하는 시간이다. 즉,

$$T_s = N_p \cdot T_i = N_p \cdot (2 \cdot N \cdot T_G + 9 \cdot N^2 \cdot T_G + 2 \cdot N \cdot T_v) \quad \dots (16)$$

이다. (식 15)에서 $n = 1$ 일 때는 학습집합 분할, $p = 1$ 일 때는 네트워크 분할에 해당된다.

한편, (식 15)에서 노드컴퓨터의 수를 고정시키고, T_G , T_v 및 T_c 를 같다고 가정하면 N 이 ∞ 가 될 때 그리고 N_p 가 ∞ 가 될 때 S_p 는 각각 다음과 같은 결과를 얻을 수 있다. 즉,

$$\lim_{N \rightarrow \infty} S_p \propto \frac{1}{p} \quad (N_p \text{ 고정}) \quad \dots (17)$$

$$\lim_{N_p \rightarrow \infty} S_p \propto \frac{1}{n} \quad (N \text{ 고정}) \quad \dots (18)$$

이다. 따라서 신경망의 크기가 커질 경우에 속도제고율은 학습집합 분할 수에 반비례하므로 네트워크 분할 수를 크게 하면 속도제고율은 증가되는 반면, 학습시킬 패턴의 갯수가 많아지면 속도제고율은 네트워크 분할 수에 반비례하므로 학습집합 분할 수를 강조하는 것이 더욱 효율적이다.

V. 결론

본 논문에서는 MLP 신경망의 패턴 학습과정을 위하여 시공간 병렬성을 고려한 병렬처리모델을 제시하였다. 시간 병렬성을 위한 학습집합 분할과 공간 병렬성을 위한 네트워크 분할을 동시 적용하여 융통성 있는 병렬처리모델을 설계하고자 하였다. 또한 성능평가모델을 통하여 과제에 따라 병렬처리시스템의 학습속도를 높일 수 있는 방안도 해석적으로 모색하였다.

성능평가모델로부터 해석적으로 구한 결과, 패턴 크기 N 이 클수록 네트워크 분할, 패턴 갯수 N_p 가 많을수록 학습집합 분할이 우수한 성능을 나타냄을 확인하였다. 따라서 대규모 과제라고 해도 패턴 크기와 패턴 갯수 중 어느 쪽이 지배적이나에 따라 분할병렬처리 방법이 절충되어야 할 것으로 본다.

참고문헌

- [1] H. Ernst, B. Mokry, and Z. Schreter, "A Transputer based general simulator for connectionist models," *Parallel Processing in Neural Systems and Computers*, pp. 283-286, 1990.
- [2] D. Jackson and D. Hammerstrom, "Distributing backpropagation networks over the Intel iPSC /860 hypercube," *International Joint Conference on Neural Networks*, vol. 1, pp. 569-574, 1991.
- [3] J. Oglesby and J. Mason, "Dynamic scheduling for feedforward neural nets using Transputer," *First IEE International Conference on Artificial Neural Networks*, no. 313, pp. 257-260, 1989.
- [4] 김계경, "Transputer 시스템에서 패턴간 병렬성을 이용한 Backpropagation의 병렬구현," 경북대학교 석사학위 논문, 1992.

- [5] P. Wasserman, *Advanced Methods in Neural Computing*, pp. 97-118, Van Nostrand Reinhold, 1993.
- [6] 강경아, 이기준, 정채영, "은닉노드의 특징값을 기반으로 한 최적신경망 구조의 BPN 성능분석," 한국컴퓨터정보학회논문지, 제5권 2호, pp. 30-36, 2000.
- [7] 이백영, 이태승, 황병원, "학습속도 개선과 학습 데이터 축소를 통한 MLP 기반 화자증명 시스템의 등록속도 향상방법," 대한전자공학회논문지SP, 제39권 3호, pp. 88-98, 2002.
- [8] 장재혁, 박중양, 박재홍, "신경회로망과 기억이론에 기반한 한글영상 인식과 복원," 한국컴퓨터정보학회논문지, 제10권 4호, pp. 17-27, 2005.
- [9] 김복만, 서경석, 최홍문 "NTGST 병렬화를 이용한 고해상도 BLU 검사의 고속화" 대한전자공학회논문지SP, 제41권6호, pp.19-24, 2004.
- [10] 송의석, 성영락, 오하령, "다중 프로세서 환경에서 연결구조에 무관한 휴리스틱 부하평형 알고리즘," 대한전자공학회논문지CI, 제42권 1호, pp. 35-44, 2005.

저자 소개



김성원

1989년 2월 경북대학교 전자공학과
공학사
1993년 2월 경북대학교 전자공학과
공학석사
1995년 8월 경북대학교 전자공학과
박사과정 수료
1997년 9월~현재 수원과학대학
인터넷정보과 조교수
〈관심분야〉 신경망, 병렬처리, 가상
현실

