論文96-33B-9-5

역전파 알고리즘의 전방향, 역방향 동시 수행을 위한 시스톨릭 배열의 설계

(Design of a Systolic Array for forward-backward Propagation of Back-propagation Algorithm)

張 明 淑 *, 柳 基 永 *

(Jang, Meung Sook and Yoo, Kee Young)

요 약

역전파 알고리즘은 분류작업에서 높은 정확도를 얻기 위하여 인공 신경망을 학습시키는데 많은 시간이 걸린다. 그래서 역전파 알고리즘을 병렬처리기로 수행하기 위하여 많은 연구가 되고 있다. 이 논문에서는 효율적인 공간-시간 변환과 PE의 구조로 역전파 알고리즘의 전방향, 역방향 연산을 동시에 수행시키는 시스톨릭 배열을 설계한다. 먼저, 역전파 알고리즘의 전방향, 역방향 연산의 자료흐름을 분석하여 알고리즘에 내재하는 병렬성을 보여주는 DG(Data dependency Graph)를 구한다. 다음, DG의 각 층에 대하여 서로 수직 방향으로 교대로 공간-시간 변환(space-time transformation)을 적용하여 뱀 모양의 선형 시스톨릭 배열(snakelike linear systolic array)을 설계한다. 또한 병렬 수행이 가능한 입력 주기를 계산하고, 전방향, 역방향 연산이 한 PE내에서 동시에 수행될 때에도 정확한 자료들이 정확한 위치에 사용되게하기 위하여 인덱스를 계산하며, 두 연산이 동시에 수행되어도 올바른 결과를 산출하는지를 검증한다. 이렇게 하므로 역전파 알고리즘에 내재하는 병렬성을 극대화시키고, 시스톨릭 배열을 선형화하여 PE의 개수를 최소화시킨다. 또한 전방향, 역방향 연산을 동시에 수행하여 휴면 PE들도 연산에 참여시키므로 수행 시간도 기존의 시스톨릭 배열들 보다 2배 감소 시켰다.

Abstract

Back-propagation(BP) algorithm needs a lot of time to train the artificial neural network(ANN) to get high accuracy level in classification tasks. So there have been extensive researches to process back-propagation algorithm on parallel processors. This paper presents a linear systolic array which calculates forward-backward propagation of BP algorithm at the same time using effective space-time transformation and PE structure. First, we analyze data flow of forward and backward propagations and then, represent the BP algorithm into data dapendency graph(DG) which shows parallelism inherent in the BP algorithm. Next, apply space-time transformation on the DG of ANN in turn with orthogonal direction projection. By doing so, we can get a snakelike systolic array. Also we calculate the interval of input for parallel processing, calculate the indices to make the right datas be used at the right PE when forward and backward propagations are processed in the same PE. And then verify the correctness of output when forward and backward propagations are executed at the same time. By doing so, the proposed system maximizes parallelism of BP algorithm, minimizes the number of PEs. And it reduces the execution time by 2 times through making idle PEs participate in forward-backward propagation at the same time.

* 正會員, 慶北大學校 컴퓨터工學科

(Department of Computer Engineering Kyungpook National University)

接受日字:1996年3月9日, 수정완료일:1996年8月13日

I. 서 론

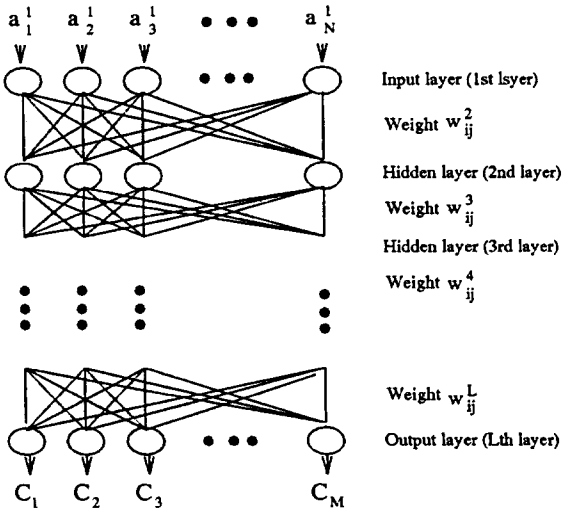
1950년대 중반부터 여러 응용 문제에서 인간과 같은 지능이나 인식 기능을 성취하려는 목적 아래 인공 신경망(Artificial Neural Network, ANN) 모델에 대해

서 많은 연구가 되어왔다. 역전파 알고리즘(Back-propagation algorithm)[14]은 인공 신경망 모델들 중 다층 퍼셉트론(multilayered Perceptron) 인공 신경망 연구에서 가장 많이 사용되는 알고리즘으로 여러 응용 분야에 이용되며 높은 정확도를 나타낸다[4]. 이 알고리즘을 실제 응용문제에 적용할 때는 대부분 순차 수행 컴퓨터 상에서 소프트웨어에 의한 시뮬레이션(simulation)으로 이루어진다[14]. 그러나 인공 신경망의 각 층의 뉴런(neuron)의 수가 증가할수록 순차 수행 컴퓨터 상에서 인공 신경망의 학습 수행은 상당히 많은 시간을 요하고 실제 이용에 많은 문제가 있다. 이 단점을 보완하기 위하여 인공 신경망 문제에 내재되어 있는 병렬성을 이용할 수 있는 병렬 컴퓨터 구조에 대한 연구가 집중되었다. 따라서 어레이 프로세서(Array Processors), 매스파(MasPar)[11], 컨넥션머신(Connection machine)[15], DAP 및 시스톨릭 배열(systolic array)[5,6,7,8,9]과 같은 여러 가지 SIMD형 병렬 컴퓨터에 인공 신경망을 구현하고자 하는 노력이 계속되었다.

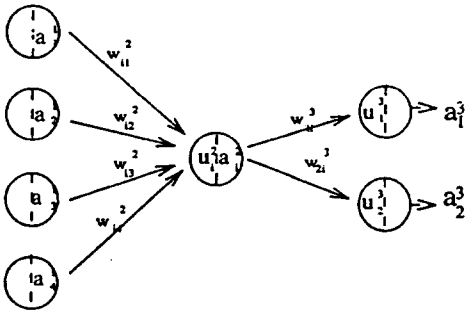
이들 중, 시스톨릭 배열은 간단한 연산을 하는 여러 처리 요소(processing elements, PE)들이 정규적(regular)으로 이웃 PE들과 연결된 네트워크로서 인공 신경망에서 고밀하게 연결된 뉴론(neuron)으로 인하여 발생하는 뉴론 간의 복잡한 통신 문제를 해결하는 가장 좋은 방법 중에 하나이다. Kung을 비롯하여 여러 연구가들이 이 분야에서 계속적인 연구를 해왔으나[2,3,8,9,10,11,12,13,17], 지금까지의 제안된 설계 방법은 귀환 연결 링크(feedback interconnection link)를 사용하여 시간적인 스큐(time skew)현상을 초래한다든지[8,9,17] 혹은 2차원 배열의 구조를 갖는[17] 등의 단점을 가지고 있다. 또한 지금까지 제안된 설계 방법은 연산에 참여하는 PE들보다 휴면 PE(idle PE)들의 개수가 훨씬 많아 PE의 활용도가 매우 낮다.

본 논문에서는 효율적인 공간-시간 변환과 PE의 구조로 역전파 알고리즘의 전방향, 역방향 연산을 동시에 수행시키는 시스톨릭 배열을 설계한다. 먼저, 역전파 알고리즘의 전방향, 역방향 연산의 자료의 흐름을 분석하여 알고리즘에 내재하는 병렬성을 보여주는 DG (data dependency graph)를 구한다. 다음, DG의 각 층에 대하여 서로 수직 방향으로 교대로 공간-시간 변환(space-time transformation)을 적용하여 뱀 모양의 선형 시스톨릭 배열(snakelike linear systolic array)을 설계한다. 또한 병렬 수행이 가능한 입력 주기를 계산하고, 전방향, 역방향 연산이 한 PE내에서 동시에 수행될 때에도 정확한 자료들이 정확한 위치에 사용되게하기 위하여 인덱스를 계산하며, 두 연산이 동시에 수행되어도 올바른 결과를 산출하는지를 검증한다. 이렇게 하므로 각각의 PE 구조는 조금더 복잡해지지만, 역전파 알고리즘에 내재하는 병렬성을 극대화시키고, 시스톨릭 배열을 선형화하여 PE의 개수를 최소화시키며, 전방향, 역방향 연산을 동시에 수행하여 휴면 PE들도 연산에 참여시키므로 수행 시간도 최소화할 수 있다.

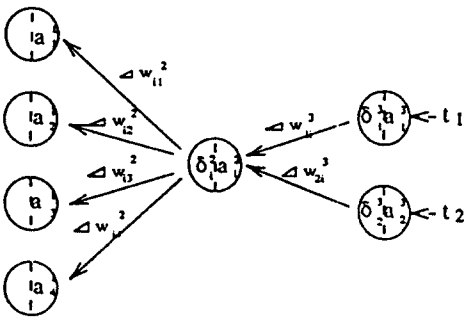
논문의 구성은 다음과 같다. 2장에서는 일반적인 역전파 알고리즘에 대하여 소개하며 이의 뱀모양 시스톨릭 배열(snakelike systolic arry)로의 설계에 대하여 소개한다.



(a) L층의 인공 신경망 구조 (L layered ANN structure)



(b) 전방향 연산 (Forward propagation)



(c) 역방향 연산 (Backward propagation)

그림 1. L층의 인공 신경망 구조와 전방향, 역방향 연산

Fig. 1. L-layered ANN structure and forward, backward propagation.

3장에서는 전방향, 역방향 연산을 동시에 수행할 수 있는 시스톨릭 배열을 제시하며, 그것의 성능, 수행에 대한 검증을, 4장에서는 제안된 시스톨릭 배열을 시뮬레이션한 실험과 결과를, 그리고 5장에서는 이 논문에 대한 결론을 내린다.

Ⅱ. 배 경

이 장에서는 일반적인 역전파 알고리즘에 대하여 설명하고, 역전파 알고리즘을 뱀모양의 선형 시스톨릭 배열로 설계하는 방법과 지금까지 설계된 대표적인 ANN 모델들을 간단히 소개한다. 또한 각 시스템의 시간-공간 복잡도와 PE 활용도(PE utilization)를 계산하여 전체적인 생산성(throughput)의 향상에 대한 필요성을 알아본다.

1. 역전파 알고리즘

계층 1에서 계층 L까지의 다 단계 신경망 모델은 계층 1의 입력 계층(input layer), 계층 2에서 계층 $L-1$까지 $L-2$개의 은닉 계층(hidden layer)들, 그리고 계층 L의 출력 계층(output layer)으로 이루어져 있다. 각 층은 인간의 신경절에 해당하는 뉴런(neuron)들로 구성되어 있으며 이웃한 층의 뉴런들은 가중치(weight)를 가진 연결선들로 서로 연결되어 있다. 그림 1은 L층의 인공 신경망 구조와 각 뉴런에서의 전방향, 역방향 연산을 보여준다. 이 구조 위에서 역전파 알고리즘은 모든 입력패턴 I ($I=\{I_p \mid p=1$에서 n까지, n은 학습이 끝날 때까지 입력된 입력패턴의 개수$\}$)에 대한 전방향과 역방향 연산을 반복 수행하며 인공 신경망을 학습시켜, 정확한 분류 작업을 할 수 있게 한다.

어느 한 입력패턴 I_p에 대한 전방향 연산은, 입력 I_p의 각 원소 $a^1_{i,p}$을 받아들여 각 층에서 활성값을 계산하여 출력을 구하는 단계로, 탐색 단계(searching phase 혹은 retrieving phase)라고도 한다. 이는 아래의 식 (1)과 (2)로 나타낼 수 있다. l 번째 은닉층 혹은 출력층에 있는 각 뉴런 i의 활성값을 계산하기 위하여, 먼저 뉴런 i로 들어오는 총 입력값(net input) $u^l_{i,p}$을 구한다. $(l-1)$층의 뉴런의 개수가 L_{l-1}, l층의 뉴런의 개수가 L_l일 때, $1 \le i \le L_l$에 대하여 뉴런 i의 총 입력값은 아래의 식 (1)과 같다.

$$u^l_{i,p} = \sum_{j=1}^{L_{l-1}} w^l_{ij}\, a^{l-1}_{j,p} + \theta^l_{i,p} \qquad (1)$$

여기서 w^l_{ij}은 $(l-1)$ 번째 층에 있는 뉴런 j에서 l 번째 층의 뉴런 i로 연결된 연결선의 가중치이고, $\theta^l_{i,p}$은 뉴런 i의 임계치이다. 그 다음, 총 입력값 $u^l_{i,p}$에 대한 비선형 함수 f에 의하여 뉴런 i의 활성값 $a^l_{i,p}$을 구하는데, 역전파 알고리즘의 경우 비선형 함수는 아래의 식 (2)와 같다.

$$a^l_{i,p} = f(u^l_{i,p}) = \frac{1}{1 - e^{u^l_{i,p}}} \qquad (2)$$

역방향 연산은 전방향 연산에서 계산된 출력값과 기대 출력값 사이의 오차를 구한 후, 이 오차값을 역방향으로 전파시키며 연결 가중치들을 변화시키는 단계로 학습 단계(learning phase)라고도 한다. 먼저 출력층의 각 뉴런에 대한 오차 $(t_{i,p} - a^L_{i,p})$이 구해진 다음, 이 오차는 아래 식과 같이 출력층에 있는 모든 뉴런 i에 대하여 제곱하여 더해진다.

$$E_P = \sum_{i=1}^{L_L} (t_{i,p} - a^L_{i,p})^2 \qquad (3)$$

역전파 알고리즘의 학습 목표는 이 오차값의 평균을 최소화하는 것이다. 이 목적을 달성하기 위하여 역전파 알고리즘은 연결 가중치 공간에서 기울기 감소(gradient descent), 즉 힐 크라이밍(hill climbing)을 수행한다. 이것은 먼저 출력층의 각 뉴런 i에 대하여 식 (1)을 1차 미분한 뒤 오차에 대한 감소치(error gra-

dient) δ_i^l $(l=1,\ldots,L)$을 아래의 식 (4), (5)와 같이 구한다.

만약 $l=L$ 일 때 $\delta_{i,p}^L=(t_{i,p}-a_{i,p}^L)f'(u_{i,p}^L)$ (4)

만약 $1\le l<L$ 일 때 $\delta_{i,p}^l=(\sum_{j=1}^{L_l}\delta_{j,p}^{l+1}w_{ji}^{l+1})f'(u_{i,p}^l)$ (5)

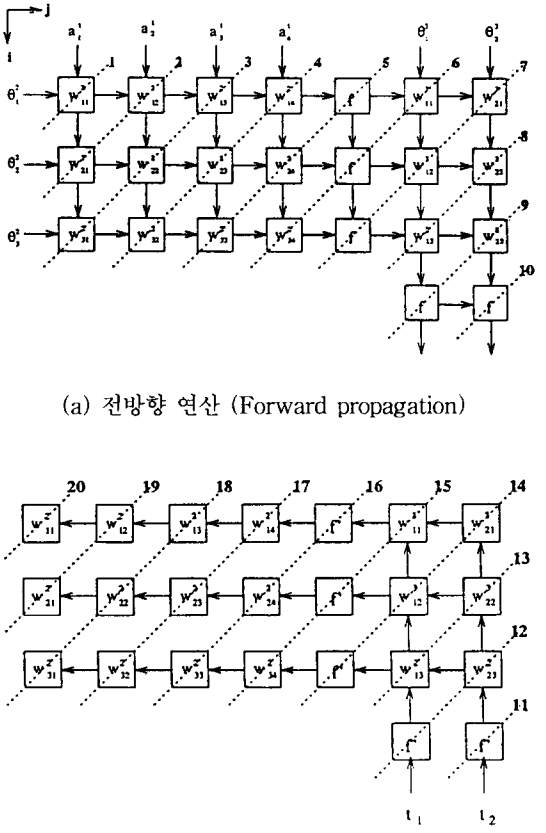
식 (4)는 출력층의 뉴런들에 대한 오차의 감소치이고 식 (5)는 은닉층의 뉴런들에 대한 오차의 감소치이다. 이 오차의 감소치는 역방향으로 신경망의 각 층을 따라 전파되며 식 (6)과 같이 각 층의 가중치를 변경시킨다.

$$w_{ij}^l = w_{ij}^l + \eta\,\delta_{i,p}^l\,a_{j,p}^{l-1} \quad (6)$$

여기서 η는 사용자가 정의한 매개변수(parameter)인 학습 변수(learning rate)이다. 위의 과정은 모든 입력 패턴 I에 대하여 반복되며, 식 (4)의 오차값이, 역시 사용자가 정의해 주는 종료 조건인 'ecrit(error criterion)' 보다 작아지면 학습을 멈춘다.

2. 역전파 알고리즘의 선형 시스톨릭 배열 설계

역전파 알고리즘을 시스톨릭 배열로 설계하기 위한 첫 번째 단계는, 알고리즘에 내재하는 병렬성과 파이프라인 성질을 최대한 이용할 수 있도록 순차 알고리즘을 단일 배정 코드(single assignment code)[6]를 사용하여 병렬 알고리즘으로 나타내는 것이다. 단일 배정 코드는 알고리즘이 수행되는 동안 모든 변수에 오직 한 값만이 배정되는 형태의 프로그램 코드이다. 종래의 코드들은 메모리 공간을 절약하기 위하여 한 변수에 여러번 값을 배정한다. 즉 메모리의 한장소에 시간적인 차이를 두고 값을 배정하므로 동시 실행이 불가능하다. 이러한 변수는 시간의 흐름에 따라 값이 누적되므로, 새로운 인덱스를 첨가하여 서로 다른 인덱스를 가진 변수로 고친 다음 각 변수에 값이 한번만 배정되도록 고친다. 두 번째 단계로 자료들의 의존 관계가 잘 파악되는 DG(data dependency graph)[6]를 구한다. DG란 알고리즘에서 연산을 수행할 때 자료들의 의존 관계를 방향 그래프(directed graph)로 나타낸 것이다. DG에서 각 노드는 연산의 수행을 나타내고, 간선(arc)들은 연산에 필요한 자료들의 의존 관계를 나타낸다. 그림 2에서는 4개의 입력 노드, 3개의 은닉 노드, 그리고 2개의 출력 노드로 이루어진 4-3-2 인공 신경망에 대한 자료의 흐름을 DG로 나타내었다.



(a) 전방향 연산 (Forward propagation)

(b) 역방향 연산 (Backward propagation)

그림 2. 4-3-2 신경망에서 역전파 알고리즘의 자료 흐름을 나타내는 DG

Fig. 2. DG which shows data flow of Back-propagation algorithm on 4-3-2 ANN.

그림 2a)는 전방향 수행을, 그림 2b)는 역방향 수행에 대한 자료의 흐름을 나타낸다. 일단 자료 의존 그래프가 그려지면 공간 변환(space transformation)과 시간 변환(time transformation)을 통해 시스톨릭 배열을 구할 수 있다. 공간 변환은 DG의 각 노드를 시스톨릭 배열의 PE(processing element)로 할당시키는 프로세서 할당(processor assignment)이고 시간 변환은 PE의 연산 순서를 결정해 주는 스케줄링(scheduling)이다. 프로세서 할당과 스케줄링에는 여러 가지 방법이 있으나 DG의 정규성(regurality)을 이용하여 가장 쉽게 할 수 있는 것이 선형 사상(linear projection)과 선형 스케줄링(linear scheduling) 방법이다. DG 상의 노드에서 동시에 수행 가능한 노드들을 선으로 연결하면 동시간 초평면(equitemporal hyperplane)을 얻을 수 있는데 그림 2에서는 점선으로 나타난 선들이다. 이

때 모든 자료의존을 나타내는 간선들은 초평면을 기준으로 같은 방향의 흐름을 가져야 하며 같은 초평면 상에 있는 모든 노드들의 연산은 동시에 이루어질 수 있으므로 서로 다른 프로세서에서 연산을 할 때 병렬성을 극대화시킬 수 있다. 초평면으로부터 초평면에 수직(normal)인 선형 스케줄 벡터 $\vec{t}$를 얻을 수 있는데 $[i,j]$ 축을 기준으로 $\vec{t}=[1,1]^T$ 이 되며, 이는 연산이 i축과 j축의 대각선 방향으로 차례로 수행되어야함을 의미한다.

다음으로 DG의 각 인덱스 포인터가 시스톨릭 배열의 어느 PE에 배정되는지를 명시해 주는 공간 사상 벡터 $\vec{s}$를 구해보면, 공간 사상 벡터 $\vec{s}$는 초평면과 평행이어서는 안된다. 즉 DG상에서 동시에 수행되어야 하는 노드들을 한 PE로 할당하면 병렬 수행이 불가능하므로 반드시 서로 다른 PE에 할당되어야 한다. 이 점을 고려하면 두 가지 가능한 공간 사상을 구할 수 있는데, 그림 2에서의 $[i,j]$ 축을 기준으로, $\vec{s}_1=[1,0]^T$과 $\vec{s}_2=[0,1]^T$이다. 이 논문에서는 공간 사상 $\vec{s}_1$을 이용하여 설계한 시스톨릭 배열을 소개한다. 그림 2에서 DG상의 각 노드들 중, $[1,0]$ 방향으로 같은 줄에 있는 노드들은 한 PE에 할당된다. 즉, 각 노드 $[i,j]$ 는 각 PE j로 사상된다. 1,2 층에 대한 PE들에서는, 입력값 a_j는 각 PE j로 입력하고 임계치 θ_i와 총 입력값 u_i는 각 PE를 따라 차례로 파이프인되며 j축 방향으로 흘러간다. 반면 2,3 층에 대한 PE들에서는, 임계치 θ_i와 총 입력값 u_j가 각 PE j로 입력하고, 입력값 a_j는 PE를 따라 차례로 파이프인되며 j축 방향으로 흘러간다.

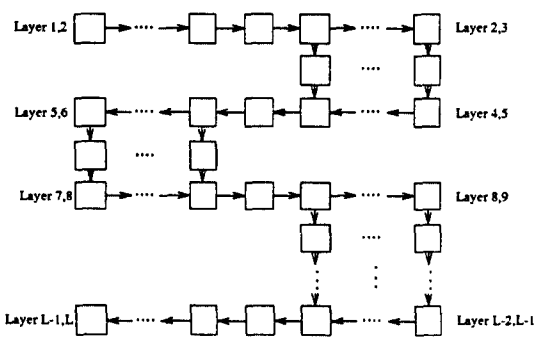


그림 3. L층의 신경망에 대한 시스톨릭 배열의 구조
Fig. 3. Systolic Array structure for L-layered ANN.

이와 같은 방법으로 서로 다른 층의 PE들에 대하여 a와 u, θ가 서로 다른 방향의 자료 흐름을 교대로 가지도록 사상을 시키면 그림 3과 같은 뱀 모양인 L층의 다 단계 신경망에 대한 선형 시스톨릭 배열을 구현할 수 있다[16].

3. 역전파 ANN 모델들의 성능 분석

Kung은 인공 신경망을 위한 행렬-벡터 곱셈에서 폭포형의 DG (Cascaded DG)를 이용하여 시스톨릭 배열을 설계하였다[7,8,9]. 이 폭포형의 DG를 사상시키면 링 구조의 긴 귀환 연결 링크를 갖는 시스톨릭 배열을 얻을 수 있는데, 링 시스톨릭 배열에서는 입력값 $a^1_{j,p}$은 연결선을 따라 원형으로 흐르고 총 입력값 $u^l_{i,p}$은 각 노드에 머물면서 누적된다. 이 경우에는 다 단계 신경망에서 입력층, 은닉층들, 그리고 출력층 노드의 개수가 모두 같아야 한다는 단점이 있다. 만일 서로 다른 뉴런의 수로 구성된 다 단계 신경망이라면 누적된 총 입력값을 다음 단계로 전달하기 위한 여분의 시간과 신호가 필요하다.

다 단계 신경망을 2차원 시스톨릭 배열로 변환한 경우에는[17] 앞절의 그림 2에서 구한 DG의 노드와 간선들을 그대로 2차원 시스톨릭 배열로 사상 시키고 각 행과 열의 PE들을 링 모양의 연결선으로 다시 연결시켰다.

앞 절에서 소개한 뱀모양의 선형 시스톨릭 배열의 경우는 자료들을 입력층에서 은닉층으로, 은닉층에서 출력층으로 또한 출력층에서 은닉층을 거쳐 다시 입력층으로 정규적으로 흐르도록 하며, 각 층의 노드들이 적절한 신호를 이용해 활성값이나 혹은 총 입력값을 선택적으로 흘려 보냄으로 각 층의 뉴런의 개수가 다른 다 단계 신경망에서도 연산이 규칙적으로 이루어질 수 있다. 또한 PE 내의 레지스트들을 적절히 사용하므로 링 연결선이 없이도 올바른 계산이 이루어질 수 있다. 링 시스톨릭 배열과 뱀 모양의 선형 시스톨릭 배열의 경우 공간 복잡도는 $O(L_k)$ (L_k는 k 번째 층의 PE의 개수)이고, 2차원 시스톨릭 배열의 공간 복잡도는 $O(L_k \times L_{k+1})$이다.

다음, 시스톨릭 배열의 시간 복잡도를 살펴보면, 순차적 시스톨릭 배열의 경우에는 총 시간 스텝이 [입력패턴의 개수 $\times 2\{\sum_{k=1}^{(L-1)}(L_k \times L_{k+1}) + \sum_{k=2}^{(L-1)} L_k\} + 1$]으로 $O(L_k \times L_{k+1})$이다. 링 시스톨릭 배열과 2차원 시스톨릭 배열, 그리고 뱀 모양의 선형 시스톨릭의 경우는

[입력패턴의 개수 $\times 2\{L_1+1+2\sum_{k=2}^{L/2}(L_{2k-1}+1)+L_L\}$]로 $O(L_k)$이다[16]. 결과적으로 역전파 알고리즘을 시스톨릭 배열을 이용하여 수행시켰을 때 선형 시스톨릭 배열 설계가 알고리즘의 병렬성을 최대한 이용할 수 있으므로 수행 시간과 공간을 최소화할 수 있다.

그러나 지금까지 설계된 모든 시스톨릭 배열들은 대부분의 경우 연산에 참여하는 PE들보다 휴면 PE들의 개수가 훨씬 많아 PE의 활용도가 매우 낮다. 그림 4는 4-3-2 선형 시스톨릭 배열상에서 수행되는 연산의 상태 중 전방향 연산을 단계별 스냅샷(snapshot)으로 나타낸 것이다.

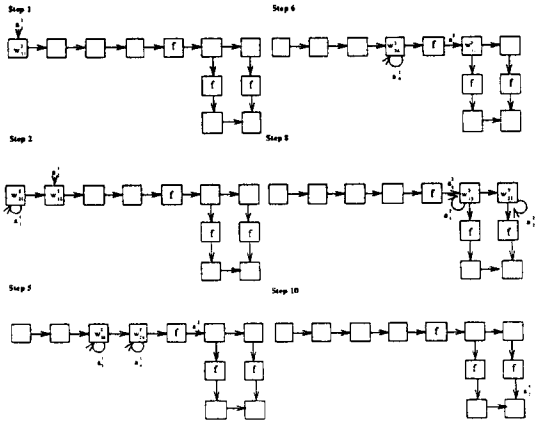


그림 4. 4-3-2 시스톨릭 배열의 연산에 대한 snapshot (전방향 연산)

Fig. 4. The snapshot for the operation of 4-3-2 Systolic Array.

그림 4에서 보듯이 선형 시스톨릭 배열의 경우 약 $\sum_{k=1}^{L}L_k$개의 PE들 중 각 단계마다 약 L_k개의 PE에서만 연산이 이루어지고, 나머지 PE들은 학습이 끝날 때까지 연산이 일어나지 않고 쉬게되어 결과적으로 약 $(L_k / (\sum_{k=1}^{L}L_k)\times 100)$ %의 PE 활용도를 나타낸다. 그림 4의 경우에는 7개의 PE들 중 각 단계에서 최대 3개의 PE들에서만 연산이 일어나므로 약 43 %의 PE 활용도를 나타낸다. 그러나 실제로 인공 신경망을 응용문제에 이용하기 위해서는 은닉층의 노드 수보다 월등히 많은, 수백 혹은 수천 개의 입력 뉴런들이 필요하며 따라서 대부분의 PE들이 연산에 참여하지 않고 쉬게 되어 PE의 낭비가 심하다. 이는 링 시스톨릭의 경우나 2차원 시스톨릭 배열의 경우에도 같다. 따라서 PE 개수의 증가에 따라 늘어나는 휴면 PE의 개수를 줄여 PE의 활용률을 높여 전체적인 생산성의 향상에 대한 연구가 필요하다. 다음 장에서는 휴면 PE를 최대한 활용하여 PE 효율성을 증가시킬 수 있는 선형 시스톨릭 배열의 설계에 대하여 설명한다.

Ⅲ. 전방향, 역방향 동시수행의 시스톨릭 배열 설계

이 장에서는 앞 장에서 설명한 뱀 모양의 시스톨릭 배열에서 서로 다른 패턴을 갖는 두가지 입력 자료를 입력시켜 휴면 PE들이 쉬지않고 연산에 참여하도록 하며, 기존의 시스템보다 생산성(throughput)을 100 % 까지 증가 시키는 선형 시스톨릭 배열의 설계에 대하여 설명한다. 이를 위하여 병렬 수행이 가능한 두 입력패턴의 입력 주기를 계산하고, 전방향, 역방향 연산을 동시에 수행할 수 있는 PE의 내부 구조를 설계하며, 전방향, 역방향 연산이 한 PE내에서 겹쳐질 때에도 정확한 자료들이 정확한 위치에 사용되게하기 위한 인덱스 계산, 그리고 두 연산이 동시에 수행되어도 올바른 결과를 산출하는지를 검증한다.

1. 전방향, 역방향 병렬 수행의 자료흐름

앞 절에서 살펴보았듯이 역전파 알고리즘은 모든 입력패턴에 대하여 (1)에서 (6)의 계산을 ecrit이 충분히 작아질 때까지 반복 수행한다. 어느 한 입력패턴 I_p에 대한 전방향, 역방향 연산이 선형 시스톨릭 배열 상에서 수행되는 상태는 그림 4에서 살펴 보았다. 여기서 만일, 입력패턴 I_p에 대한 전방향 연산과 입력패턴 I_{p-1}에 대한 역방향 연산을 겹친다면 전방향, 역방향 연산이 동시에 수행되는 선형 시스톨릭 배열을 구할 수 있을 것이다. 이를 위하여 먼저 전방향, 역방향 병렬 연산이 동시에 일어날 때의 자료흐름을 생각해본다. 그림 2의 4-3-2 인공 신경망에 대한 DG의 자료 흐름을 분석해보면, 역방향 수행은 전방향 수행과 대칭으로 이루어 진다는 것을 알 수 있다.

즉, 전방향 연산은 $[i,j]$ 축을 기준으로 $[1,1]$ 방향인 1번에서 10번 점선으로, 역방향 연산은 $[-1,-1]$ 방향인 11번에서 20번 점선으로 서로 간섭없이 차례대로 수행되므로 이들은 동시 수행이 가능하다. 이들 전방향, 역방향 DG를 동시수행을 위하여 겹치면 그림 5와 같은 병렬 수행을 위한 DG를 얻을 수 있다.

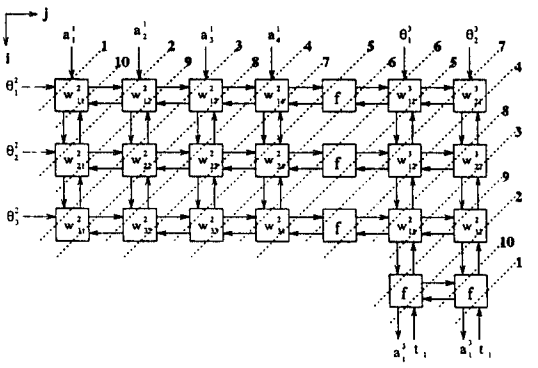


그림 5. 4-3-2 신경망에서 전방향, 역방향 병렬 수행의 자료 흐름을 나타내는 DG

Fig. 5. DG which shows data flow of forward-backward parallel processing on a 4-3-2 Systolic Array.

그림 5에서 $[i,j]$ 축을 기준으로 했을 때, DG의 $[1,1]$ 방향은 한 입력패턴 I_p에 대한 전방향 연산으로, 1,2층에 대한 DG에서는, 입력값 $a^1_{i,p}$이 $[1,0]$ 방향으로 흐르며 w^2_{ij}와 곱해지고 임계치 $\theta^2_{j,p}$와 총 입력값 $u^2_{j,p}$는 $[0,1]$ 방향으로 흐르며 누적된다. 2,3층에 대한 DG에서는, f노드에서 계산된 활성값 $a^2_{i,p}$는 $[0,1]$ 방향으로 흐르며 w^3_{ij}과 곱해지고 임계치 $\theta^3_{j,p}$과 총 입력값 $u^3_{j,p}$은 $[1,0]$ 의 방향으로 흐르며 누적된다. 다음 f노드에서는 비선형함수 계산으로 $a^3_{i,p}$을 산출한 후 출력층으로 입력되는 기대 출력값 $t_{i,p}$와 비교되어 오차가 계산된 다음 학습을 계속할 것인가를 결정한다.

동시에 DG의 $[-1,-1]$ 방향은 이전 학습 단계에서 미리 입력된 입력패턴 I_{p-1}에 대한 역방향 연산으로, 전방향 연산의 결과로 산출된 출력층의 활성값 $a^3_{i,p-1}$과 출력층으로 입력되는 기대 출력값 $t_{i,p-1}$과의 오차값에 의하여 식 (4)와 같이 오차 감소치 $\delta^3_{i,p-1}$이 f노드에서 계산된다. 다음 $\delta^2_{i,p-1}$를 구하기 위해 $\delta^3_{i,p-1}$이 w^3_{ji}과 곱해지며 누적되고 이 값은 2,3층에 대한 DG를 따라 역방향인 $[0,-1]$ 방향으로 연결선을 따라 흐른다. 이때 $\delta^3_{i,p-1}$은 w^3_{ji}을 (6)식과 같이 변경시켜 가중치에 대한 학습을 수행한 후 $[-1,0]$ 방향으로 연결선을 따라 전파된다.

전방향, 역방향 연산에서 사용되는 가중치 w^l_{ij} $(l=2,3)$들의 인덱스 변화를 살펴보면, 전방향 연산의 경우, $l=2$일 때에는 같은 $[i,j]$ 축을 따라 각각 1씩 증가하고, 반대로 역방향 연산의 경우, $l=3$일 때에는 각각 1씩 감소하는 것을 알 수 있다. 그러므로 가중치 w^l_{ij} $(l=2,3)$들을 각 PE내의 레지스트에 저장하여 두고 매 시간 스텝마다 각 PE내의 연산에 필요한 가중치의 인덱스를 1씩 증가 혹은 감소시키며 흘려보내면 정확한 자료를 적절한 위치에서 올바르게 사용하여 같은 시간에 전방향, 역방향 연산을 동시에 수행할 수 있을 것이다.

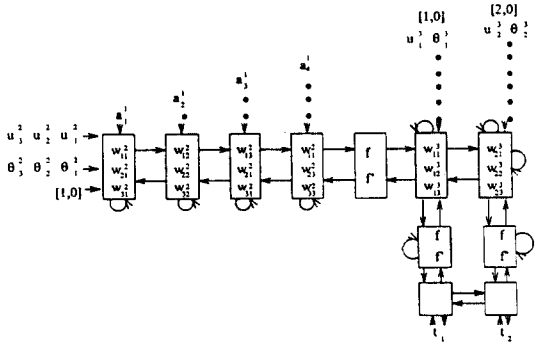


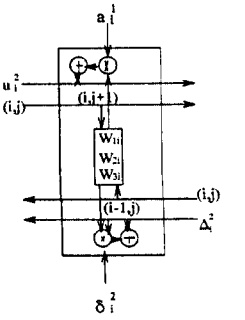
그림 6. 4-3-2 신경망에 대한 전방향 역방향 동시수행 시스톨릭 배열의 구조

Fig. 6. Systolic Array structure which operates forward-backward propagation at the same time for the 4-3-2 ANN.

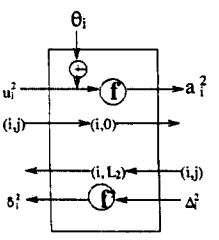
2. 전방향, 역방향 병렬 수행을 위한 PE의 구조

앞절에서와 같이 공간사상 $\vec{s}_1=[1,0]^T$ 를 이용하여 그림 5의 DG를 사상시키면 그림 6과 같은 전방향, 역방향 동시 수행이 가능한 시스톨릭 배열을 얻을 수 있다. $L(L\geq 2)$층의 시스톨릭 배열에 대한 PE의 개수는 논문 [16]의 정리 1과 같다.

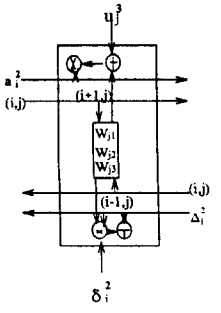
그러나 각 PE는 한 PE내에서 전방향, 역방향 연산을 동시에 수행시킬 수 있는 구조를 가져야한다. 그림 7은 전방향, 역방향 동시 연산이 가능한 각 층의 PE들의 구조를 나타낸다.



1,2 층에 대한 PE
(PE for 1st,2nd-layer)



f, f' PE
(PE for f, f')



2,3 층에 대한 PE (PE for 1st,2nd-layer)

그림 7. 각 층의 PE 구조
Fig. 7. PE structure on each layer.

1,2층에 대한 시스톨릭 배열에서는, 전방향 연산을 위하여, 입력값 $a^1_{j,\,p}$는 각 PE j로 입력하여 레지스트에 저장되고 총 입력값 $u^2_{i,\,p}$는 각 PE를 따라 차례로 파이프인되며 누적된다. 임계치 $\theta^2_{i,\,p}$는 각 PE를 따라 전달되며 마지막 f PE에 가서 활성값을 구하는데 이용된다. 가중치 w^2_{ij}들은 PE내의 레지스터에 저장되어 있으며, 매 시간 스텝에서 연산에 참여하는 가중치를 구하기 위하여 인덱스 $[i,j]$ 값을 입력하여 각 PE들을 파이프라인하며 $[i,j+1]$ 로 누적된다. 전방향 연산시 각 PE로 입력되는 입력값 $a^1_{j,\,p}$과 $\theta^2_{i,\,p}$는 각각 한 시간스텝의 차이를 두고 입력되어 매 스텝마다 각 PE에서는 $(u^2_{i,\,p}+w^2_{ij}\times a^1_{j,\,p})$의 연산이 이루어진 뒤 다음 PE로 파이프라인 된다. 이렇게 누적된 값들은 f PE에서 활성값 $a^2_{i,\,p}$로 변환된다. 역방향 연산을 위하여 $\delta^2_{i,\,p-1}$ 값은 역방향 연결선을 따라 앞쪽의 PE로 전달되고, 매 시간 스텝에서 연산에 참여하는 가중치를 구하기 위하여 인덱스 $[i,j]$ 가 각 PE들을 파이프라인하며 $[i-1,j]$ 로 누적된다.

이때 $a^1_{j,\,p-1}$은 한단계 이전에 미리 입력된 입력패턴 I_{p-1}에 대한 전방향 연산시 사용된 값들이 레지스트에 저장되어 있다가 다시 사용된 것이다. 이 값들을 이용하여 가중치를 갱신하는 연산 $(w^2_{ij}=w^2_{ij}+\eta\,\delta^2_{i,\,p-1}\,a^1_{j,\,p-1})$이 각 PE에서 수행된다.

반면 2,3층에 대한 시스톨릭 배열에서는, 앞 층에서 계산된 활성값 $a^2_{i,\,p}$는 각 PE i로 전달되며 레지스트에 저장되고, 총 입력값 $u^3_{i,\,p}$은 각 PE에 머물며 누적된다. 임계치 $\theta^3_{i,\,p}$도 각 PE에 머물다가 f PE에 가서 활성값을 구하는데 이용된다. 가중치 w^3_{ij}들은 PE내의 레지스터에 저장되어 있으며, 매 시간 스텝에서 연산에 참여하는 가중치를 구하기 위하여 인덱스 $[i,j]$ 가 각 PE들을 파이프라인하며 $[i+1,j]$ 로 누적된다. 전방향 연산시 각 PE로 전달되는 활성값 $a^2_{j,\,p}$와 $\theta^3_{i,\,p}$은 $u^3_{i,\,p}+w^3_{ij}\times a^2_{j,\,p}$의 연산을 하며 PE내에서 누적된다. 이렇게 누적된 값들은 f PE에서 활성값 $a^3_{i,\,p}$으로 변환된다. 역방향 연산을 위하여 $\delta^3_{j,\,p-1}$ 값은 역방향 연결선을 따라 앞쪽의 PE로 전달되고, $(\delta^2_{j,\,p-1}+\delta^3_{j,\,p-1}\,w^3_{ij})$ 연산과 $(w^3_{ij}=w^3_{ij}+\eta\,\delta^3_{i,\,p-1}\,a^2_{j,\,p-1})$ 연산이 각 PE에서 이루어진다. 이때 매 시간 스텝에서 연산에 참여하는 가중치를 찾기 위하여 인덱스 $[i,j]$ 가 각 PE들을 파이프라인하며 $[i-1,j]$ 로 누적되고, $a^2_{j,\,p-1}$는 한단계 이전에 미리 입력된 입력패턴 I_{p-1}에 대한 전방향 연산시 계산된 값들이 레지스트에 저장되어 있다가 다시 사용된 것이다.

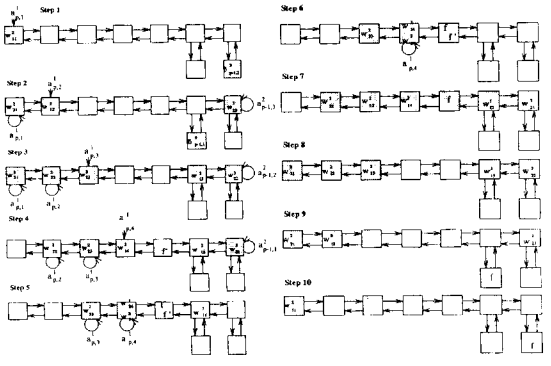


그림 8. 4-3-2 시스톨릭 배열에서 전방향 역방향 동시 수행 연산에 대한 snapshot
Fig. 8. Snapshot of simultaneous forward-backward propagations on 4-3-2 Systolic Array.

3. 선형 시스톨릭 배열에서 전방향, 역방향의 병렬 수행

그림 8은 4-3-2 선형 시스톨릭 배열에서 전방향과 역방향이 동시에 일어날 경우의 실행을 단계별로 나타낸 것이다. 첫 번째 스텝에서, 첫 번째 PE에서는 입력패턴 I_p에 대한 전방향 연산으로 입력값 $a^1_{1,\,p}$과 가중치 w^2_{11}과의 곱이, 일곱 번째 PE에서는 입력패턴 I_{p-1}에 대한 역방향 연산으로 오차의 감소치 $\delta^3_{2,\,p-1}$의 계산이 동시에 수행된다. 두 번째 스텝에서는, 첫 번째 PE에서 $a^1_{1,\,p}$과 w^2_{21}의 곱셈이, 두 번째 PE에서 $a^1_{2,\,p}$과 가중치 w^2_{12}의 곱이 첫 번째 스텝에서 전해져온 $u^2_{1,\,p}$ 값에 누적되고, 동시에 여섯 번째 PE에서 $\delta^3_{2,\,p-1}$ 계산이, 일곱 번째 PE에서는 은닉층으로 전달

될 누적 오차 감소치 $\delta^3_{2,p-1}$계산과 가중치 w^3_{23}의 갱신이 동시에 수행된다. 이렇게 매 스텝마다 병렬 계산되어 양 방향으로 전해져 온 연산들은 다섯 번째 그리고 여섯 번째 스텝에서는 네 번째와 다섯 번째 PE에서 각각 만나게 되며, 이들 PE에서는 한 PE내에서 양방향 연산이 동시에 수행된다. 열 번째 스텝에서, 첫 번째 PE에서는 w^1_{11}의 갱신이, 일곱 번째 PE에서는 $a^3_{2,p}$을 계산한 후, 다시 새로운 입력패턴 I_{p+1}에 대한 전방향 연산과 입력패턴 I_p에 대한 역방향 연산인 가중치 갱신이 반복된다.

구현된 시스톨릭 배열의 수행 시간을 살펴보면 아래와 같은 정리를 얻을 수 있다.

[정리 1] L_1개의 노드를 갖는 입력층, 각각 L_2, L_3, ..., L_{L-1}개의 노드를 갖는 2층에서 $L-1$층의 은닉층들, 그리고 L_L개의 노드를 갖는 출력층으로 구성된 L ($L\geq 2$)층의 역전파 신경망에 대한 전방향, 역방향 동시 수행의 선형 시스톨릭 배열은 아래와 같은 총 학습 수행 횟수를 갖는다.

만약 $L=2$ 이면 [입력 패턴의 개수 $\times\{2(L_1+L_2+1)-1\}/2$]

만약 $L=3$ 이면 [입력 패턴의 개수 $\times\{2(L_1+L_2+L_3+2)-1\}/2$]

만약 $L\geq 4$ 이고 L이 짝수이면

[입력 패턴의 개수 $\times 2(L_1+1+2\sum_{k=2}^{\frac{L}{2}}(L_{2k-1}+1)+L_L)/2$]

만약 $L\geq 4$ 이고 L이 홀수이면

[입력 패턴의 개수 $\times 2(L_1+1+2\sum_{k=2}^{\lfloor\frac{L}{2}\rfloor}(L_{2k-1}+1)+L_{L-1}+L_L+1)/2$]

[증명] 설계된 선형 시스톨릭 배열에서 전방향, 역방향 연산의 수행에 걸리는 시간은 논문 [16]의 정리 1과 같다. 같은 구조의 선형 시스톨릭 배열에서 전방향, 역방향을 동시에 수행할 경우는 전방향, 역방향 연산이 겹치므로 수행 시간을 반으로 줄일 수가 있다. 따라서 전방향 역방향 동시 수행 선형 시스톨릭 배열은 위의 식과 같은 총 학습 수행 횟수를 갖는다. ■

여기서 전방향, 역방향 동시 수행을 위한 입력패턴 I_p와 I_{p-1}에 대한 입력 주기 S 값은 아래의 정리 2에서와 같이 구할 수 있다.

[정리 2] L_1개의 노드를 갖는 입력층, 각각 L_2, L_3, ..., L_{L-1}개의 노드를 갖는 2층에서 $L-1$층의 은닉층들, 그리고 L_L개의 노드를 갖는 출력층으로 구성된 L ($L\geq 2$)층의 역전파 신경망에서 한 입력패턴 I_{p-1}이 입력 된 후 S스텝 다음에 입력패턴 I_p가 입력된다면 입력 주기 S는 아래와 같이 구할 수 있다.

만약 $L=2$ 이면 $S=L_1+L_2+1$

만약 $L=3$ 이면 $S=L_1+L_2+L_3+2$

만약 $L\geq 4$ 이고 L이 짝수이면 $S=L_1+1+2\sum_{k=2}^{L/2}(L_{2k-1}+1)+L_L$

만약 $L\geq 4$ 이고 L이 홀수이면 $S=L_1+1+2\sum_{k=2}^{\lfloor L/2\rfloor}(L_{2k-1}+1)+L_{L-1}+L_L+1$

[증명] 전방향, 역방향 동시 수행은 각각 따로 수행할 때보다 1/2의 시간이 걸린다. 따라서 설계된 선형 시스톨릭 배열에서 전방향, 역방향 연산의 수행에 걸리는 시간을[16] 2로 나누면 된다. ■

다음은 위의 정리 2에서와 같은 주기로 입력된 두 입력패턴에 대한 학습이 동시에 이루어져도 올바른 학습이 이루어지는지에 대하여 살펴본다. 역전파 알고리즘의 학습에 대한 기본 개념은 LMS(Least Mean Squared Error)와 신경망 가중치에 대한 함수의 미분값을 이용하여 가중치를 학습에 적절한 값으로 변경시키는 것이다. 이때 한 입력패턴 I_p를 학습한 후의 가중치 변경값은 아래와 같이 나타낼 수 있다.

$$\Delta w_{ij,p} = -k\frac{\partial E_p}{\partial w_{ij,p}} = \varepsilon\delta_{i,p}\, a_{j,p}$$

여기서 $a_{j,p}$, $\delta_{i,p}$, E는 각각 한 입력패턴 I_p에 대하여 2장의 식 (2), (3), (4), (5)에서 얻은 값이다. 이 가중치 변경값은 'ecrit'이 주어진 값보다 작아질 때까지 계속 변경되며, 만일 ecrit이 주어진 값보다 작아질 때까지의 변경 횟수, 즉 학습 횟수를 N이라 하면 아래와 같은 식을 얻을 수 있다.

$$\lim_{p\to N}\Delta w_{ij,p} \cong 0$$

즉,

$$\lim_{p\to N}\varepsilon\delta_{i,p}\, a_{j,p} = \lim_{p\to N}\varepsilon(t_{i,p}-a_{i,p})\, f(u_{i,p})\, a_{j,p} \cong 0$$

이때 모든 입력패턴 I_p는 서로 독립적이며 모든 입력패턴 I_p에 대하여 학습이 계속될수록 가중치 변경값은 감소한다. 여기서 만일 한 입력패턴 I_p에 대한 전방향 연산과 I_{p-1}에 대한 역방향 연산이 동시에 수행된다면 두 연산이 겹쳐지는 노드에서는, 2장의 총 입력값 계산

식 (1)과 가중치 변경 부분의 식 (6)은 각각 아래와 같이 나타날 것이다.

$$u_i^l = \sum_{j=1}^{L_{l-1}} (w_{ij,p}^l - \Delta w_{ij,p}) \; a_j^{l-1} + \theta_i^l$$

$$\Delta w_{ij,p} = \Delta w_{ij,p} - \Delta w_{ij,p-1}$$

그러나 $(0 \leq \Delta w_{ij,p-1} \leq 1)$이고, 식 (6)에 의하여 학습이 계속될수록 $\Delta w_{ij,p-1}$이 0에 가까운 값을 가지므로 $\Delta w_{ij,p-1}$의 값은 학습수행에 영향을 미치지 않는다는 것을 알 수 있다.

그러므로 두 연산이 동시에 수행되어도 학습이 올바로 진행된다는 것을 알 수 있다. 이를 정리하면 아래와 같은 정리 3을 얻을 수 있다.

[정리 3] L_1개의 노드를 갖는 입력층, 각각 $L_2, L_3, \ldots, L_{L-1}$개의 노드를 갖는 2층에서 $L-1$층의 은닉층, 그리고 L_L개의 노드를 갖는 출력층으로 구성된 L $(L \geq 2)$층의 역전파 신경망에서 한 입력자료 I_p에 대한 전방향 연산과 S 스텝 이전에 입력된 다른 입력패턴 I_{p-1}에 대한 역방향 연산을 병렬로 수행하여도 올바른 학습이 이루어진다.

Ⅳ. 시뮬레이션 실험과 결과

이 장에서는 앞장에서 제안한 전방향, 역방향 동시수행을 위한 시스톨릭 역전파 신경망 시스템을 C 프로그래밍 언어로 시뮬레이션하여 올바른 결과를 산출하는지를 검사하고 기존의 여러 역전파 신경망 시스템들과 수행시간을 비교한다.

1. 실험을 위한 자료와 실험방법

실험을 위한 자료는 미국 캘리포니아대학(University of California, Irvine)의 공공 데이타 베이스로 콩의 병의 종류와 병에 걸렸을 때 나타나는 현상들에 관한 기록을 사용하였다. 이 자료는 47개의 서로 다른 대상(object)들이 날씨, 일년 중 어느 때, 잎과 줄기의 상태 등과 같은 35가지의 속성(attribute)들로 표현되어 있으며, 이들 각각의 속성에 대응되는 속성값들을 2진수로 표현한 총 105개의 입력 노드를 이루고 있다. 각 대상은 'charcoal-rot', 'phytophthora-rot', 'rhizoctonia-root-rot', 그리고 'diaporthe-stem-canker' 등과 같은 4가지 콩의 병들 중 한 부류(class)에 속하며 이들 역시 2진수로 바꾸어 기대 출력값(target output)으로 사용하였다. 각 입력자료들은 무작위로 순서를 바꾼 다음, 신경망을 훈련 시키는데 사용될 훈련용 자료(training data set)와 훈련을 마친 뒤 신경망의 학습 정도를 측정하기 위한 성능 검사용 자료(test data set)로 나누었다. 훈련용 자료 집합의 각 대상들은 순서대로 한줄씩 신경망에 입력되어 각 층에 있는 뉴런들에서의 활성값을 차례대로 산출한 뒤 출력값을 계산해내고, 출력값과 결과값 사이의 오차에 의해 연결 가중치를 변화 시킨다. 이러한 동작을 종료 조건이 만족될 때까지 모든 입력 자료에 대하여 연속적으로 수행한다.

전체 자료 집합에 대하여 훈련을 수행한 후에도 종료 조건이 만족되지 않으면 이 자료 집합은 종료 조건이 만족될 때까지 반복해서 다시 차례로 입력되어 훈련에 사용된다. 역전파 신경망 모델은 이렇게 먼저 훈련용 자료의 집합으로 부터 분류코드를 학습한 뒤, 성능 검사용 자료 집합을 이용하여 얼마나 분류를 정확하게 수행하는 지에 대한 정확도가 측정된다. 실험은 각각의 자료들에 대하여 은닉층의 뉴런의 개수는 10개, 모멘툼은 0.9, 학습율은 0.20에 대하여 행하였는데 이들은 가장 보편적인 매개변수 값이다. 모든 실험은 극단적인 결과를 피하기 위하여 무작위로 순서를 바꾸어 섞어놓은 10개의 서로 다른 훈련용 자료 집합과 검사용 자료 집합에서 각각 실험을 행한 뒤 평균값을 취하였다.

3.1절의 그림 6과 같은 전방향, 역방향 동시수행을 위한 3층의 시스톨릭 배열을 구현하면 105개의 구조체들이 1,2층에 대한 PE들을 이루고 각각의 PE들은 은닉층의 뉴런의 개수와 같은 수인 10개의 배열원소에 신경망의 초기 가중치값인 0을 모두 저장하고 있다. 또한 각 구조체 내에는 a_j^i, u_i^i, θ_j^i , 그리고 인덱스 값을 저장할 4개의 버퍼들이 있다. 1,2층에 대한 PE 구조체는 2진수로 변환된 입력값 a_j^1을 차례대로 입력받아 해당되는 버퍼에 저장해 두었다가, 다음 입력값이 입력되었을 때 다시 사용하며, u_i^2 값을 계산하여 옆의 PE로 전달한다. θ_j^2 값은 변화없이 옆으로 전달되어 f 노드에서 비선형 함수를 계산하는데 이용된다. 연산에 사용할 인덱스 값을 계산하기 위하여 초기 인덱스값 [1,0] 이 버퍼로 입력되어 그림 7a)에서와 같은 연산을 수행한다. 다음의 한 PE는 비 선형함수 계산을 해주는 PE로 1개의 버퍼가 있어 총 입력 값 u_j^2를 저장하였다가 역방향 연산 시 이용한다. 2,3층에 대한 PE

구조체는 출력 노드의 개수와 같은 4개로 a_j^2 값을 저장하였다가 역방향 연산 시 이용하기 위한 10개의 배열원소와 인덱스 값을 계산하여 찾기 위한 버퍼를 가지고 그림 7b)에서와 같은 연산을 수행한다. 다음 4개의 PE들은 출력층의 비선형 함수를 구하기 위한 PE들로 1개의 버퍼가 있어 총 입력값 u_i^3을 저장하였다가 역방향 연산 시 이용한다. 마지막으로 출력층의 모든 출력 값들을 종합하여 학습을 계속할 것인가를 결정해 주는 연산을 수행하는 PE의 구조체들이 있다.

이러한 구조를 가진 전방향, 역방향 동시 수행의 시스톨릭 배열은 앞절의 그림 8에서와 같은 연산을 모든 입력패턴 I에 대하여 반복하여 수행한다. 그런데 입력패턴 I_1에 대한 전방향 수행시에는 계산된 출력값이 없으므로 반대쪽에서 오는 역방향 연산의 동시 수행이 불가능하다. 그러므로 처음부터 전방향, 역방향 동시 수행을 위해서 역방향 연산을 위한 잉여입력(dummy input)이 필요하다. 즉 여분의 출력값으로 실험의 결과에 영향을 미치지 않는 0을 출력층의 각 노드로 입력시키면 첫 번째 수행부터 전방향, 역방향 연산이 가능하다. 이렇게 계속된 연산은 각 입력패턴 I_p에 대한 전방향 연산과 I_{p-1}에 대한 역방향 연산을 계속하여 PE내의 가중치들을 누적, 변경시켜서 훈련용 자료 집합에서의 판단 정확도가 95 %(ecrit 0.02)가 될 때까지 반복 훈련을 시킨다.

2. 수행시간과 정확도 비교

앞절에서와 같은 방법으로 3층의 105-10-4 신경망을 기존의 순차적 학습 방법, 이미 제시된 링 시스톨릭 배열, 2차원 시스톨릭 배열, 그리고 제안된 시스톨릭 배열로 구현했을 때의 시간 스텝을 비교하여보면 그림 9와 같다. 훈련용 자료 집합에서의 판단 정확도가 95 %가 될 때까지 반복 훈련에 입력된 총 입력 패턴의 개수는 순차적 수행의 경우에는 7회로, 즉 I_p에서 $p=7$, 총 시간 스텝이 $[7\times 2(105\times 10+10\times 4)]=29{,}400$번의 시간 스텝이 걸린다. 선형시스톨릭 배열의 경우에는 8회로, 즉 I_p에서 $p=8$, $[8\times\{2(105+10+4+2)-1\}]=1{,}928$번의 시간 스텝이 걸린다. 이 논문에서 제안된 방법인 입력 패턴을 겹쳐 훈련을 시키는 방법을 사용한 경우에는 총 962번의 시간 스텝으로 앞절의 정리 3을 확인할 수 있다. 결과적으로 시스톨릭 배열을 이용한 병렬 수행이 기존의 순차 수행의 경우보다 약 [입력 패턴의 개수 $\times\sum_{k=1}^{(L-1)}(L_k\times L_{k+1})$] 스텝을 줄일 수 있다. 따라서 역전파 알고리즘을 제안된 방법으로 수행시켰을 때 수행시간을 반으로 줄일 수 있다.

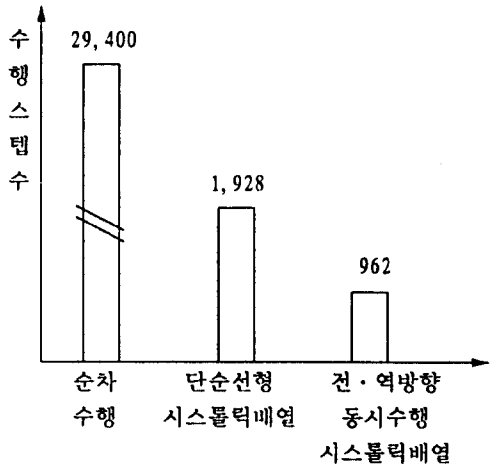


그림 9. 역전파 알고리즘을 순차 수행, 단순 선형 시스톨릭 배열, 전방향, 역방향 동시 수행 시스톨릭 배열에서 수행할 때의 시간 비교

Fig. 9. The time comparison of sequential, simple linear Systolic Array, and simultaneous forward-backward Systolic Array processing.

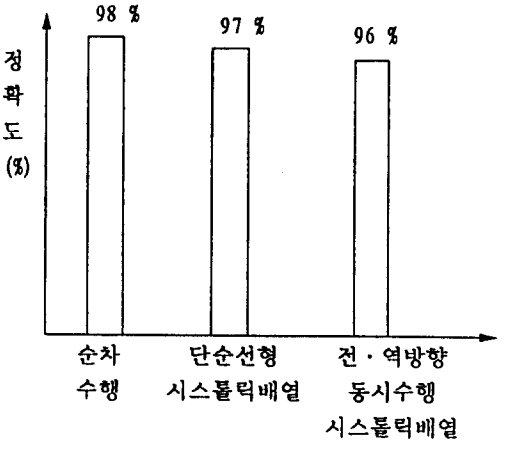


그림 10. 역전파 알고리즘을 순차 수행, 단순 선형 시스톨릭 배열, 전방향, 역방향 동시 수행 시스톨릭 배열에서 학습한 후의 정확도 비교

Fig. 10. Accuracy comparison after training of sequential, simple linear Systolic Array, and simultaneous forward-backward Systolic Array processing.

다음, 위에서와 같이 훈련용 자료 집합에서 판단 정

확도가 95 %가 될 때까지 반복 훈련을 시킨 뒤 성능 검사용 자료를 이용한 학습의 정확도를 비교해보면 그림 10과 같다. 결과적으로 역전파 알고리즘을 전방향, 역방향 동시 수행을 위한 선형 시스톨릭 배열로 설계했을 때, 각각의 PE 구조는 더 복잡해지지만, 판단 정확도를 떨어뜨리지 않고도 수행 시간을 순차 수행의 경우보다 1/29, 단순 선형 시스톨릭 배열의 경우보다 2배의 시간을 단축할 수 있다.

Ⅳ. 결 론

이 논문에서는 역전파 알고리즘의 전방향, 역방향 연산을 동시에 수행시키는 시스톨릭 배열을 설계하였다. 이를 위하여 역전파 알고리즘의 자료 흐름을 분석하여 전방향, 역방향 연산이 동시에 수행 가능한지를 검증한 후 이를 전방향, 역방향 동시 수행을 위한 DG로 나타내었다. 그리고 [16] 논문에서 제시한 방법을 이용하여 DG의 각 층에 서로 수직 방향의 공간 변환을 교대로 행하여 뱀 모양의 선형 시스톨릭 배열을 설계하였다. 다음으로 병렬 수행이 가능한 학습 주기를 계산하고, 전방향, 역방향 연산이 한 PE내에서 동시에 수행될 때에도 정확한 자료들이 정확한 위치에 사용되게하기 위하여 인덱스를 계산하며, 두 연산이 동시에 수행되어도 올바른 결과를 산출하는지를 검증하였다. 이렇게 하므로 시스톨릭 배열을 선형화하여 PE의 개수를 최소화시키며 전방향, 역방향이 동시에 수행가능하게 하여 휴면 PE(idle PE)의 수를 줄여 수행 시간도 기존의 시스톨릭 배열들 보다 2배 감소 시켰다.

또한 이 방법은 PE의 수를 최소화하면서도 역전파 알고리즘에 내재하는 병렬성을 극대화하여 수행 시간을 최소화하였다. 한편 l층의 뉴런의 개수가 L_l인 L층의 다 단계 신경망의 역전파 알고리즘을 2차원 시스톨릭 배열로 설계할 경우, PE의 개수가 $O(L\times L_l\times L_{l-1})$이지만 여기서 설계된 시스톨릭 배열의 PE의 개수는 $O(L\times L_l)$이다. 시뮬레이션을 통해서 설계된 시스톨릭 배열의 수행을 시간 스텝 별로 검증한 결과, 설계된 시스톨릭 배열 상에서 역전파 알고리즘의 전방향과 역방향의 연산이 바르게 수행됨을 확인하였다. 이 논문에서 제안된 역전파 신경망과 같은 유형의 인공 신경망을 시스톨릭 배열로 체계적으로 설계할 수 있는 방법은 또 다른 인공 신경망에 적용하여 그 신경망이 구현되는 시스톨릭 배열을 쉽게 설계할 수 있을 것이다.

앞으로 보다 증가된 PE 개수를 처리할 수 있는 체계적 설계 방법과 그에 따른 적용 범위에 대한 연구가 계속되어야 하며, 또한 PE 개수의 증가에 따라 늘어나는 휴면 PE를 더욱더 줄여 모든 PE들이 연산에 참여할 수 있도록 하는 연구가 계속되어야 할 것이다.

참 고 문 헌

[1] Chinn, G., et al., "Systolic array implementations of neural nets on the MasPar MP-1 massively parallel processor," *International Joint Conference on Neural Networks*, San Diego, Vol. 2, pp. 169-173, 1990.

[2] Chung, J. H., Yoon, H., and Meang, S. R., "A Systolic Array Exploiting the Inherent Parallelisms of Artificial Neural Networks," *Proc. of the International Conference on Parallel Processing*, St. Charles, Illinois, Vol I, pp. 652-653, 1991.

[3] Faure, B. and Mazare, G., "Implementation of back-propagation on a VLSI asynchronous cellular architecture," *International Neural Networks Conference*, Paris, Vol. II, pp. 631-634, 1990.

[4] Fisher, D., McKusick, K., Mooney, R., Shavlik, J., and Towell, G., "Processing Issues in Comparisons of Symbolic and Connectionist Learning Systems," *Processing of the Sixth International Machine Learning Workshop,* Ithica, N.Y.: Morgan Kaufmann, 1989.

[5] Kung, H. T., "Why systolic architecture ?," *IEEE Computer*, Vol. 15, No. 1, pp. 37-48, 1978.

[6] Kung, S. Y., *VLSI Array Processors*, Prentice-Hall, 1987.

[7] Kung, S. Y., "Parallel architectures for artificial neural nets," *International Conference on Systolic Arrays,* San Diego, CA., pp. 163-174, 1988.

[8] Kung, S. Y.,and Hwang, J. N., "A unified systolic architecture for artificial neural networks," *J. of Parallel and Distrib.*

Comput., No. 6, 1989.

[9] Kung, S. Y. and Hwang, J. N., "Parallel architectures for artificial neural nets," *International Conference on Neural Networks*, San Diego, CA., Vol 2, pp. 165-172, 1988.

[10] Lippman, R. P., "An introduction to computing with neural nets," *IEEE Trans. Acoustics Speech Signal Process*, Mag. 4 pp. 4-22, 1987.

[11] Mann, R., and Haykin, S., "A parallel implementation of Kohonen feature maps on the Wrap systolic computer," *Intenational Joint Conference on Neural Networks*, Washington, D.C., Vol. II, pp. 84-87, 1990.

[12] Petkov, N., "Systolic simulation of multilayer feedforward neural networks," *Parallel Processing in Neural Systems and Computers*, pp. 303-306, 1990.

[13] Ramacher, U., and Wesseling, M., "Systolic synthesis of neural networks," *International Neural Network Conference*, Paris, France, Vol. 2, pp. 572-576, 1990.

[14] Rumelhart, D. E., and McClelland, J. L., *Explorations in Parallel Distributed Processing*, MIT Press, Cambridge, MA. 1988.

[15] Tomboulian, S., "Introduction to a system for implementing neural net connections on SIMD architectures," *Neural Information Processing Systems*, Denver, CO., pp. 804-813, 1987.

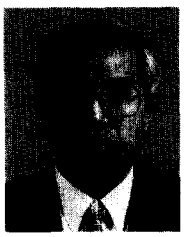
[16] 장 명숙, 유 기영, "역전파 신경망을 위한 시스톨릭 배열의 설계", 한국 정보 과학회 논문지, Vol. 24, No. 4, pp. 201-212, 1995

[17] 정 재훈, 윤 현수, 맹 승렬, "역전파 신경망 모델을 위한 시스톨릭 배열의 설계 및 성능 비교", 한국 정보 과학회 논문지, Vol. 19, No. 4, pp. 401-408, 1992

저 자 소 개

張 明 淑(正會員)

1979년 2월 경북대학교 사범대학 수학교육학과 학사. 1990년 5월 미국 Vanderbilt 대학교 전자계산학과 석사. 1992년 3월 ~ 현재 경북대학교 컴퓨터 공학과 박사과정. 1995년 9월 ~ 현재 대구 태창종합기술연구소 선임연구원. 관심분야는 인공 신경망, 병렬처리



柳 基 永(正會員)

1976년 경북대학교 수학교육학과 졸업(이학사). 1978년 한국과학기술원 전산학과 졸업(공학석사). 1992년 미국 Rensselaer Polytechnic Institute 졸업(이학박사). 1978년 ~ 현재 경북대학교 컴퓨터공학과에 재직. 관심분야는 병렬처리, DSP array processor설계, 병렬컴파일러 등임