

역전파 ANN의 시스톨릭 어레이를 위한 시뮬레이터 개발

(Systolic Array Simulator Construction for the Back-propagation ANN)

박 기 현* 전 상 윤**
(Kee-Hyun Park,) (Sang-Yoon Jeon)

요 약 시스톨릭 어레이는 간단한 연산능력을 가진 처리요소들이 정규적이고 국부적인 통신선들로 연결된 병렬처리 시스템이다. 시스톨릭 어레이는 인공신경망에서 고밀하게 연결된 뉴런으로 인하여 발생하는 뉴런 간의 복잡한 통신 문제를 해결하는 가장 좋은 방법 중의 하나로 알려져 있다. 본 논문에서는 주어진 뉴런 수에 적합한 역전파 인공신경망을 자동으로 생성하는 시스톨릭 어레이 시뮬레이터를 설계하고 구현한다. 시뮬레이터의 애니메이션 기법을 이용하여, 설계된 시스톨릭 어레이 상에서의 역전파 알고리즘의 실행 상황을 사용자가 단계별로 쉽게 관찰할 수 있다. 또한, 시뮬레이터는 역전파 알고리즘의 전 방향, 역 방향 연산을 각각 따로 실행시키거나, 병렬로 실행하게 할 수 있다. 병렬 실행은 입력 자료를 연속적으로 입력받아 시스톨릭 어레이의 모든 처리요소들에서 역전파 알고리즘의 양방향 전파를 동시에 실행시킴으로써 가능하다.

Abstract A systolic array is a parallel processing system which consists of processing elements of basic computation capabilities, connected with regular and local communication lines. It has been known that a systolic array is one of effective systems to solve complicated communication problems occurred between densely connected neurons on ANN(Artificial Neural Network). In this paper, a systolic array simulator for the back-propagation ANN, which automatically constructs the proper systolic array for a given number of neurons of the ANN, is designed and constructed. With animation techniques of the simulators, it is easy for users to be able to examine the execution of the back-propagation algorithm on the designed systolic array step by step. Moreover the simulator can perform forward and backward operations of the back-propagation algorithm either in sequence or in parallel on the designed systolic array. Parallel execution can be performed by feeding continuous input patterns and by executing bidirectional propagations on all of processing elements of a systolic array at the same time.

1. 서 론

인공신경망(ANN, Artificial Neural Network)은 패턴 인식, 분류 문제, 자연어 처리 등의 분야에서 좋은 결과를 산출하므로 실생활에 많이 이용된다[1-6]. 여러 ANN 모델들 중에서 역전파 알고리즘[7]을 이용하는 다층 퍼셉트론(역전파 ANN)은 가장 많이 이용되는 모델이다. 그러나 역전파 알고리즘은 수많은 뉴런에서의 계산으로 인하여 엄청난

시간이 소요되는 단점이 있다. 그래서 역전파 알고리즘의 병렬성을 이용한 병렬처리에 대한 연구가 이루어졌으며, 이런 노력의 일환으로서 많은 연구자들이 역전파 알고리즘을 병렬처리할 수 있는 시스톨릭 어레이에 대한 연구를 계속하였다[1, 2, 8-12].

시스톨릭 어레이는 간단한 연산능력을 가진 여러 개의 처리요소(Processing Element, PE)들이 정규적이고 국부적인 통신선들로 연결된 병렬 처리기이다[11,12]. 시스톨릭 어레이는 계산위주의 문제를 위한 특수 목적의 보조 처리기로써, ANN에서 고밀하게 연결된 뉴런으로 인하여 발생하는 뉴런간의 복잡한 통신 문제를 해결하는 가장 좋은 방법 중의 하나이다. 또한 시스톨릭 어레이는 많은 병렬성

*계명대학교 컴퓨터 및 전자공학부 교수

**계명대학교 컴퓨터 및 전자공학부 석사과정

과 파이프라이닝을 이용할 수 있으며, VLSI 구현에도 적합한 구조를 갖는다.

본 논문에서는 역전파 ANN을 시스틀릭 어레이로 설계하고 그 구조와 실행 상태를 볼 수 있게 하는 역전파 시스틀릭 어레이 시뮬레이터를 설계하고 구현한다. 시뮬레이터에서는 마우스를 이용하여 역전파 알고리즘의 동작을 단계별로 컴퓨터 화면상에서 자동으로 혹은 수동으로 애니메이션함으로써, 설계된 시스틀릭 어레이 및 역전파 알고리즘의 수행 상황을 쉽게 관찰할 수 있도록 한다. 시뮬레이터는 역전파 알고리즘의 전 방향, 역 방향 연산을 각각 따로 수행시키거나, 입력 자료를 연속적으로 입력받아 모든 PE들에서 역전파 알고리즘의 양방향 전파를 동시에 병렬 수행시킬 수 있다.

논문의 구성은, 2절에서는 이 논문에 대한 배경 설명으로서 역전파 알고리즘의 식들에 대하여 소개하고, 3절에서는 역전파 알고리즘의 자료 흐름을 분석하여 양방향 전파의 병렬처리를 수행하는 시스틀릭 어레이를 설계하는 과정을 소개한다. 4절에서는 3절에서 제시한 시스틀릭 어레이 시뮬레이터의 설계와 수행상태를 보여준다. 5절에서는 이 논문에 대한 결론 및 추후 연구 방향에 대하여 논한다.

2. 역전파 알고리즘

계층 1에서 계층 L 까지의 다단계 신경망 모델은 계층 1의 입력 계층(input layer), 계층 2에서 계층 $(L-1)$ 까지 $(L-2)$ 개의 은닉 계층(hidden layer)들, 그리고 계층 L 의 출력 계층(output layer)으로 이루어져 있다. 각 층은 인간의 신경 절에 해당하는 뉴런들로 구성되어 있으며, 이웃한 층의 뉴런들은 가중치를 가진 연결선들로 서로 연결되어 있다.

어느 한 입력패턴 I_p 에 대한 전방향 연산은, 입력 I_p 의 각 원소 $a_{i,p}^1$ 을 받아들여 각 층에서 활성 값을 계산하여 출력을 구하는 단계로, $(l-1)$ 층의 뉴런의 개수가 L_{l-1} , l 층의 뉴런의 개수가 L_l 일 때, 아래의 식 (1)과 (2)로 나타낼 수 있다.

(1) 총 입력 값 계산 :

$$u_{i,p}^l = \sum_{j=1}^{L_{l-1}} w_{ij}^l a_{j,p}^{l-1} + \theta_{i,p}^l$$

(2) 활성 값 계산 :

$$a_{i,p}^l = f(u_{i,p}^l) = 1/(1 - \exp(-u_{i,p}^l))$$

역 방향 연산은, 먼저 출력 층의 각 뉴런에 대한 오차

$(t_{i,p} - a_{i,p}^L)$ 이 구해진 다음, 이 오차는 아래 식과 같이 출력 층에 있는 모든 뉴런 i 에 대하여 제공하여 더해진다.

(3) 총 오차 계산 :

$$E_p = \sum_{i=1}^{L_i} (t_{i,p} - a_{i,p}^L)^2$$

역전파 알고리즘의 학습 목표는 이 오차 값의 평균을 최소화하는 것이다. 이것은 먼저 출력 층의 각 뉴런 i 에 대하여 식 (1)을 1차 미분한 뒤 오차에 대한 감소치(error gradient) δ_i^l ($l=1, \dots, L$)을 아래의 식 (4)와 같이 구한다.

(4) 오차의 감소치 계산

만약 $l=L$ 일 때

$$\delta_{i,p}^L = (t_{i,p} - a_{i,p}^L) f'(u_{i,p}^L)$$

만약 $1 \leq l < L$ 일 때

$$\delta_{i,p}^l = \left(\sum_{j=1}^{L_{l+1}} \delta_{j,p}^{l+1} w_{ji}^{l+1} \right) f'(u_{i,p}^l)$$

이 오차의 감소치는 역 방향으로 신경망의 각 층을 따라 전파되며 식 (5)와 같이 각 층의 가중치를 변경시킨다.

(5) 연결선의 가중치 변경 :

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

여기서 η 는 사용자가 정의한 매개변수인 학습 비율(learning rate)이다. 위의 과정은 모든 입력패턴 I 에 대하여 반복되며, 식 (3)의 오차 값이, 역시 사용자가 정의해주는 종료 조건보다 작아지면 학습을 멈춘다.

3. 양방향 처리의 시스틀릭 어레이

시스틀릭 어레이를 설계하기 위하여, 전 방향, 역 방향 식들로부터 자료 흐름을 분석한다. 앞 절의 식 (1), (2)를 살펴보면 활성값 $a_{i,p}^l$ 는, $[i, j]$ 축을 기준으로 하였을 때, $[1, 0]$ 방향으로 흐르고, 총 입력 값 $u_{i,p}^l$ 와 바이어스 값 $\theta_{i,p}^l$ 는 1-2 층 사이에서는 $[0, 1]$ 방향으로, 2-3 층 사이에서는 $[1, 0]$ 방향으로 흐른다. 식 (4), (5)에서 오차의 감소치 δ_i^l 는 2-3 층 사이에서는 $[0, -1]$ 방향으로 흐르고, 1-2 층 사이에서는 $[-1, 0]$ 방향으로 흐른다. 이 분석을 토대로 자료의 흐름 그래프(Data dependency Graph, DG)를 구할 수 있다.

다음은 DG에 공간-시간 변환(space-time transformation)을 적용한다. DG의 노드들 중, 동시에 연

산이 이루어지는 노드들을 선으로 연결한 것을 동시간 초평면(equitemporal hyperplane)이라 한다. DG 상에서 모든 자료 흐름은 이 동시간 초평면을 기준으로 같은 방향의 흐름을 가져야 하며, 이는 같은 초평면상의 노드들의 동시 수행을 가능하게 한다. 그러므로 같은 초평면상의 노드들을 시스톨릭 어레이의 서로 다른 PE로 사상시키면 병렬처리가 가능해진다. 동시간 초평면을 이용하여 이에 수직인 시간 벡터 $\vec{t}=[1,1]^T$ 를 구할 수 있다. 다음, 공간벡터 s 는 동시간 초평면과 평행하면 안 된다. 즉, DG 상에서 동시에 수행되는 노드들이 같은 PE로 사상되면 병렬처리가 불가능하다. 공간 변환 $\vec{s}_1=[1,0]^T$ 와 $\vec{s}_2=[0,1]^T$ 를 각 층에 교대로 적용하면 시스톨릭 어레이를 얻을 수 있다. 그림 1은 병렬처리를 위한 시스톨릭 어레이와 각 층의 PE들의 구조를 나타낸 것이다[2]. 이 때, (1-2 층)에 대한 PE는 공간 변환 $\vec{s}_1=[1,0]^T$ 를, (2-3 층)에 대한 PE는 공간 변환 $\vec{s}_2=[0,1]^T$ 를 교대로 적용한 것이다.

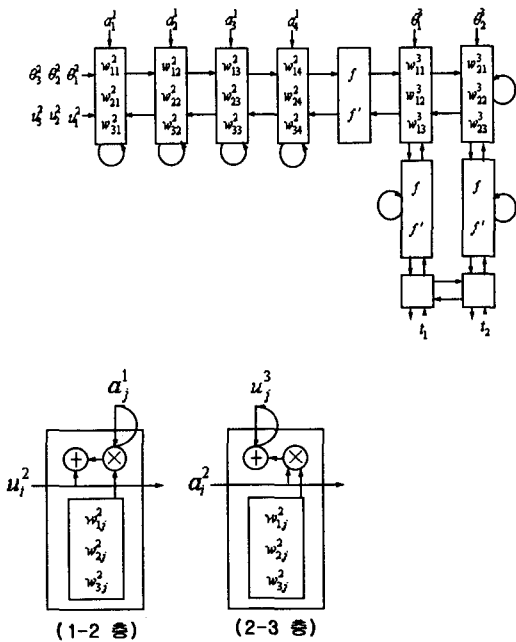


그림 1. 4-3-2 역전과 ANN에 대한 시스톨릭 어레이와 각 층에 대한 PE 구조

4. 역전과 알고리즘을 연속적으로 병렬 처리하는 시스톨릭 어레이 시뮬레이터 설계

이 절에서는 앞 절에서 설명한 역전과 알고리즘을 위한 시스톨릭 어레이의 구조와 동작을 시뮬레이션하는 시뮬레이터를 설계, 구현한다. 시뮬레이터는 Visual C++를 사용하여 구현하였다. Visual C++는 객체 지향적인 언어이기 때문에 시스톨릭 어레이와 각각의 요소들의 동작을 구현하는데 적절하다. 또한 사용자 인터페이스를 위한 풍부한 객체들을 제공하기 때문에 시각적인 표현에서도 적합하다. 시뮬레이터는 그림 2와 같은 단계로 이루어진다. 먼저, 초기 설정 자료로부터 정보를 읽어서 PE들의 어레이를 생성하고, 다음, 구성된 PE들의 어레이를 기반으로 자료를 읽고 학습과 처리한다.

시뮬레이터는 크게 초기 설정, 입력, 출력, PE, 가중치에 대한 객체들로 나누어진다. PE와 가중치는 초기 설정의 변화에 따라 동적으로 생성되고 소멸되도록 설계되었으며, 훈련이나 자료의 처리는 thread를 invoke해서 한다. 훈련과정은 자료의 처리에 대한 출력은 없고 가중치만을 변경하는 과정이며, 처리과정에서는 훈련과정에서 변경된 가중치를 기반으로 해서 출력을 얻어내는 과정이다. 모든 과정을 통해서 수행중인 PE의 내부에서 어떤 자료가 입력되고 출력되는지를 대화상자를 통해서 볼 수 있게 하였으며, 시각적으로 자료의 흐름을 볼 수 있도록 구현함으로써 시각적인 시스톨릭 어레이를 구현하고자 하였다. 또한 초기 설정을 다시 설정함으로써 다른 입·출력형태를 가지는 여러 가지의 모델을 동시에 처리할 수 있도록 하였다.

4.1 전처리 단계

전처리 단계에서는 먼저 초기 설정 자료를 읽어 화면에 역전과 ANN을 위한 시스톨릭 어레이를 생성하고 다음 단계인 시뮬레이션을 수행하기 위한 각종 버튼들을 화면에 생성해 준다. 초기 설정 파일로부터 다층 퍼셉트론의 입력, 은닉, 출력층의 구조에 대한 자료를 읽어들이어서 화면에 그리드를 만든 후 초기 설정 객체를 생성해서 자료의 처리에 필요한 층의 개수와 각 층에 대한 PE의 개수, 입력, 출력 층의 각 열에 대한 이름을 저장시킨다. 이것을 바탕으로 PE를 생성한다.

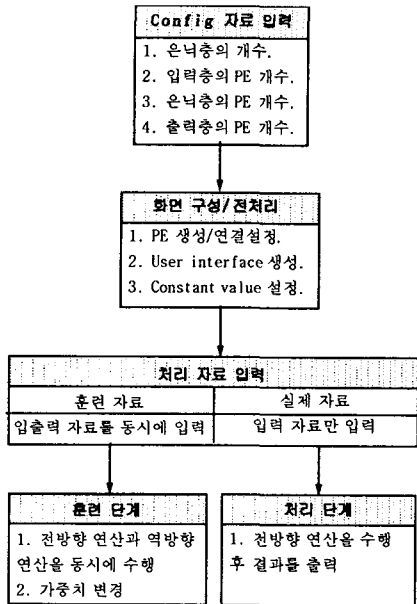


그림 2. 시뮬레이터의 전체적인 수행 구조

생성된 PE는 상하좌우의 PE들과 연결선이 설정되어서 상호 자료를 주고받을 수 있도록 한다. 홀수 번째 층의 PE들은 전 방향, 역 방향의 연결선을 가지고, 짝수 번째 층의 PE들은 전 방향, 역 방향, 아래, 위의 연결선을 가진다. 각각의 PE는 역전파 알고리즘의 전 방향 전파와 역방향 전파를 동시에 수행할 수 있도록 설계되었다. 그리고 각 PE 내의 가중치 값들은 0으로 초기화 되어있다. 다음, 각각의 PE가 위치한 층과 열에 따라 PE가 수행해야 할 일이 할당된다.

초기 설정은 언제든지 새로운 것으로 설정이 가능한데, 만약에 현재 시뮬레이터가 자료를 처리중일 경우에는 재설정이 불가능하다. 자료를 처리중에 새로운 초기 설정을 하고자 할 경우에는 현재 진행중인 작업을 멈추고 새로운 초기 설정 자료를 읽어들이어야 한다. 훈련단계에서는 양방향의 연결선이 동시에 사용되어서 가중치를 변화시키게 되고, 실제 자료의 처리에서는 훈련단계에서 구성된 가중치를 가지고 전방향 전파만을 거친 후, 결과를 출력하게 된다. 그림 3은 3-2-2-1 ANN을 위한 시스틀릭 어레이 시뮬레이터의 초기화면을 보여준다. 초기 화면에서는 인공신경망의 각 층의 노드의 개수가 주어지면 자동으로 그림과 같은 시스틀릭 어레이가 설계된다. 그림에서 네모는 각 층의 PE들을 나타내고 각각의 PE들은 전방향, 역방향의

연결선들로 연결되어 있다.

4.2 화면 구성

4.2.1 Start Simulation

다층 퍼셉트론의 입력, 은닉, 출력 층에 관한 속성을 기술해 놓은 파일(*.cfg)로부터 초기 설정 자료를 읽어들이어서 PE를 생성하고 각각의 PE들을 적절한 연결선들로 연결시켜주는 것이다. 훈련자료나 실제 처리할 자료를 읽어 오기 전에 항상 초기 설정이 되어있어야 한다.

4.2.2 Terminate Simulation

현재 처리중인 자료의 처리를 강제로 종료시킨다. 이전에 처리된 자료에는 영향을 미치지 않으며 훈련자료를 처리하는 도중이었을 경우에는 최종적으로 변경된 가중치 값이 유지된다. 새로운 시뮬레이션을 설정하고자 할 경우에는 우선 이것을 실행한 후 현재 진행중인 작업을 멈춘 다음에 "Start Simulation"을 실행해야 한다.

4.2.3 Run Training

훈련자료를 읽어와서 가중치의 값을 변경시키고자 할 경우에 사용된다. 초기 설정이 처음으로 설정되었을 경우에는 모든 가중치의 값이 0이므로 실제 처리하고자 하는 자료를 처리하기 이전에 이 과정을 통해 시뮬레이터를 훈련시켜서 가중치 값을 변경시켜 주어야 한다.

4.2.4 Run Processing

충분한 훈련 자료를 통해 시뮬레이터의 가중치가 적절하게 들어갈 수 있도록 한 후, 실제 처리하고자 하는 자료를 읽어와서 처리하는 명령이다. 여기서는 훈련과정을 통해 얻어진 가중치를 이용해서 결과를 얻어내고, 가중치를 변경시키지는 않는다.

4.2.5 Interleaving

입력 레코드간의 입력 간격을 조정하기 위한 control 이다. 한 레코드의 마지막 열의 값이 입력된 후 다음 clock 에서 바로 입력이 이루어지는 경우 Interleaving value는 0 이다. Interleaving value는 항상 0 이상이어야 하며 음수는 가질 수가 없다.

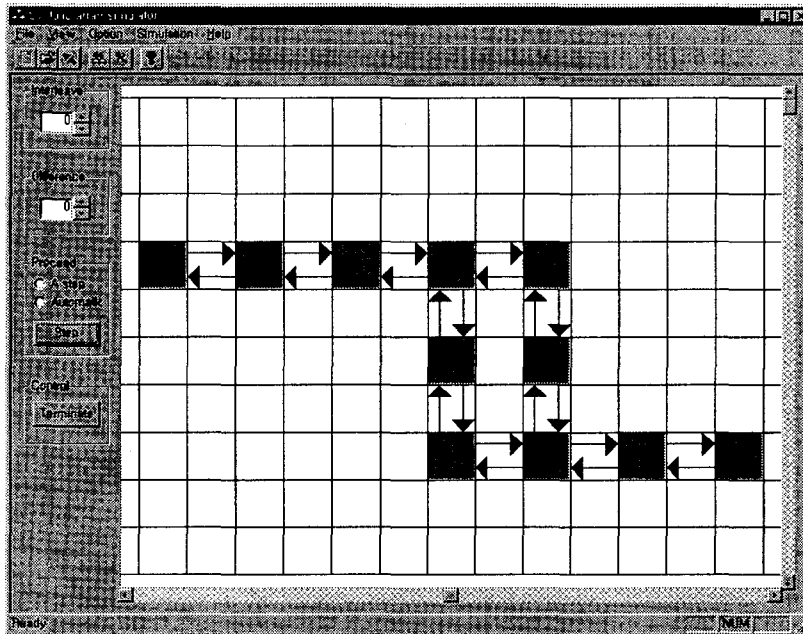


그림 3. 3-2-2-2-1 ANN을 위한 시스틀릭 어레이 시뮬레이터의 초기화면

4.2.6 Difference

허용 가능한 최저 평균 오차 값을 지정할 수 있도록 해주는 control이다. 이 control의 변경 폭은 0.1로 설정되어 있으며 처리할 시뮬레이션에 따라 이 값을 바꾸어 줌으로써 적당한 최저 평균 오차 값을 지정해줄 수 있다.

4.2.7 Processing Method

Processing method에는 STEP_BY_STEP mode와 AUTOMATIC mode의 두 가지가 있다. STEP_BY_STEP mode는 사용자가 시뮬레이터의 각각의 clock을 Step button을 통해서 진행시키는 방법이고, AUTOMATIC mode는 clock의 처리가 자동으로 이루어지도록 하는 것이다. 두 가지 방법에서 수행되어지는 모든 계산 과정이나 작업은 같지만 STEP_BY_STEP mode의 경우에는 각 단계마다 PE의 상태를 시각적으로 확인할 수 있고, 활성화된 PE의 입출력 값도 알아볼 수가 있다.

4.2.8 View Panel

자료의 처리 상황을 나타내기 시각적으로 보여주기 위한 창이다. 여기서는 PE들의 연결상태와 PE들이 처리 중인 일을 SLEEP, FORWARD, BACKWARD로 구분해서 각각 다른 색깔과 크기로 나타내고 있다. SLEEP중인 PE

를 제외한 FORWARD, BACKWARD 작업을 하고 있는 PE에 대해서는 현재 처리 중인 자료의 내용을 같이 볼 수 있게 되어 있다.

4.2.9 PE Color

시각적으로 나타나는 PE의 움직임은 사용자가 확인하기 쉬운 색깔로 지정할 수 있도록 하기 위한 명령이다. 시각적인 구분을 위해서 전 방향, 역 방향 연산 시 각각 다른 색깔을 사용함과 동시에 도형의 크기를 다르게 해서 확연히 구분이 가능하도록 하였다.

4.3 시뮬레이션 단계

시뮬레이션 단계는 훈련자료를 통해서 시뮬레이터를 훈련시키는 단계와 처리할 자료를 통해서 결과값을 얻어내는 과정이 모두 포함된다. 이 두 자료의 차이점은 훈련 자료는 입력 값에 출력 값도 같이 있어서 시뮬레이터를 훈련 시키는데 이용한다는 것이다. 처리단계에서는 진행방법을 STEP_BY_STEP mode, AUTOMATIC mode 중의 한 가지 방법으로 할 수가 있으며, 자료 처리 중에도 바뀌가면서 진행시킬 수가 있다. 시뮬레이터의 동작은 아래와 같다.

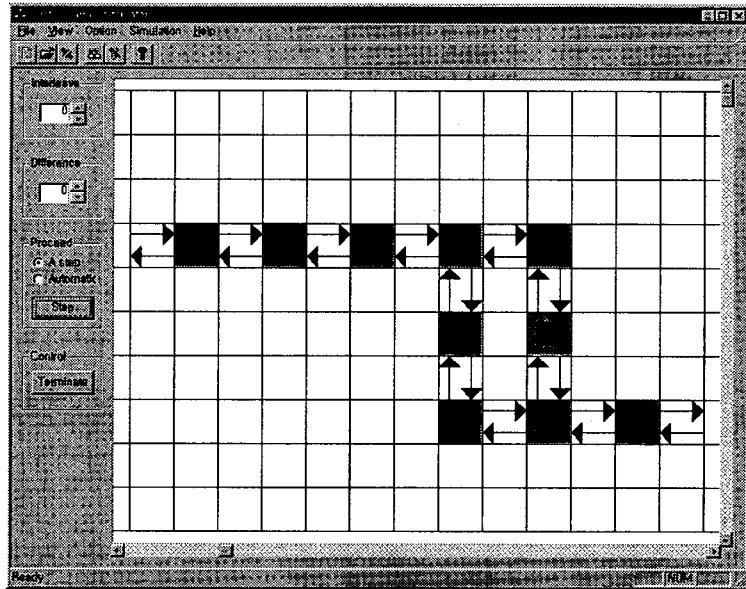


그림 4. 3-2-2-2-1 ANN을 위한 시스틀릭 어레이 시뮬레이터의 초기 설정 후의 화면

4.3.1 초기 설정

Config 객체를 생성해서 자료의 처리에 필요한 층의 개수와 각 층에 대한 PE의 개수, 입력, 출력 층의 각 열에 대한 이름을 저장시키고, 이것을 바탕으로 가중치와 PE를 생성한다. 생성된 PE는 상하좌우의 PE들과 연결선이 설정되어서 자료를 서로 주고받을 수 있도록 하였다. 이것이 이루어진 후 각각의 PE가 위치한 층과 열에 따라 PE가 수행해야 할 일이 할당된다. 그리고 초기 설정은 언제든지 새로운 것으로 설정이 가능한데, 만약에 현재 시뮬레이터가 현재 자료를 처리중일 경우에는 재 설정이 불가능하다. 자료를 처리 중에 새로운 초기 설정으로 설정하고자 할 경우에는 현재 진행중인 작업을 멈추고 새로운 config 파일을 load해야 한다.

4.3.2 실행화면

그림 4는 3-2-2-2-1 ANN을 위한 시스틀릭 시뮬레이터가 훈련자료를 STEP_BY_STEP mode로 처리하고 있는 과정을 나타내고 있는 화면이다. 그림에서 서로 다른 입력에 대한 연산을 수행하는 PE들은 서로 다른 PE 색으로 나타난다. 그리고 서로 다른 입력에 대한 전방향, 역방향의 연산이 겹치는 PE들은 두 가지 색으로 PE 상태를

나타내었다. 이 상태에서 활성화되어 있는 PE에 대해서는 현재 수행중인 작업의 내용을 관찰할 수가 있으며, 만약에 처리과정을 관찰하지 않고 자료를 빨리 처리하고자 한다면, AUTOMATIC mode를 선택하면 된다. 이 경우에는 PE의 상태를 갱신하지 않고 자료를 처리한다.

4.3.3 PE 상태 창

그림 5는 역전파 알고리즘의 전 방향, 역 방향 연산을 동시에 처리중인 PE의 상태를 보여주는 창이다. 이 창의 상태는 Forward 상태, Backward 상태, 그리고 Weight 상태로 나누어진다. Forward 상태에서는 하나의 PE를 클릭하면 현재 처리되고 있는 총 입력 값과 Weight의 곱을 보여준다. 총 입력 값은 활성 값으로 계산된 후 다음 계층의 PE로 전달된다.

Backward 상태에서는 오차의 감소치를 계산하기 위해 누적된 하위 층의 오차 감소치와 Weight의 곱을 보여준다. 이렇게 누적된 값은 역방향으로 전파되면서 다음 층에 대한 오차의 감소치가 계산된 후 전달된다. Weight 상태에서는 변경시킨 Weight의 위치와 값을 보여준다.

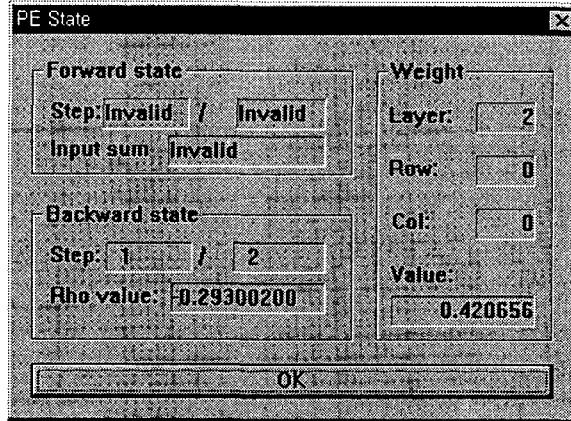


그림 5. 3-2-2-2-1 ANN을 위한 시스틀릭 어레이 시뮬레이터의 PE 상태를 나타내는 창

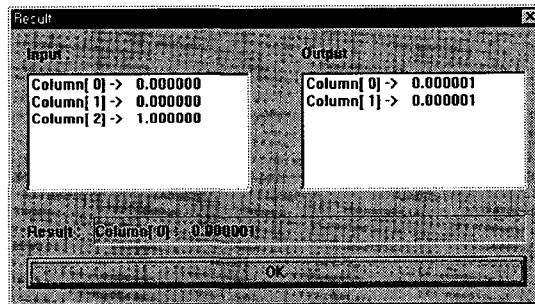


그림 6. 3-2-2-2-1 ANN을 위한 시스틀릭 어레이 시뮬레이터의 수행 결과를 나타내는 창

4.2.3 Result 창

그림 6은 시뮬레이터가 훈련과정을 거친 후, 처리할 record의 Forward 처리가 끝나면 그것의 결과를 보여주는 창이다. Input list box에서는 결과를 계산하기 위해 사용된 입력 값들을 보여주고, Output list box에서는 처리된 후의 마지막 층에서 계산된 값을 보여준다. Output list box를 정렬한 후 최대값을 찾아내서 그것의 Column과 값을 Result 난에 나타낸다.

5. 결 론

본 논문에서는 주어진 역전파 ANN으로부터 자동으로 시스틀릭 어레이를 설계하여 화면상에 PE와 연결선의 구조를 그래픽으로 나타내고, 시스틀릭 어레이 상에서 역전

파 알고리즘의 수행 상태를 시각적으로 보여주는 역전파 신경망에 대한 시스틀릭 어레이 시뮬레이터를 제작하였다. 이를 위하여 역전파 알고리즘의 자료 흐름을 분석하고, DG로 나타낸 뒤, DG에 효율적인 공간-시간 변환을 적용하였다. 그리고 역전파 ANN의 노드 개수와 시스틀릭 어레이의 PE 개수와의 관계식을 구하여 대수적인 방법으로 시스틀릭 어레이 시뮬레이터를 설계하였다.

본 논문에서 제작한 시뮬레이터에서는 역전파 ANN의 동작을 단계별로 컴퓨터 화면상에서 자동으로 혹은 수동으로 애니메이션함으로써, 설계된 시스틀릭 어레이 및 역전파 알고리즘의 수행 상황을 쉽게 관찰할 수 있도록 하였다. 시뮬레이터는 역전파 알고리즘의 전 방향, 역 방향 연산을 각각 따로 수행시키거나, 입력 자료를 연속적으로

입력받아 모든 PE들에서 역전파 알고리즘의 양방향 전파를 동시에 병렬 수행시킬 수 있도록 하였다.

참 고 문 헌

[1] 김지연, 장명숙, 박기현, "역전파 ANN을 위한 고정크기 시스틀릭 어레이 설계," 「한국정보과학회 가을학술논문발표논문집」, 25권, 2호, 1998년, pp. 691-693.

[2] 장명숙, 박기현, "교차공간변환을 이용한 역전파 알고리즘의 양방향 병렬처리 시스틀릭 어레이," 「정보과학회논문지」, 25권, 12호, 1998년 12월, pp. 1375-1383.

[3] Fahlman, S. E. Hinton G. E. and Sejnowski, T. J. "Massive-parallel Architectures for AI: NETL, THISTLE and Boltzmann Machines," National Conference on Artificial Intelligence, Washington, DC.(1983), pp. 109-113.

[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,(1978) pp. 37-48.

[6] Kung, H. T. Miller, R. E. and Winograd, S. "The Organization of Computations for Uniform Recurrence Equations," Journal of the ACM, Vol. 14(1967), pp. 563-590.

[7] Kung, S. Y. "Parallel Architectures for Artificial Neural Nets," International Conference on Systolic Arrays, San Diego, CA.(1988), pp. 163-174.

[8] Kung, S. Y. VLSI Array Processors, Prentice-Hall(1987).

[9] Kung, S. Y. and Hwang, J. N. "A Unified Systolic Architecture for Artificial Neural Networks," J. of Parallel and Distrib. Comput., No. 6(1989).

[10] Kung, S. Y. and Hwang, J. N. "Parallel Architectures for Artificial Neural Nets," International Conference on Neural Networks, San Diego, CA., Vol 2(1988), pp. 165-172.

[11] Schlimmer, J. and Fisher, D. "A Case Study of Incremental Concept Learning," Proceeding of the Fifth

National Conference on Artificial Intelligence, Philadelphia, PA., Morgan Kaufmann (1986).

[12] Tomboulia, S. "Introduction to a System for Implementing Neural Net Connections on SIMD Architectures," Neural Information Processing Systems, Denver, CO., (1987), pp. 804-813.

박 기 현



1979년 경북대학교 전자공학과 (컴퓨터 전공) 졸업(학사)

1981년 한국과학기술원 전자계산학과 졸업(석사)

1990년 미국 Vanderbilt 대학교 전자계산학과 졸업(박사)

1981년 - 현재 : 계명대학교 컴퓨터 및 전자공학부 교수

관심분야 : 병렬/분산처리 시스템, 운영체제, 성능분석 등

전 상 윤



1997년 계명대학교 도시공학과 졸업(학사)

1999년 계명대학교 전자계산학과 졸업(석사)

1999년 - 현재 : 한국통신 근무

관심분야 : 멀티에이전트 시스템, 시뮬레이션 등