

신경회로망을 이용한 Rank-Order 필터의 구현과 성능 평가

준희원 윤 숙*, 정희원 박 동 선*

Performance Evaluation and Implementation of Rank-Order Filter Using Neural Networks

Sook Yoon* *Associate Member* Dong-Sun Park* *Regular Member*

요 약

본 논문에서는 rank-order 필터의 구현을 위해 세 가지 신경회로망의 구조를 제시하고 분석하며 용도를 제안한다. 첫 번째 신경회로망을 이용하여 2-입력 정렬기를 제안하고 이를 이용하여 계층적인 N-입력 정렬기를 구성한다. 두번째로 입력 신호간의 상대적인 크기 정보를 이용하여 학습 패턴을 구성한 후 역전파 학습 기법을 이용하여 구현되는 순방향 신경회로망을 이용한 rank-order 필터를 구현한다. 세번째로 신경회로망의 구조의 출력층에 외부 입력으로 순위 정보를 가지도록 하는 rank-order 필터를 순방향 신경회로망을 이용하여 구현한다. 그리고 이러한 제안된 기술들에 대해 확장성, 구조의 복잡도와 시간 지연등에서의 성능을 비교, 평가한다. 2-입력 정렬기를 이용하는 방식은 확장이 용이하고 비교적 구조가 간단하나 입력 신호들의 정렬을 위해 신경회로망은 순환하는 구조를 가지며 입력 신호의 수에 비례하는 반복 연산 후에 결과를 얻게 된다. 반면에, 순방향 신경회로망을 이용한 rank-order 필터의 구현 방식은 이러한 반복 연산으로 인한 시간 지연을 줄일 수 있으나 상대적으로 복잡한 구조를 가진다.

ABSTRACT

In this paper, three neural network architectures are proposed for the implementation of the rank-order filter. Firstly, we propose a two-input neural sorter and then construct a N-input neural sorter with required number of modules of the two-input neural sorters. It is a recurrent neural network which completes sorting of input signals after a fixed number of iterations. An input signal at any rank can be chosen directly after sorting an array of input signals. Secondly, we propose a feed-forward neural network architecture which determines the position of an input signal with a given rank. The backpropagation learning rule is used to train the network with the training patterns generated from an operation which compares each pair of input signals for all possible cases of input signals. A magnitude-invariant feature is designed to obtain binary information of the relative magnitude of input signals in a given order. With this neural network, a rank-order filter can be implemented in a parallel structure and it performs the sorting process very fast without having any iterative operation. Thirdly, the rank-order filter with a rank input is implemented using a feed-forward neural network. The desired rank is controlled by the given bias. Three proposed structure are compared and analyzed. The recurrent N-input neural sorter has a simple structure and is flexible to construct due to the inherent modularity. However it requires a number of iterations to get the final outputs. The feedforward backpropagation network can greatly reduce the time delay but it needs more hardware than the recurrent sorter.

* 전북대학교 전자정보공학부, 정보통신연구소 (judy@iceng.chonbuk.ac.kr, dspark@moak.chonbuk.ac.kr)
논문번호: K01036-0127, 접수일자: 2001년 1월 27일

I. 서론

영상 처리를 포함한 많은 디지털 신호 처리 기술은 입력 신호상의 국부적인 이웃에 수행되는 공간상의 연산을 기본으로 한다. 신호는 공간적 마스크 (spatial mask)라 불리는 유한 임펄스 응답 필터 (finite impulse response filter)와의 중첩 연산을 통해 변화된다^[1]. 여러 영상 처리에서 이용되는 rank-order 필터는 비선형 필터로서, 주어진 신호들을 오름차순이나 내림차순으로 배열한 후 특정 순위의 신호를 선택한다. 예로서, 선택하는 순위에 따라 최대값 필터(maximum filter), 중앙값 필터 (median filter) 또는 최소값 필터(minimum filter)등으로 불리기도 한다. 정렬된 배열에서 중앙값을 선택하는 중앙값 필터는 디지털 영상 처리에서 임펄스 잡음을 제거하는데 이용되며 음성 처리등에서는 잡음을 가지는 유크를 평활화하는데 이용된다^{[1][2][4]}. 최대값 필터와 최소값 필터는 형태학적 연산에서 사용되는 팽창(dilation)과 수축(erosion)의 기본 연산자로서 사용된다^{[3][5][6][7]}.

이러한 rank-order 필터는 비선형적인 정렬 처리 단계를 요구하기 때문에 매우 높은 계산상의 복잡도를 요구한다. 그러므로 비선형 연산을 효율적이고 신속하게 처리할 수 있는 기술적 모델이 요구된다. 신경회로망이 병렬로 비선형 연산을 수행할 수 있다는 것은 널리 알려진 사실이다^{[8][9]}. 많은 다양한 응용 분야에서 신경회로망은 일반화 능력, 함수 근사화와 분류등과 같은 탁월한 특성을 보여왔다^{[10][11]}. 신경회로망의 활성화 함수와 구조등을 효과적으로 조절하여 rank-order 필터를 구현할 수 있다. 본 논문에서는 세 종류의 신경회로망을 이용한 rank-order 필터의 구현을 제안한다. 첫째는, 2-입력 정렬기를 설계하고 2-입력 정렬기를 기본 모듈로 사용하고 이 모듈들을 2단으로 구성하여 N-입력 모듈러 신경회로망 정렬기를 구현한다. N-입력 모듈러 신경회로망 정렬기는 전체적으로 순환 신경회로망 (recurrent neural network) 구조를 가지며 정해진 횟수의 반복을 통해 주어진 입력 신호들을 정렬하게 된다. 이때 N-입력 모듈러 신경회로망 정렬기의 반복횟수는 입력 신호의 개수에 의해 결정된다. 둘째로, 입력 신호간의 상대적인 크기 정보를 이용하여 학습 패턴을 구성한 후 역전파(backpropagation) 알고리즘을 이용한 학습을 통해 순방향 신경회로망 (feed-forward neural network) 구조의 rank-order

필터를 구현한다. 이 구조의 rank-order 필터는 이전의 N-입력 정렬기를 이용하는 것보다 상대적으로 빠른 시간 안에 특정 순위의 신호를 얻어낼 수 있다. 셋째로, 순방향 신경회로망 구조를 이용하여 어떤 임의의 순위의 신호라도 얻어낼 수 있는 rank-order 필터를 구현한다. 이때 원하는 순위정보는 신경회로망의 외부 입력(bias)을 통해 제어된다. 순방향 구조의 신경회로망을 이용하므로 빠른 응답을 보이거나 앞의 두 구조에 비하여 하드웨어가 복잡하다. 제시된 세 rank-order 필터에 대하여 구조와 성능을 비교 분석한다.

본 논문은 다음과 같이 구성된다. II장에서는 모듈러 신경망 정렬기에 대해 설명한다. 기본적인 모듈로서 2-입력 정렬기가 설계되고 N-입력으로서의 확장이 설명된다. 그리고 III장에서는 순방향 신경회로망을 이용한 rank-order 필터의 구현에 관한 기술을 언급한다. 실험과 이에 대한 분석, 그리고 검토가 IV장에서 보여지며 마지막으로 V장에서는 본 논문의 결론을 맺는다.

II. 모듈러 신경망 정렬기

2-입력 정렬기는 두 입력이 주어지면 정오된 순서로 이들을 정렬시켜 출력하는 시스템이다. 일반적으로 2-입력 정렬기는 그림 1에서 보여지는 바와 같이 하나의 비교기와 두 개의 2-to-1 다중화기로 구성될 수 있다. 또한 2-입력 정렬기는 간단하게 신경회로망 구조를 이용하여 구현할 수도 있다. 가중치(weight)와 입력간의 내적을 수행하고 비선형 활성화 함수 (nonlinear activation function)를 이용하는 신경회로망의 특성은 정렬기와 같은 비선형 연산자를 구현하는 작업을 쉽고 간단하게 만든다.

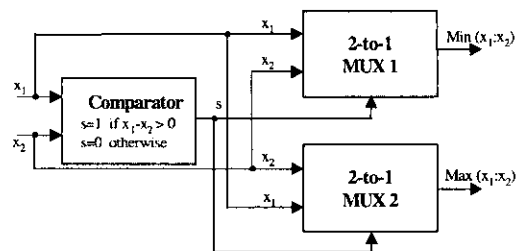
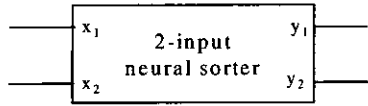


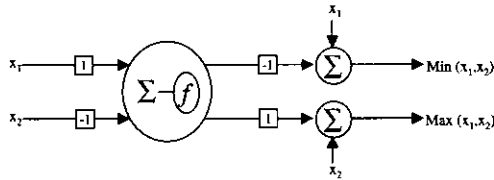
그림 1. 2-입력 정렬기

본 논문에서는, 2-입력 정렬기를 위한 가중치를 수학적 해석에 의한 방법을 통해 얻는다. 그림 2는

오름차순의 2-입력 정렬기를 위한 신경회로망 구조를 보여주고 있다.



(a) 블록도



(b) 신경회로망 구조

그림 2. 신경회로망을 이용한 2-입력 정렬기

가중치를 결정하기 위해 그림 1의 구조를 이용할 수 있다. 먼저 2-입력 비교기에서 입력 x_1 과 x_2 에 대해 출력 s 는 식 (1)과 같이 정의된다.

$$s = \begin{cases} 1 & \text{if } x_1 - x_2 > 0 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

다중화기의 출력은 식 (2)와 같은 관계를 보여준다.

$$\begin{aligned} \text{if } s=1 & \begin{cases} \min = x_2 \\ \max = x_1 \end{cases} \\ \text{if } s=0 & \begin{cases} \min = x_1 \\ \max = x_2 \end{cases} \end{aligned} \quad (2)$$

식(2)을 행렬을 이용하여 식(3)과 같이 표현할 수 있다.

$$\begin{aligned} \text{if } s=1, & \begin{bmatrix} \min \\ \max \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ \text{if } s=0, & \begin{bmatrix} \min \\ \max \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{aligned} \quad (3)$$

식 (3)의 행렬식의 오른쪽의 두 번째 행렬은 식 (4)와 같이 s 를 이용하여 하나의 행렬로 표현 가능하다.

$$\begin{bmatrix} 1-s & s \\ s & 1-s \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} -s & s \\ s & -s \end{bmatrix} \quad (4)$$

식 (3)의 오른쪽의 행렬식을 식 (4)를 이용하여 다시 표현하면 식 (5)와 같다.

$$\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} -s & s \\ s & -s \end{bmatrix} \right) \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} -s & s \\ s & -s \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (5)$$

여기서 식 (5)의 두 번째 항은 s 가 1일 때는 식 (6)과 같고 s 가 0일 때는 식 7과 같다.

$$\begin{aligned} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} -s & s \\ s & -s \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} &= \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} -x_1 + x_2 \\ x_1 - x_2 \end{bmatrix} \end{aligned} \quad (6)$$

$$\begin{aligned} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} -s & s \\ s & -s \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} &= \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{aligned} \quad (7)$$

식 (6)과 식 (7)에서 첫 번째 항은 최소 값을 나타내고 있으며 두 번째 항은 최대 값을 보여주고 있다. 이 관계를 이용하여 신경회로망의 구조와 가중치를 결정할 수 있다. 입력 x_1 과 x_2 에 대해 정렬을 위해 두 층의 신경회로망 구조를 이용한다. 첫 번째 층은 식 (6)과 (7)의 행렬식의 두 번째 항을 표현하도록 구현된다. 두 입력의 차($x_1 - x_2$)를 구하여 결과의 부호에 따라 양수이면 결과를 그대로 활용하고 음수이면 0이 되도록 한다. 이 연산은 그림 3에서 보여주는 함수를 활성화 함수로서 이용하여 얻어낼 수 있다. 두 번째 층은 식 (6)과 (7)에서의 두 항의 덧셈 연산을 담당한다.

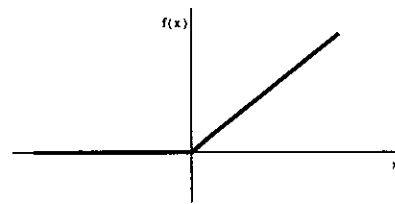


그림 3. 활성화 함수

2-입력 정렬기를 기본으로 N-입력 정렬기를 만들기 위해, 그림 4에서와 같이 2 단의 순환적인 구조를 사용한다. N-입력 정렬기를 위해 L개의 2-입력 정렬기가 사용되며 N과 L간의 관계는 식 (8)와 같다.

$$L = \begin{cases} N - 1, & N \text{ is even} \\ N, & N \text{ is odd} \end{cases} \quad (8)$$

1단의 볼륨은 L/2 개의 2-입력 정렬기로 구성되

고 2단의 블록은 $L/2 - 1$ 개의 2-입력 정렬기로 구성되며 이들은 교대로 동작한다. 이 순환적인 신경회로망은 수번의 반복 적용 후에 입력 신호를 정렬하게 된다. N 개의 입력을 정렬하기 위해 각 단은 $(L+1)/2$ 번 연산된다.

그림 4는 오름차순으로 출력을 제공하는 8-입력 정렬기의 예제를 보여주고 있다. 8-입력 정렬기의 경우에 정렬된 결과는 각 단이 교대로 4번씩 반복 적용된 후에 얻을 수 있다. 7-입력 정렬기도 그림 4와 같은 구조를 가지며 8-입력 정렬기와 같은 연산 횟수를 가지게 된다.

N -입력 정렬기의 경우 2-입력 정렬기를 기본으로 하기 때문에, 전체 입력의 개수가 짝수가 된다. 그러므로 N 이 홀수일 때 하나의 더미(dummy) 입력이 추가로 덧붙여져야 한다. 1 단 블록의 마지막 입력이 더미 입력으로 사용될 수 있다. 더미 입력의 값은 입력으로서 가질 수 있는 가장 큰 값으로 할당함으로써 정렬된 후 rank에 영향을 주지 않도록 한다.

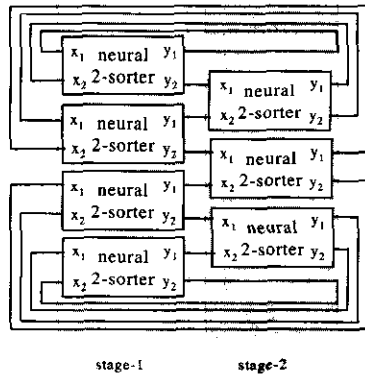


그림 4. 8-입력 정렬기

위의 예제에서 보인 바와 같이 N -입력 정렬기는 2-입력 정렬기를 기본으로 쉽게 확장될 수 있다. 그렇지만 입력의 숫자가 증가함에 따라 반복 횟수 때문에 계산상의 지연도 증가하게 된다.

N -입력을 가지는 M -rank-order 필터는 이 N -입력 정렬기를 이용하여 구현될 수 있다. 일단 N -입력 정렬기를 통해 입력을 정렬한 후 원하는 rank인 M 번째에 위치하는 데이터를 선택하면 된다.

III. 순방향 신경회로망을 이용한 Rank-order 필터의 구현

앞 절에서 설명된 순환적 구조의 모듈러 신경회로망 정렬기를 이용한 rank-order 필터는 비교적 간단한 구조를 가지며 2-입력 정렬기를 기본적인 모듈로 하여 N -입력 정렬기로 확장하는데 용이한 특성을 가진다. 그러나 이러한 장점에 비해 반복적인 연산을 통해 결과를 획득하게 됨으로써 야기되는 계산상의 지연은 입력 노드의 수가 증가함에 따라 큰 문제점으로 인식될 수 있다. 이러한 문제점을 개선하기 위해 순환적인 구조를 통한 반복 연산을 배제하는 구조를 고려해야 하며 본 논문에서는 두 가지의 순방향 신경회로망을 이용할 수 있다. 순방향 신경회로망의 구조와 가중치를 결정하기 위해 입력 신호간의 상대적인 크기 정보를 이용한다. 3.1절에서 입력 신호간의 크기 정보를 표현하는 특징값을 추출하는 방법에 대해 다루고 3.2절에서는 추출된 특징값을 학습 패턴으로 활용하여 신경회로망의 구조와 가중치를 결정하기 위해 역전파 알고리즘을 이용하는 방법에 대해 설명한다. 그리고 3.3절에서는 특징값의 성질을 활용하여 임의의 rank 입력이 외부에서 주어질 수 있는 범용의 rank-order 필터를 설명한다.

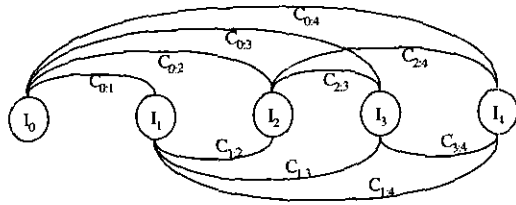
3.1 불변 특징값 추출

M -rank-order 필터는 입력들 중 M 번째 작은 값을 선택하는 필터를 의미하며 구현 방법은 일반적으로 먼저 입력값들을 오름차순으로나 또는 내림차순으로 정렬해야만 한다. 이러한 정렬 과정에서 입력 서로간의 크기 관계에 관한 정보는 정렬을 병렬화하는데 유용하게 이용될 수 있다. 본 논문에서는 이 크기 정보를 표현하기 위해 몇몇 파라미터를 정의한다. 그림 5는 5 개의 임의 순서의 입력($I_0 - I_4$)으로부터 추출 되는 불변 특징값 정보인 $C_{0:1} - C_{3:4}$ 를 보여주고 있다. 파라미터 $C_{m:n}$ 은 입력 I_m 과 I_n 간의 크기 관계를 나타내며 식 (9)과 같이 결정된다.

$$C_{m:n} = \begin{cases} 1, & I_m > I_n \\ 0, & I_m \leq I_n \end{cases} \text{ for } m < n \quad (9)$$

N 개의 입력 $I_0 - I_{N-1}$ 이 주어졌을 때, 각 특징값 패턴의 크기, P 는 N 개중 두 개를 선택하는 조합인 ${}^N C_2 = N(N-1)/2$ 에 의해 주어진다. 또한 실제 특징값 패턴의 개수는 입력들간의 크기 관계의 제한점 때문에 2^P 이 아니라 $N!$ 로 주어진다.

그림 6에서는 입력 노드가 3개인 경우에 대하여 필요한 특징값 정보를 보이고 있다. P 가 3인 경우에 표 1에서 볼 수 있는 것처럼 6개의 패턴이 가능

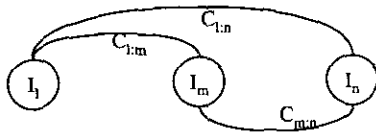


Pattern = $C_{0,1}, C_{0,2}, C_{0,3}, C_{0,4}, C_{1,2}, C_{1,3}, C_{1,4}, C_{2,3}, C_{2,4}, C_{3,4}$

$P = 5! = 10$

그림 5. 크기 관계에 따른 불변 특징값 추출

하다. 예를 들어 $C_{l,m} = 0$ 은 $I_m > I_l$ 임을 의미하고 $C_{l,n} = 1$ 은 $I_l > I_n$ 을 나타낸다. 여기에 $C_{m,n} = 0$ 은 $I_m < I_n$ 일 때 가능하나 I_l 보다 큰 I_m 은 I_l 보다 작은 I_n 보다 작을 수 없다. 이는 논리적으로 발생할 수 없는 경우이다. 이와 같이 표기상 3비트에 표현될 수 있는 경우의 수는 8이지만 크기 관계의 제한점을 고려해 볼 때 실제 특징값으로 이용될 수 있는 경우의 수는 6 가지만이 허락됨을 알 수 있다.



Pattern = $C_{l,m}, C_{l,n}, C_{m,n}$

그림 6. 인접한 세 노드를 위한 특징값 추출

표 1. 인접한 세 노드의 특징값 정보와 관계

$C_{l,m}$	$C_{l,n}$	$C_{m,n}$	remark
0	0	0	$I_n > I_m > I_l$
0	0	1	$I_m > I_n > I_l$
0	1	0	NA
0	1	1	$I_m > I_l > I_n$
1	0	0	$I_n > I_l > I_m$
1	0	1	NA
1	1	0	$I_l > I_n > I_m$
1	1	1	$I_l > I_m > I_n$

3.2 역전파 신경회로망을 이용한 순방향 Rank-order 필터

이 절에서 다루어질 구조는 순방향 신경회로망으로 구성되며 이를 위한 가중치는 입력 신호간의 상대적인 크기 정보를 이용하여 학습 패턴을 구성한

후 역전파 알고리즘을 이용하여 학습을 통해 얻어진다.

제안된 신경회로망은 그림 7에서 보여지는 바와 같이 4 계층 - 입력층, 특징값 추출층, 은닉층과 출력층 - 으로 구성된다. N 개의 입력을 가지는 M-rank-order 필터에 대해, 신경회로망의 입력층은 N 개의 노드를 가지며 특징값 추출층은 $N(N-1)/2$ 개의 노드를 가지고 출력층은 N 개의 노드를 가진다. 은닉층의 노드 수는 학습을 통하여 가장 최적치를 결정하도록 한다. 특징값 추출층에서는 단위 계단 함수를 활성화 함수로 사용하고 특징값 추출층 이외의 다른 층에서는 시그모이드(sigmoid) 함수를 이용한다.

입력층과 특징값 추출층사이의 가중치는 불변 특징값 정보를 얻기 위해 앞 절에서 논의한 점을 바탕으로 하여 1과 -1로 미리 정의되어져 있다. 특징값 추출층과 출력층 간의 가중치는 앞 절에서 설명된 불변 특징값 패턴을 이용하여 역전파 알고리즘에 의해 결정된다.

역전파 신경회로망 부분을 위한 가중치를 결정하기 위하여, N! 개의 학습 패턴이 그림 5에서 보여지는 불변 특징값 패턴들로부터 생성된다. 각 학습 패턴은 입력쪽으로 $N(N-1)/2$ 개의 이진 성분과 목표값으로서 N 개의 이진 성분으로 구성된다. 목표값 패턴은 입력중 M번째로 작은 값의 위치를 이용하여 생성한다. 그림 7은 N이 5인 신경회로망의 예제를 보이고 있다. 그리고 표 2는 5-입력 3-rank-order 필터를 위한 120개의 학습 패턴중 5개의 예제를 보이고 있다. 출력층에서 M번째로 작은 값의 위치의 출력 노드만이 1을 출력하므로 출력 노드의 값들을 입력층에 trigger로서 적용시키면 원하는 rank의 값을 얻을 수 있다.

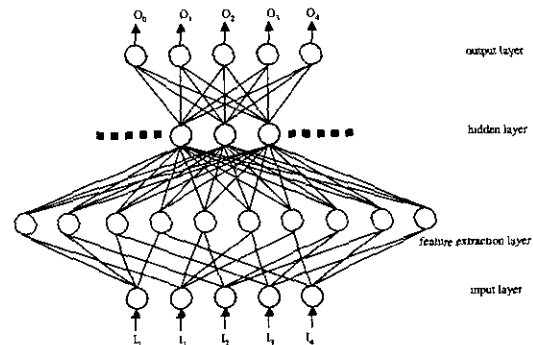


그림 7. 역전파 신경회로망을 이용한 5-입력 rank-order 필터의 구조

표 2. 역전과 신경회로망의 학습을 위한 패턴의 예 (5-입력 3-rank-order 필터를 위한 120개의 학습 패턴중 5개 패턴)

Training patterns										Target patterns				
C ₀₁	C ₀₂	C ₀₃	C ₀₄	C ₁₂	C ₁₃	C ₁₄	C ₂₃	C ₂₄	C ₃₄	T ₀	T ₁	T ₂	T ₃	T ₄
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	1	1	1	1	1	0	0	0	1	0
0	0	0	0	0	0	0	1	1	1	0	0	0	0	1
1	1	1	1	0	1	1	1	1	0	0	1	0	0	0
1	1	1	0	1	0	0	0	0	0	0	0	0	1	0

3.3 rank 입력을 가지는 순방향 rank-order 필터

앞 절에서와 같이 순방향 rank-order 필터를 구현하기 위해 역전과 신경회로망을 이용할 경우 필터를 위한 신경망의 구조와 가중치는 학습을 통해 얻을 수 있다. 이 필터는 귀환적인 구조를 보이지 않고 순방향으로 진행하므로 빠른 응답을 보인다. 그러나 요구되는 rank에 따라 학습이 개별적으로 요구되며 가중치도 달라지므로 임의의 rank에 적용되는 rank-order 필터를 구현하기 위해서 입력과 같은 수의 필터를 요구한다. 이 문제점을 보완하기 위해 임의의 rank 입력에 적용될 수 있는 구조를 고려해 볼 수 있다. 신경회로망은 활성화 함수를 어떤 함수로 선택하느냐에 따라 다양한 응답을 보이는 시스템으로 간주된다. 이 활성화 함수를 원하는 연산을 수행하도록 선택함으로써 효율적인 rank-order 필터를 구현할 수 있다.

rank 입력을 가지는 순방향 rank-order 필터를 위한 신경회로망은 하나의 입력층, 하나의 은닉층 그리고 하나의 출력층으로 구성된다. 가중치들은 학습을 통해서가 아니라 수작업으로 결정되며 '1' 또는 '-1'의 값을 가진다. N개의 입력을 위해 입력층과 출력층은 각각 N개의 노드를 가진다. 그리고 은닉층은 N 개의 unit으로 구성된다. 각 unit은 각 입력 노드를 담당한다. i 번째 unit의 노드의 수, L은 식 (10)에서와 같이 표현된다.

$$L = (N-1) + i, \quad i = 0, 1, \dots, N-1 \quad (10)$$

은닉층은 node type 1과 node type 2의 두 종류의 노드로 구성된다. 이들은 사용하는 활성화 함수에 따라 구분된다. 각 node에 대한 활성화 함수는 식 (11)와 (12)과 같다.

$$u_1(x) = \begin{cases} 0, & x \leq 0 \\ 1, & x > 0 \end{cases} \quad (11)$$

$$u_2(x) = \begin{cases} 1, & x = 0 \\ 0, & otherwise \end{cases} \quad (12)$$

은닉층상의 각 unit의 노드중 N-1 개의 노드는 node type 1을 사용하고 나머지는 node type 2를 사용한다. node type 1은 노드간의 상대적인 크기 정보를 표현하기 위해 사용되고 node type 2는 같은 rank를 가지는 값들의 개수에 대한 정보를 표현한다.

출력 노드들은 활성화 함수로서 선형 함수를 사용한다. 출력층의 각 노드는 M 번째로 작은 값의 위치를 찾기 위한 지침으로서 사용되는 바이어스 성분을 가진다. M-rank-order 필터에 대해 bias 값은 M-1이 사용된다.

입력층의 i번째 노드와 은닉층의 j번째 유닛의 k 번째 노드간의 가중치를 w_{ijk} 라 할 때 이 값은 식 (13)와 같이 정의된다. 또한 은닉층의 i번째 유닛의 j 번째 노드와 출력층의 k번째 노드사이의 가중치 v_{ijk} 는 식 (14)와 같이 정의된다.

$$w_{ijk} = \begin{cases} 1, & \text{if } i=j \\ -1, & \text{if } (i \neq j) \text{ and } (i=k \text{ or } i=k-(N-1)) \\ 0, & otherwise \end{cases} \quad (13)$$

$$v_{ijk} = \begin{cases} 1, & \text{if } i=j \\ 0, & otherwise \end{cases} \quad (14)$$

그림 8은 5-입력을 가지는 M-rank-order 필터를 보여주고 있으며 M번째로 작은 값의 위치의 출력 노드가 '0'을 결과로서 출력하게 된다. 출력층에서 M번째로 작은 값의 위치의 출력 노드만이 0을 출력하므로 출력 노드의 값들을 입력층에 trigger로서 적용시키면 원하는 rank의 값을 얻을 수 있다.

IV. 성능평가

이 논문에서는 신경회로망을 이용한 rank-order 필터를 위한 세 모델을 제안하였다. 이 장에서는 제안한 rank-order 필터의 성능을 평가한다. 2장에서 설명한 2-입력 정렬기를 기본으로 하는 모델과 3.3절에서 설명된 rank 입력을 가지는 rank-order 정렬기는 학습 절차를 요구하지 않는다. 요구되는 가중치는 사전의 수학적 분석에 의하여 결정된다.

그러나 3.2절에서 설명된 역전과 알고리즘을 사용하는 모델은 rank-order 필터를 위해 필요한 가중치와 은닉층 노드의 최적의 수를 얻기 위해 학습 절차를 요구한다. 시뮬레이션을 위해 그림 7에서 보여

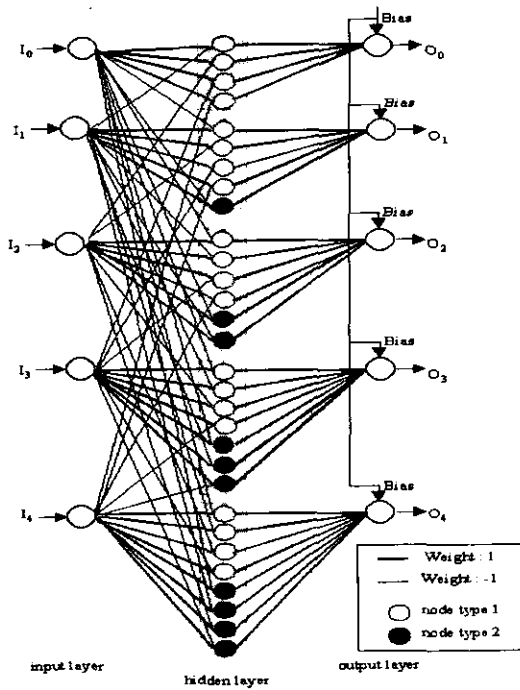


그림 8. 5-입력 rank-order 필터

진 구조를 사용하였다. 표 3은 중간값 필터를 구현하기 위한 실험에서 사용된 파라미터들을 보여주고 있다. 또한 학습을 통해 얻어진 최적의 은닉층의 노드 수도 보여주고 있다. 학습 패턴은 3.2절에서 설명된 방법을 이용하여 생성하였다.

2-입력 정렬기를 기본으로 한 rank-order 필터는 단순한 구조를 가지며 요구되는 노드와 가중치의 수는 상대적으로 적다. 그러나 반복적인 연산을 수행하기 때문에 원하는 출력을 얻기 위해서는 많은 시간이 소요된다.

역전파 신경회로망을 이용한 순방향 rank-order 필터는 상대적으로 복잡한 구조를 가진다. 그러나 원하는 출력을 얻기 위한 시간은 상대적으로 적게 소요된다. 이것은 4 개층의 순방향 신경회로망이기 때문에 반복적인 모델에 비해 매우 빨리 rank를 결정한다. 그러나 신경회로망은 요구되는 rank에 따라 다른 가중치 세트를 가져야만 한다. 이러한 가중치를 얻기 위해서 독립적인 학습 절차를 필요로 한다. 이 구조는 많은 수의 가중치를 필요로 한다.

Rank 입력을 가지는 rank-order 필터는 상대적으로 복잡한 구조를 가지나 역전파 신경회로망을 이용한 rank-order 필터보다는 간단한 연결을 가진다. N 개의 입력을 가지는 임의의 M-rank-order 필터를 위해 하나의 구조만을 필요로 한다. Rank M은 신

표 3. 그림 7에서 보여준 구조를 가지는 중간값 필터를 얻기 위한 실험에서 이용된 파라미터

number of input (node) (N)	3	5	7	9
parameters				
number of nodes in feature extraction layer (f_c)	3	10	21	38
number of training patterns (N)	6	120	5040	362880
number of hidden nodes	3	10	16	30

경회로망의 출력층에 바이어스로 제공되어 제어될 수 있다. 이것은 또한 요구되는 출력을 얻는데 상대적으로 적은 시간이 소요된다.

표 4는 제안된 세 모델간의 비교를 보이고 있다. 여기에서 신경회로망은 층을 통해 연산되기 때문에 하나의 층을 시간상의 지연의 단위로 계산하였다. 예를 들어 rank 입력을 가지는 rank-order 필터는 3 개의 층으로 구성되기 때문에 계산 시간은 3이다. 2-입력 정렬기를 기본으로 하는 모델의 경우에 이 구조는 4 계층을 가지고 있으나 순환적으로 동작하기 때문에 계산 시간은 입력 수에 따라 증가하게 된다. 표는 입력 수에 따른 세 신경회로망 모델의 계산상의 지연, 전체 노드 수 그리고 전체 가중치 수를 비교하고 있다.

표 4. 신경회로망을 이용한 rank-order 필터의 구조 및 성능 비교

- (a) 모듈러 신경회로망 정렬기를 이용한 Rank-order 필터
- (b) 역전파 알고리즘을 이용한 순방향 rank-order 필터
- (c) rank 입력을 가지는 순방향 rank-order 필터

number of input data	3			5			7			9		
type of neural network	a	b	c	a	b	c	a	b	c	a	b	c
computation time (one per layer)	8	3	2	12	3	2	16	3	2	20	3	2
total number of nodes	9	9	12	15	25	35	21	44	70	20	77	117
total number of weights	18	24	27	30	170	90	42	490	189	40	1486	324

V. 결론

이 논문에서는 병렬로 비선형 연산을 수행할 수

있는 신경회로망을 이용하여 rank-order 필터를 구현하기 위한 세 가지 방법을 제안하였다. 첫째로 N-입력 신경회로망 정렬기를 2-입력 모듈러 신경회로망 정렬기를 2단으로 구성하여 구현하였다. 이 N-입력 신경회로망 정렬기는 입력 데이터의 수에 따라 정해지는 반복적인 연산을 통해 입력으로 주어진 값들을 정렬하는 순환적 신경회로망이다. 둘째로 처리 시간을 단축하기 위하여 역전파 알고리즘을 이용한 순방향 신경망을 이용하여 rank-order 필터를 구현하였다. 이 방법은 순환적인 방법에 비해 빠른 응답을 보이지만 학습을 통해 구조와 가중치를 결정하기 때문에 특정 rank에 대해서만 동작한다는 단점이 있다. 이를 보완하기 위해 셋째로 rank 입력을 신경회로망의 입력으로 주어지도록 하여 모든 rank에 대해 동작하는 구조를 구현하였다.

빠른 응답을 요구하는 응용에서는 순방향 신경회로망을 이용한 rank-order 필터가 적합하며 작은 규모의 하드웨어를 요구하는 응용에서는 신경회로망 2-입력 정렬기를 이용하는 방법이 보다 효율적으로 이용될 수 있다. 또한 rank 입력을 가지는 rank-order 필터는 범용의 필터를 요구하는 응용에서 보다 유용하다. 제안된 신경회로망의 구조를 살펴보면 신경회로망의 가중치와 활성화 함수가 간단히 디지털화 될 수 있음을 알 수 있다. 현재 제안된 방법들의 디지털 하드웨어로의 구현이 연구중이다.

Reference

[1] Anil K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, 1989
 [2] Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing*, Addition Wesley, 1992
 [3] Ramesh Jain, Rangachar Kasturi, Brian G. Schunck, *Machine Vision*, 1995
 [4] B. Zeng, "Convergence Properties of Median and Weighted Median Filters", *IEEE Trans. on Signal Processing*, Vol. SP-42, December 1994.
 [5] Hu, M., Vainio, O., Astola, J., Egiazarian, K. & Gevorkian, D."Ram-based programmable stack filter implementation", *Proceedings of ISCAS99*, 1999
 [6] Nodes T. and Gallagher N. "Median filters: some modifications and their properties", *IEEE*

Transactions on Acoustics, Speech and Signal Processing, 1982, pp 739-746.

[7] I. Hatirnaz, F. K. Gürkagnak, Y. Leblebici, "Realization of a Programmable Rank Order Filter Architecture Using Capacitive Threshold Logic Gates", *Proceedings of ISCAS99*, 1999
 [8] Jacek M. Zurada, *Introduction to Artificial Neural Systems*, west publishing company, 1992
 [9] Mohamad H. Hassoun, *Fundamentals of Artificial Neural Networks*, The MIT Press, 1995
 [10] David E. Rumelhart, James L. McClelland and The PDP Research Group, *Parallel Distributed Processing*, A Bradford Book, 1987
 [11] James A. Anderson, *An Introduction to Neural Networks*, A Bradford Book, 1996

윤 숙(Sook Yoon)

준회원



1993년 2월 : 전북대학교
 정보통신공학과 졸업
 1995년 2월 : 한국대학교
 정보통신공학과 석사
 1995년 3월~현재 : 전북대학교
 전자공학과 박사과정

<주관심 분야> 신경회로망, 인공지능, 영상 처리

박 동 선(Dong-Sun Park)

정회원

한국통신학회 논문지 제 24권 제5B호 참조