

단층 코어넷 다단입력 인공신경망회로의 함수에 관한 구현가능 연구

The Implementable Functions of the CoreNet of a Multi-Valued Single Neuron Network

박 종 준*

Jong Joon Park*

Abstract

One of the purposes of an artificial neural network(ANN) is to implement the largest number of functions as possible with the smallest number of nodes and layers. This paper presents a CoreNet which has a multi-leveled input value and a multi-leveled output value with a 2-layered ANN, which is the basic structure of an ANN. I have suggested an equation for calculating the capacity of the CoreNet, which has a p-leveled input and a q-leveled output, as $a_{p,q} = \frac{1}{2}p(p-1)q^2 - \frac{1}{2}(3p-1)(p-2)q + (p-1)(p-2)$.

I've applied this CoreNet into the simulation model 1(5)-1(6), which has 5 levels of an input and 6 levels of an output with no hidden layers. The simulation result of this model gives, the maximum 219 convergences for the number of implementable functions using the $\cot(\sqrt{x})$ input leveling method. I have also shown that, the 27 functions are implementable by the calculation of weight values(w,θ) with the multi-threshold lines in the weight space, which are diverged in the simulation results. Therefore the 246 functions are implementable in the 1(5)-1(6) model, and this coincides with the value from the above equation $a_{5,6}(=246)$. I also show the implementable function numbering method in the weight space.

요 약

인공신경망회로 목표 중의 하나는 최소한의 회로구성으로 구현가능함수를 가능한 많게 하는데 있다. 본 논문은 인공신경망회로의 가장 기본이 되는 하나의 입력노드와 하나의 출력노드, 그리고 입출력에 다단(multi-level)값을 갖는 단층(입출력 2 layer) 다단 코어넷(CoreNet)을 제안하고 그 처리 용량을 구하였고, 무게값 공간에서 구현 가능한 함수와 각 무게값 좌표(ω,θ)를 계산으로 구하여 한 함수의 구현 가능 여부를 알 수 있게 하였다. 또 입력 단계(level)값 설정 방법으로 $\cot(\sqrt{x})$ 을 제안하였다.

제안된 p단 입력과 q단 출력을 갖는 코어넷의 처리용량(구현 가능한 함수의 수)은 $a_{p,q} = \frac{1}{2}p(p-1)q^2 - \frac{1}{2}(p-2)(3p-1)q + (p-1)(p-2)$ 임을 유도 증명하였다.

시뮬레이션으로 5단(level) 입력 값과, 6단 출력 값을 갖는 1(5)-1(6) 모델을 분석한 결과, $\cot(\sqrt{x})$ 입력 레벨링법에서 총 246가지의 함수가 구현가능 함을 보였다. 이 모델의 시뮬레이션 결과에서는 최대 219개의 함수가 수렴(구현 가능)하였고, 구현가능 함수 중에서 나머지 수렴되지 않은 27개의 함수는 무게값 공간에서 무게값 좌표를 계산하여 구현가능함을 보였다. 이는 앞에서 제시된 코어넷 처리용량 $a_{5,6}(=246)$ 에 의한 계산 값과 일치하였다. 무게값 공간에서, 구현 가능한 함수가 차지하는 영역의 함수번호 매김 방법도 제시하여 구현 가능함수의 번호도 알 수 있도록 하였다.

Key words: Artificial Neural Network, Multi-Valued Logic, Machine Learning, Linguistic Process, Perceptron

* Dept. of Computer Science, College of Natural Science & Engineering, Seokyeong University jong@skuniv.ac.kr, 010-5231-8649

※ This Research was supported by Seokyeong University in 2012.

Manuscript received Nov. 17, 2014; revised Dec. 23, 2014; accepted Dec. 24, 2014

I. 서론

인공신경망회로(Artificial Neural Network)의 목적은 주어진 자료로부터 그 특성에 가장 근사한 규칙을 찾아내는 것이다. 현재까지 어느 인공신경망회로도 모든 문제를 다 풀 수 있는(모든 함수 구현 가능) 알고리즘은 발견되지 않았을 뿐 아니라, 모든 가능한 함수(문제) 중 극히 일부분만 구현할 수 있을 뿐이다. 또한 가능한 최소한의 회로 구성으로 최대한의 많은 함수 구현을 시키는 것이 중요하다.

VLSI 칩과 같이 내부 스케일은 커지고 외양 크기는 더 작고, 입출력의 핀 수가 제한된 경우, 다차원의 문제를 푸는 인공신경회로망의 구현에서 여러 개의 입력이 필요한 2치논리(binary logic) 보다는 입력이나 출력 수를 줄일 수 있는 다치논리(multi-valued logic) 회로가 적합하다.[1]

여러 가지 다치논리(multiple-valued logic)가 제안되었으나 보통 2치논리(binary logic)에 비하여 그 비용이 크다. 따라서 여기서는 p-level의 입력단과 q-level의 출력단을 갖고, 다단 문턱값(multiple-thresholds, q-1 thresholds)을 갖는 함수를 사용하는 단세포 회로망(single neuron network)으로 가능한 많은 $f(x_p, y_q)$ 함수 문제를 구현하고자 한다.

다치논리의 인공신경회로망 구현에서 가장 어려운 것은 주어진 문제, 즉 함수에 맞는 무게값 공간을 찾는 것이다. 다단 입출력 조합으로 이루어진 모든 함수의 구현은 불가능하며, 사실 인공신경망 회로로 구현 가능한 함수의 수도 크지 않다. 인공신경망회로의 학습 훈련에서 중요한 점은 무게값 공간(weight value space)에서 주어진 학습 데이터의 특성에 맞는 함수를 찾는 것이다. 즉 해당 함수에 따른 입출력 간 오차가 작아지면서, 충분히 작은 오차 범위로 수렴해야 한다.

다단계값 인공신경망회로(MVNN: Multiple-Valued Neuron Network)를 MVN(Multi-Valued Neurons)으로 부르기도 한다. 이 MVN에서는 무게값과 입출력 값을 복소수로 표현하여 k단(k-valued) 논리 구현가능 함수를 복소평면 함수 $e^j = e^{i(2\pi j/k)}$, $j \in \{0, k-1\}$ 의 k차 근으로 표현하였다. [2][3][4] 코어넷은 실수만을 사용하는 MVNN으로 입출력 노드(세포)가 각각 하나씩이며 중간층(hidden layer)이 없는 구조이다.

문헌[2]는 MVN을 이용한 다단신경망회로에 관한 것으로 복소수 무게값을 이용하여 Parity-n 문제와 두 나선형 자료 분류(two spirals) 문제에 적용하였다. 문헌[3]은 MVN을 이미지 인식에 적용한 예로

흐린 이미지를 스펙트럼 분석과 주파수 도메인에서 분석하여 인식률을 높이고자 하였다. MVN은 비선형 자료의 분류를 비롯한 구현 가능한 함수의 수를 증가시키는 면에서는 장점이나, 복소수 계산과 여러 단계의 활성화 함수(multi-threshold activation function)를 이용함으로 해서 계산 횟수가 많아지고 처리 속도가 늦어진다. 또한 입출력 값들이 복소수공간의 단위원(unit circle)상에 위치하므로 벡터 공간에서 부분적인 편중이 발생하며, 물리적 실용을 위해서 실수로의 변환과정도 필요하다.

본 논문에서는 가장 기본적인 인공신경망회로인 코어넷(CoreNet)을 정의하고 그의 특성과 처리 방법을 이해하여 가능한 많은 함수를 구현할 수 있는 회로를 설계하고자 한다. 즉 이 코어넷의 입출력 값으로 다단계(multi-valued, or multi-level) 방법과, 다단 문턱함수(multi-threshold function)를 사용하여 구현 가능한 함수의 수를 가능한 많게 하고자 한다.

단층(입출력 2 layer)이라는 제한적인 모델이지만 신경망회로의 처리용량을 수식($a_{p,q}$)으로 유도 증명하고, 입력 다단계 값으로 $\cot(\sqrt{x})$ 레벨링 방법을 제시하여, 이전 방법에 비하여 구현가능 함수가 증가함을 보였다.* 또, 코어넷 처리용량 계산식 $a_{p,q}$ 의 유도에 다른 방법을 적용하여 같은 결과를 얻었고, 시뮬레이션 실험으로 그 결과를 비교 검증 하였다.

제시된 코어넷을 합성하여 2개 이상의 다단 입력과 출력으로 모델을 구성하면 MVN과 비슷한 계산량으로 구현가능 함수의 영역을 넓힐 수 있을 것으로 예측된다.

II. 인공신경망회로와 다단계법

1. 코어넷의 정의

인공신경망회로의 가장 기본이 되는 단층 회로로서 다음과 같이 정의하고 그 처리 용량에 관한 수식을 유도한다.[5][6][7][8][9][10]

정의 (코어넷, CoreNet):

코어넷은 하나의 입력 노드를 갖는 입력층과 다단 문턱값(multi-threshold value)을 가진 하나의 출력 노드로 된 출력층으로 구성된 단층 신경망회로로서, 입력노드와 출력노드 값은 모두 다단계(multi-value) 값을 가진다.

코어넷의 구조는 그림 1과 같다. 입출력 값의 표현을 위한 다단계 값은 식 (12)과 (13)에 따른다. 시그

* 문헌 [5]의 $\cot(x)$ 입력 레벨링 방법에서는 n 축에 중첩점이 발생하면 $(p-1)(q-2)/2$ 개의 영역이 감소될 수 있다.

모이드(Sigmoid) 활성화 함수로부터 q단(level) 중 k 단(level) 출력값을 다음과 같이 쓸 수 있다.

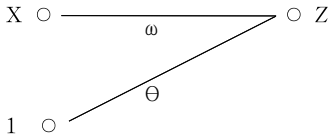


Fig. 1. CoreNet: a Neural Network, with an input and an output

그림 1. 코어넷: 하나의 입력과 하나의 출력을 갖는 신경망회로

$$Z_k = \frac{1}{1 + \exp[-(wX + \theta)/\theta_0]} \quad (1)$$

$$= \frac{k}{q}, \text{ for } 1 \leq k < q, k \text{는 분할선 레벨 값.}$$

여기서 Z_k 는 k단 출력값, X_i 는 i단 입력값, ω 는 무게값, θ 는 문턱값이다. 이를 θ 에 대하여 역변환 하면 식 (2)와 같다. q는 출력 노드의 단계(level)값이다.

$$\theta = -Xw + \theta_0 \ln \frac{k}{q-k} \quad (2)$$

그림 1에서 입력 값 $X = \{x_1, x_2, x_3, \dots, x_p\}$, $|X| = p$, 출력 값 $Z = \{z_1, z_2, z_3, \dots, z_q\}$, $|Z| = q$ 일 때, 이들 입력과 출력 집합으로 만들 수 있는 총 함수의 수는 q^p 개이다.

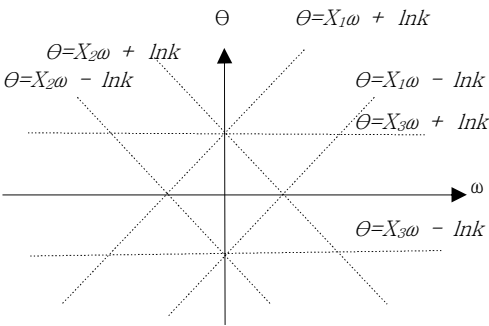


Fig. 2. Separable lines and sectors in weight space for the model of 1(3)-1(3)

그림 2. 모델 1(3)-1(3)의 무게값 공간 분할선과 분할 영역

코어넷의 처리용량은 그림 2와 같은 무게값 공간 (ω, θ) 에서 식 (2)에 의한 다단 문턱선 (multi-threshold lines)으로 나누어지는 부분 즉, 분할 가능한 영역(segment)의 수이다. 예로서, 입력 레벨 수가 3이고 출력 레벨이 3일 때, 그림 2에 보느냐와 같이 무게값 공간을 나눌 수 있는 영역의 수는 17이다. 다단 입출력 레벨링법을 사용하면 2치(binary) 입출력 값 표현에 비하여 공간 분할 영역의 수가 많

아진다.

2. 코어넷의 처리용량(무게값 공간의 분할가능 수)

구현 가능한 함수의 수(처리 용량)는 가능한 함수의 부분집합(subset)으로 예를 들면, p 레벨의 입력과 q 레벨의 출력으로 구성된 모델의 구현 가능한 함수의 수(처리 용량)는 총 조합 가능한 함수는 q^p 가지 함수의 부분집합이다.

$$\{f | f \text{는 구현 가능한 함수}\} \subset \{f_1, f_2, \dots, f_{q^p}\} \quad (3)$$

정리 (코어넷의 용량) :

코어넷의 용량을 무게값 공간의 분할 가능한 영역의 수라하고, p 레벨 입력과 q 레벨 출력을 갖는 코어넷의 용량을 $a_{p,q}$ 라 하자. 입출력이 p와 q레벨인 경우 $p(q-1)$ 개의 공간 분할 선이 나오며, 여기서 p, $q \in \mathbf{N}$, $p \geq 1$ 이며 $q \geq 2$, \mathbf{N} 은 자연수 집합이다. 이 때 코어넷의 처리용량 $a_{p,q}$ 은 다음과 같다.

$$a_{p,q} = \frac{q^2}{2}p(p-1) - \frac{q}{2}(3p^2 - 7p + 2) + (p-1)(p-2) \quad (4)$$

증명 : 모든 선들이 중첩과 평행이 없는 일반분포(in general position)에서 분할되는 영역의 수로부터 점의 중첩과 평행선들에 의하여 감소되는 영역의 수를 뺀다.

$$a_{p,q} = h_{p(q-1)}^{(2)} - O_{pq} - Q_{pq} \quad (5)$$

여기서, O_{pq} 는 p개의 선들이 중첩된 한 점을 지나는 $(q-1)$ 그룹에 의하여 감소되는 영역의 수이며, Q_{pq} 는 $(q-1)$ 개의 선들이 서로 평행으로 된 p 그룹에 의하여 감소되는 영역의 개수이다. 중첩점 O_{pq} 의 개수는 $q-1$ 개 존재하며, Q_{pq} 의 평행 선 그룹은 p개가 존재한다.*

(1) 2차원 평면상에서 일반분포인 n개의 선으로 구분할 수 있는 영역의 수는 다음과 같다.[11]

$$h_n^{(2)} = \binom{n}{0} + \binom{n}{1} + \binom{n}{2}$$

$$= 1 + n + \frac{n(n-1)}{2!}$$

$$= \frac{1}{2}n^2 + \frac{1}{2}n + 1 \quad (6)$$

여기서 $\binom{a}{b}$ 는 a개 중 b개를 취하는 조합의 수이다.

(2) O_p 를 p개의 선들이 한 점을 겹쳐 지날 때 일반분

* $\cot(x)$ 입력 레벨링법에서는 p가 홀수이고 q가 짝수 일 때 ω 축을 중심으로 중첩점(세 개 이상의 선들의 교차)들이 발생하여 $O_e = \frac{1}{2}(p-1)(q-2)$ 수만큼의 영역이 줄었으나 $\cot(\sqrt{x})$ 레벨링법에서는 ω 축과 평행한 선이 없으므로 ω 축 상의 중첩점이 존재하지 않아 O_e 만큼 감소시킬 필요가 없다(문헌 [5] 참조).

포의 수로부터 감소되는 영역의 수라고 하자.

p개의 선들이 나눌 수 있는 영역은 $h_p^{(2)}$ 개이며, p개의 선들이 모두 한 점을 통과해서 만드는 영역의 수는 $2p$ 개이다. 따라서 p개의 선들이 한 점을 통과(p중첩점)하여 감소되는 영역의 수는 식 (7)과 같다.

$$O_p = h_p^{(2)} - 2p \tag{7}$$

$$= \binom{p}{0} + \binom{p}{1} + \binom{p}{2} - 2p$$

$$= \frac{1}{2}(p-1)(p-2), \text{ for } p \geq 3$$

앞에서 정의된 코어넷에서 식 (2)에 따라 θ 축 절편 값들은 $Z_k=k/q, 1 \leq k < q, \omega=0$ 에서, $\theta=\theta_0 \ln(k/(q-k))$ 이며, 이에 따라 q-1개의 점들이 θ 축과 교차한다. 따라서 p개의 선이 한 점을 통과하여 이 q-1개의 p중첩점에 의하여 감소되는 총 영역의 수는 식 (8)과 같다.

$$O_{pq} = (q-1)O_p \tag{8}$$

$$= \frac{1}{2}(p-1)(p-2)(q-1)$$

(3) Q_q 는 (q-1)개의 선들이 서로 평행으로 되어 감소되는 영역의 개수라고 하자. 식 (6)에서와 같이 (q-1)개 일반분포의 선으로 나눌 수 있는 영역의 수는 $h_{q-1}^{(2)}$ 개이며, (q-1)개의 선들이 모두 평행할 때 나누어지는 영역의 수는 q개이다. 따라서 (q-1)개의 선이 모두 평행할 때 감소되는 영역의 수 Q_q 는 식 (9)와 같다.

$$Q_q = h_{q-1}^{(2)} - q \tag{9}$$

$$= \binom{q-1}{0} + \binom{q-1}{1} + \binom{q-1}{2} - q$$

$$= \frac{1}{2}(q-1)(q-2), q \geq 3$$

이 평행 선 그룹은 식 (2)로부터, 기울기 X_i 가 $(2i-1)/2p, i$ 는 $1 \leq i \leq p$ 이므로 이 그룹은 p개가 존재한다. 따라서 총 감소되는 영역의 수 Q_{pq} 는 다음과 같다.

$$Q_{pq} = \sum_{i=1}^p Q_q \tag{10}$$

$$= \frac{1}{2}p(q-1)(q-2)$$

(4) p 단계값을 갖는 하나의 입력과, q 단계값을 갖는 하나의 출력으로 이루어진 코어넷은 한 점을 p개의 선들이 지나가고, 이러한 점들이 (q-1)개 있으며, (q-1)개의 선들이 서로 평행하고, 이러한 평행 그룹이 p개 존재한다(예, 그림 2 참조). 따라서 이 코어넷에서 분리된 영역의 개수는 일반분포 $h_n^{(2)}$ (여기서 $n=p(q-1)$)으로 만들어진 영역의 수로부터 O_{pq} 와 Q_{pq} 값을 뺄 수이다. 따라서 $a_{p,q}$ 는 식 (11)과 같다.

$$a_{p,q} = h_{p(q-1)}^{(2)} - O_{pq} - Q_{pq} \tag{11}$$

$$= \frac{1}{2}[p(q-1)]^2 - \frac{1}{2}p(q-1) + 1$$

$$- \frac{1}{2}(p-1)(p-2)(q-1) - \frac{1}{2}p(q-1)(q-2)$$

$$= \frac{1}{2}[p(p-1)q^2 - (3p-1)(p-2)q + 2(p-1)(p-2)]$$

3. 모델 1(5)-1(6)과 NMLGR(New Multi-Level Grading Rule)에 의한 다단계 신경망회로

다단계(multi-level) 인공신경망회로에서 각 단계(레벨)의 값을 표현하기 위하여 다단계법(MLGR)이 제안되었다.[9][10]

가. 입력단 $\cot(\sqrt{x})$ 레벨링 방법

입력값의 레벨링 방법으로 식 (12)와 같은 단계치를 사용하였다(그림 3).

$$x_i = \begin{cases} -\cot(\sqrt{\frac{2i+1}{2p}} \pi), & \text{for } 0 \leq i < p/2 \\ \cot(\sqrt{\frac{2i+1}{2p}} \pi), & \text{for } p/2 \leq i < p \end{cases} \tag{12}$$

Table 1. The 5 input level values(p=5)

표 1. 5단 입력 레벨링 값(p=5)

| p 입력단 값 i | $\pm \cot(\sqrt{(2i+1)\pi/2p})$ |
|-----------|---------------------------------|
| 0 | -1.593257 |
| 1 | -0.684112 |
| 2 | -0.328597 |
| 3 | 0.684112 |
| 4 | 1.593257 |

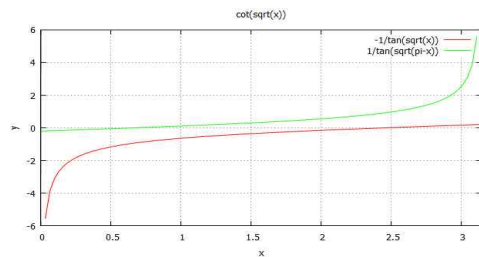


Fig 3. The graph of $-\cot(\sqrt{x}), \cot(\sqrt{\pi-x})$

그림 3. $-\cot(\sqrt{x}), \cot(\sqrt{\pi-x})$ 그래프

입력단계(leveling)법에서 무계값 공간에서 분리되는 영역의 크기가 가능한 고르게 분포되도록 하기 위 $\pm \cot(\sqrt{x})$ 함수를 사용하였으며, $(2i+1)/2p$ 균등 레벨링 방법에 비하여 무계값 공간 영역의 크기가 넓어졌다.

이 방법에서는 기울기가 0인 ω 축과 평행한 선들을 피하여 중첩점들을 감소시키기 위하여 \sqrt{x} 를 사용하였다. 즉 x 중간 값인 $x=\pi/2$ 에서 $\pm \cot(\text{sqrt}(x)) \neq 0$ 이므로 무게값 공간에서 ω 축과 평행한 선(기울기가 0)이 존재하지 않는다. 이로써 p 가 홀수, q 가 짝수인 경우 발생하는 추가 중첩점들을 방지할 수 있다(그림 2 참조).

나. 출력단 레벨링 방법

총 q 단계를 갖는 기호에서 k 번째 등급의 값은 다음과 같이 표현되며 출력단에 적용된다.

$$\frac{2(k-1)+1}{2q}, \text{ for } 1 \leq k \leq q \quad (13)$$

따라서 최대 유효범위는 $-1/2q \sim 1/2q$ 이다. 여기서 유효범위란 해당 레벨 값을 기준으로 $\pm 1/2q$ 허용오차 범위 이내를 의미하며, 활성화 함수의 출력 범위는 $[0, 1]$ 이다.

다. 1(5)-1(6) 모델

1(5)-1(6) 모델은 5단계 값을 갖는 하나의 입력 노드와 6단계 값을 갖는 하나의 출력 노드로 구성된다.

Table 2. The 6 output level values($q=6$)
표 2. 6단 출력 레벨링 값($q=6$)

| q 출력단 값 k | $[2(k-1)+1]/2q$ |
|-----------|-----------------|
| 1 | 0.08333 |
| 2 | 0.25 |
| 3 | 0.41667 |
| 4 | 0.58333 |
| 5 | 0.75 |
| 6 | 0.91667 |

이 모델에 의한 함수는 다음과 같이 표현되며,

$$f: \{x_1, x_2, x_3, x_4, x_5\} \rightarrow \{z_1, z_2, z_3, z_4, z_5, z_6\} \quad (14)$$

$X_i \in \{x_1, x_2, x_3, x_4, x_5\}$, $|X_i|=5$ 이며 $Z_k \in \{z_1, z_2, z_3, z_4, z_5, z_6\}$, $|Z|=6$ 이다. 출력값 Z_k 는 식 (15)와 같다.

$$Z_k = \frac{1}{1 + \exp[-(wX + \theta)/\theta_0]}, \text{ for } 1 \leq k \leq q \quad (15)$$

$$= \frac{2(k-1)+1}{2q}$$

이 함수의 입출력 레벨로 조합 가능한 함수의 수는 6^5 으로 7776가지이다. 이 중에서 구현 가능한 함수의 최대 수는 최대 무게값 공간 분할선의 수 $25(=p(q-1)=5*5)$ 개가 되며, 2차원 공간에서 이 선들이 임의로 배치(일반분포)되어 분할하는 영역의 수는

$\binom{25}{0} + \binom{25}{1} + \binom{25}{2} = 326$ 개이다. 즉 임의로 배치된 25개의 선으로 분할 할 수 있는 최대 영역의 수는 326개이다. 그림 5에서 보듯이 이 모델에서는 5개씩 5쌍의 평행선이 있으므로, $246(=a_{5,6})$ 개의 분할 영역이 나온다.

4. p,(q-1)쌍의 선으로 이루어진 그래프 상에서 구현 가능 함수번호 매김

p 입력단과 q 출력단을 갖는 p, q 모델($1(p)-1(q)$)은 p 개의 선이 기울기를 달리하여 한 점을 지나는 p 중첩선이며 이들은 각각 $(q-1)$ 개의 평행선으로 구성되어 있다. 식 (2)로 표현된 문턱선(threshold lines)들로 구분된 영역에 해당하는 함수번호는 다음과 같다.

(1) 함수번호는 p 자릿수의 q 진법으로 표현된다. 즉 p 는 함수번호의 자릿수 개수이며, 각 자릿수는 q 진법으로 표시된다(표 3 참조).

Table 3. The function numbers for the $1(p)-1(q)$ model(p input levels and q output levels)

표 3. $1(p)-1(q)$ 모델의 함수번호(입출력 각 p 와 q 단)

| $i \setminus f_n$ | $f_{0...0}$ | $f_{0...1}$ | $f_{0...2}$ | ... | $f_{q-1...q-1}$ |
|-------------------|-------------|-------------|-------------|-----|-----------------|
| 0 | 0 | 0 | 0 | ... | $q-1$ |
| 1 | 0 | 0 | 0 | ... | $q-1$ |
| 2 | 0 | 0 | 0 | ... | $q-1$ |
| ... | 0 | 0 | 0 | ... | $q-1$ |
| $p-1$ | 0 | 1 | 2 | ... | $q-1$ |

(2) p 개의 기울기는 q 진법 표현에서 각 자릿수에 해당한다. 식 (2)와 (12)와 같이 입력 레벨링법에 따라 $0 \leq i \leq (p-1)$ 에 따른 문턱선(분할선)의 기울기는

$$-\cot\left(\sqrt{\frac{2i+1}{2p}}\pi\right)$$

이 되며, 이 기울기 그룹은 $(p-i)$ 번째 자릿수로 표현된다. 즉, $i=0$ 일 때 무게값 공간에서 기울기가 가장 큰 선이 되며, 이 기울기의 그룹은 최상위 p 번째 자릿수가 된다. 같은 방법으로 $i=p-1$ 일 때 기울기가 가장 작아(음수), 이 분할선 그룹은 번호의 가장 낮은 자릿수로 표현 된다(표 1 참조).

(3) 출력단 문턱선은 식 (2) $1 \leq k < q$ 로써, $q-1$ 개의 평행선 중에서 $k=1$ 일 때, 가장 아래 선(y 축 절편이 음수 값)에 해당하며, 이 문턱선의 아래 영역은 함수번호 0으로, 위 영역(혹은 $i=2$ 선의 아래 영역)은 번호 1로 표현된다. 같이하여 $i=q-1$ 에 의한 문턱선은 가장 값이 높은 선으로, 이 선 아래 영역은 $(q-2)$ 값, 위 영역은 $(q-1)$ 값으로 표현된다(q 진법 표현).

함수번호 매김 예로서 그림 4는 입출력 각 3단으로 구성된 $1(3)-1(3)$ 모델이며, 함수번호는 000에서

부터 222까지 세(p)자리의 3(q)진법으로 표현된다. 각 선들로 구분된 영역과 번호는 각각 해당되는 구현 가능 함수의 번호를 의미한다(표 3 참조).

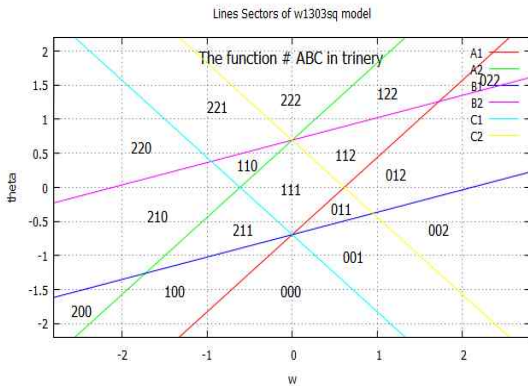


Fig. 4. The threshold lines and sectors in the weight space of the p=3, q=3 model(trinary function numbers used)
 그림 4 무계값 공간에서 p=3, q=3인 모델의 구분선과 각 영역에 해당하는 구현 함수번호(3진법 표현)

이 모델도 그림 2와 같이 세 쌍의 선들로 나눌 수 있는 영역의 수는 $a_{3,3}=17$ 로써 구현 가능한 함수도 17가지이다. 완전히 임의 배열(general position)된 6개 선으로 구분 가능한 영역의 수는 $h_6^{(2)}=22$ 개이나, 이 중 함수번호 $f_{010}, f_{012}, f_{101}, f_{102}, f_{201}, f_{202}, f_{221}, f_{222}$ 7개의 함수는 그림 4에서와 같이 영역 구분이 안 되어 구현 불가능하다.

III. 시뮬레이션 결과와 분석

모델 1(5)-1(6)을 역전파 알고리즘을 기반으로 다단계 입출력 단층 코어넷 신경망회로로 구현하였다. 각 레벨의 유효영역은 최대 유효영역인 100%로 정하고, 활성화 함수 θ_0 는 0.3, 학습률(learning rate) η 는 0.16, 모멘텀(momentum) α 는 0.17에서 가장 많은 구현가능 함수가 나왔다. 오류 교정 최대 반복횟수는 100만회로 비교적 큰 값이나 구분 영역의 크기가 매우 작은 관계로 출력 오차를 충분히 줄이는데 실패하는 경우도 많다.

Table 4. Simulation results of the model 1(5)-1(6)
 표 4. 모델 1(5)-1(6) 시뮬레이션 결과

```

=== function # 0 ===
0: -1.593 0.083
1: -0.684 0.083
2: -0.329 0.083
3: 0.684 0.083
4: 1.593 0.083
Layer 1:
w00=0.00353766 Theta=-0.490785

sample  output  target
0      0.16047  0.083
1      0.16192  0.083
2      0.16249  0.083
3      0.16412  0.083
4      0.16560  0.083
0.out: Total number of iteration is 19
Normalized system error is 0.00346758 <=
0.00347222
Successful Exit with individual error.

=== function # 1 ===
0: -1.593 0.083
1: -0.684 0.083
2: -0.329 0.083
3: 0.684 0.083
4: 1.593 0.250
Layer 1:
w00=0.156628 Theta=-0.591813

sample  output  target
0      0.05708  0.083
1      0.08868  0.083
2      0.10487  0.083
3      0.16582  0.083
4      0.24216  0.250
1.out: Total number of iteration is 17
Normalized system error is 0.000829447 <=
0.00347222
Successful Exit with individual error.
...
=== function # 7775 ===
0: -1.593 0.917
1: -0.684 0.917
2: -0.329 0.917
3: 0.684 0.917
4: 1.593 0.917
Layer 1:
w00=-0.0993351 Theta=0.641211

sample  output  target
0      0.93492  0.917
1      0.91403  0.917
2      0.90432  0.917
3      0.87111  0.917
4      0.83339  0.917
7775.out: Total number of iteration is 19
Normalized system error is 0.00103206 <=
0.00347222
Successful Exit with individual error.
    
```

1. 모델 1(5)-1(6)의 시뮬레이션 결과

표 6는 코어넷 시뮬레이션 출력결과와 일부이다.

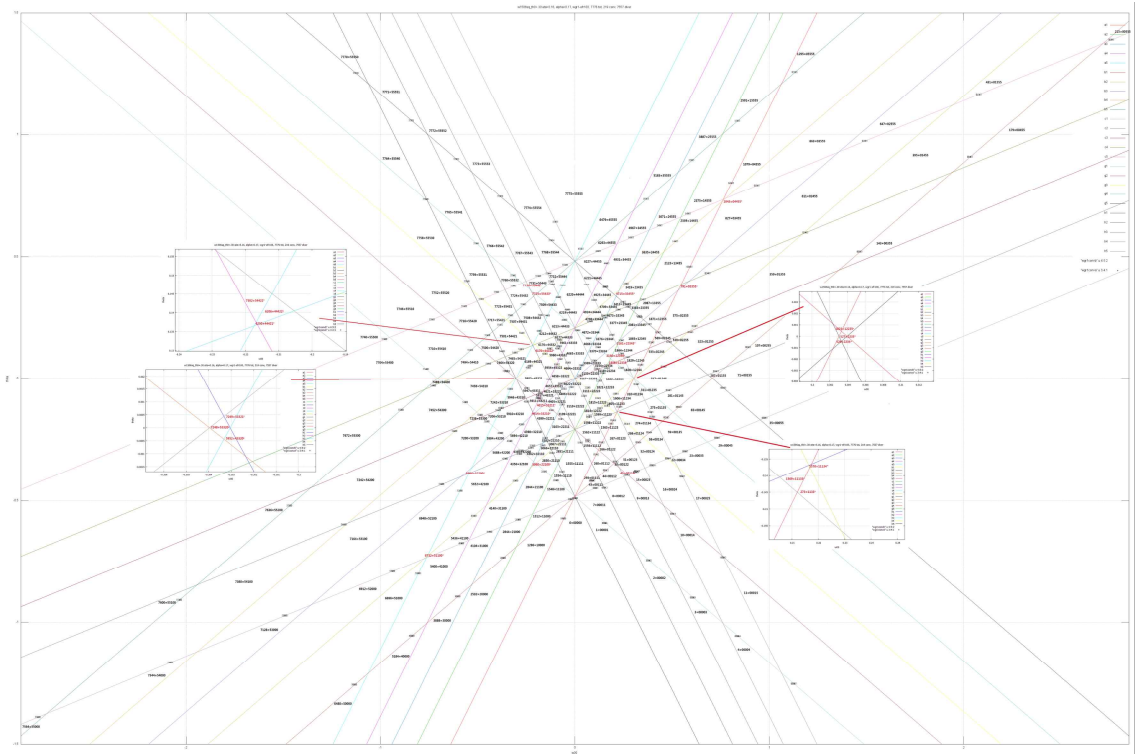


Fig. 5. The converged function sectors and numbers from the simulation results of the model 1(5)-1(6) with $\cot(\sqrt{x})$ input leveling method

그림 5. $\cot(\sqrt{x})$ 입력 레벨링을 이용한 1(5)-1(6) 모델의 구현 가능 영역과 함수번호

이 표에 함수번호와 훈련에 사용된 표본 입력 값들이 나와 있고, 훈련 결과인 출력값(output)과 목표치(target)값, 입출력 연결망의 weight(w00)값, Θ (Theta)들이 나와 있다. 편의상 함수번호는 0번부터 q^p-1 까지로 목표 값(target value)에 따라 붙여졌다. 즉 순서대로 0(00000₆)번부터 7775(55555₆)까지 6진법을 10진수로 표시하였다.

그림 5는 모델 1(5)-1(6)의 시뮬레이션 결과(표 4) 무계값 공간에서 각 함수가 차지하는 영역을 해당 함수 위치 좌표(w00,Theta)와 함께 함수번호로 나타낸 것이다. 그림 6은 그림 5의 중간 부분을 확대한 것으로 시뮬레이션에서 수렴되지 못한 27개의 함수 위치와 함수번호는 6진값 옆에 '*'(빨간색 번호)으로 표시되어 있다. 이들 함수번호는 45, 273, 526, 527, 791, 1043, 1569, 1570, 1823, 1858, 3060, 3160, 3161, 4614, 4615, 4715, 5952, 6170, 6205, 6206, 6732, 6984, 7248, 7249, 7502, 7725, 7730이다.

2. 분할 공간에서 무계값 좌표 계산법에 의한 출력 결과

표 5는 시뮬레이션 결과 수렴되지 못한 27개의 함수에 대하여 그래프로부터 (ω, Θ) 값을 구하여 계산으로 나온 출력 값의 일부분이다. 표에서 보듯이 출력 오차가 모두 $1/2q=0.8333$ 이내에 있으며 다. 총 7776가지의 조합 가능한 함수 중에서 219개의 함수가 수렴되었으며 나머지 7557개의 함수는 발산하였다. 그림 5와 6에서 보듯이 공간 분할선은 5쌍의 5평행선으로 모두 25개이다. 이들 25개의 선으로 분할되는 영역의 수는 모두 246개이며, 이들 구현 가능한 함수 중에서 시뮬레이션 결과 실제로 오차 값이 충분히 작도록 수렴한(구현 가능한) 함수는 최대 219개로 나왔다. 나머지 27개의 함수 출력오차는 표 6에 계산되어 나와 있듯이 모든 함수에서 허용오차(0.8333) 이내에 있다.

따라서 모델 1(5)-1(6)에서 구현 가능한 총 함수의 수는 246개로 식 (11)에서 계산한 $a_{5,6}$ 결과와 일치한다.

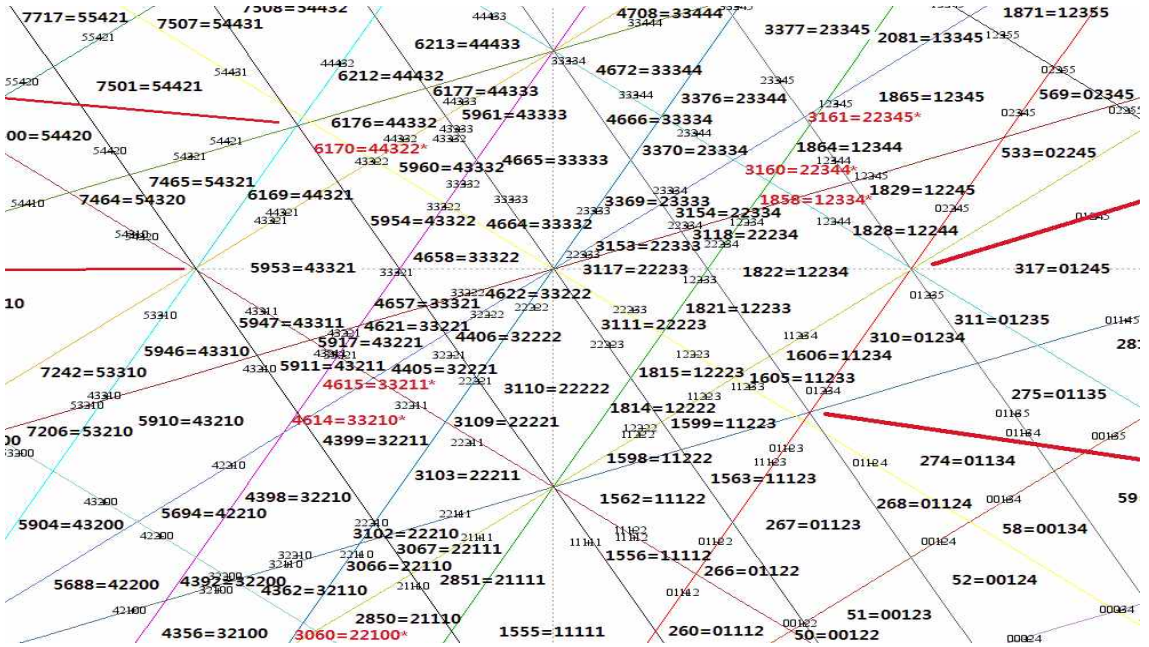


Fig. 6 The enlarged figure from the center part of the Fig. 5: show the implementable function sectors and function numbers as the form of "digit=6nary", the asterisked functions are calculated ones

그림 6 그림 5의 중앙부분 확대: 각 선분들로 나누어진 구분 영역이 무게값 공간 영역에서 구현 가능한 함수 영역이며 해당 함수의 번호(함수번호 10진수=6진수 형식, 별 표시된 함수번호는 시뮬레이션에서 구현 안 된 함수).

Table 5. The calculated outputs of the diverged 27 functions from the simulation results of the model 1(5)-1(6)

표 5. 1(5)-1(6) 모델 시뮬레이션에서 수렴 안 된 27개 함수의 무게값에 의한 출력 계산 값과 오차 (i0=-1.5933, i1=-0.6841, i2=-0.3286, i3=0.68411, i4=1.59325)

| 함수번호 ω, θ | 입력단 | | 출력 값 | | 출력 오차 |
|-----------------------------|-----|---|----------|--------|----------|
| | 단 | 단 | 계산 값 | 목표값 | |
| 45= 00113 ₆ | 0 | 0 | 0.512553 | 0.0833 | 0.0208 |
| | 1 | 0 | 0.327436 | 0.0833 | 0.0738 |
| | 2 | 0 | 0.171144 | 0.25 | 0.0772 |
| 0.25409, -.38977 | 3 | 0 | 0.132543 | 0.25 | 0.0346 |
| | 4 | 0 | 0.06607 | 0.5833 | 0.0692 |
| 273= 11134 | 0 | 4 | 0.662013 | 0.5833 | 0.0787 |
| | 1 | 3 | 0.503539 | 0.5833 | 0.0798 |
| | 2 | 1 | 0.327628 | 0.25 | 0.0776 |
| 0.21717, -.14432 | ... | | | | |
| 7730= 55442 ₆ | 0 | 2 | 0.489662 | 0.4167 | 0.073 |
| | 1 | 4 | 0.672846 | 0.75 | 0.0772 |
| | 2 | 4 | 0.827838 | 0.75 | 0.0778 |
| | 3 | 5 | 0.866289 | 0.917 | 0.0504 |
| | 4 | 5 | 0.932829 | 0.917 | 0.0162 |

IV. 결론

본 논문에서는 단층 다단계 신경망회로망인 코어 넷의 처리용량과 관련하여, 입력값 레벨링 방법으로 $\cot(\sqrt{x})$ 각도법을 제안하여 구현 가능한 함수를 증가시켰다. 또 p단 입력과 q단 출력을 갖는 단층 신경망회로의 처리용량을 구하는 식을 기존 방법과 다른 방법으로 유도, 증명 하였으며, 무게값 공간에서 분할 영역을 함수 번호로 표기하는 함수번호 매김 방법을 제시하여 구체적으로 구현 가능한 함수와 구현 불가능한 함수를 함수번호로 표현 할 수 있게 하였다. 이를 5레벨 입력단과 6레벨 출력단을 갖는 1(5)-1(6) 모델로 시뮬레이션하여 결과를 비교, 분석 하였다.

시뮬레이션 실험 결과 제시된 코어넷 처리용량 식에서 계산된 값(246)보다 27개 적은 219개의 함수가 수렴되었다. 수렴되지 않은 27개의 함수는 무게값 공간에서 차지하는 영역이 매우 작다. 비슷한 크기의 다른 함수 영역에서는 수렴되는 경우도 있으나 시뮬레이션에서 여러 매개변수들 값을 바꾸어도 수렴되지

않았다. 27개의 발산 함수의 무게값 좌표를 그래프로 부터 읽어 식 (1)에 대입하여 계산한 결과 출력 오차가 허용오차 범위 안에 있어 이 함수들은 모두 구현 가능한 함수임이 나타났다.

본 논문에서 제시된 단층 다단계 코어넷의 처리용량을 계산하는 식 (11)은 모든 입력과 출력단의 레벨 수(p 와 q)에 적용할 수 있다(모델 $1(p)-1(q)$). 그러나 그림 5에 보듯이 입력력 레벨값 p 와 q 를 크게 할수록 무게값 공간의 분할 영역이 작아져 시뮬레이션 실험에서는 수렴되지 못하는 함수가 급격히 증가한다 (1(3)-1(6) 모델에서는 2개 함수만이 발산). 즉, 표현 단계(level)값이 커질수록 역전파 알고리즘으로는 함수 수렴에 많은 한계가 있음을 보인다. 그러나 발산된 많은 함수도 무게값 공간 그래프에서 함수번호 매김 방법으로 구현 가능한 함수번호를 찾을 수 있으며, 이 영역의 무게값 좌표를 이용한 출력 계산 결과 허용오차 범위 이내에 있음이 확인되어 모두 구현 가능한 함수로 분류됨을 보였다.

코어넷은 문제(함수)의 종류에 따라 VLSI와 같이 그 크기가 제한된 상태에서 최소한의 구조로 이루어진 다단 입력으로 학습 기능을 최대화 하는데 효율적인 방법이 될 수 있을 것이다. 또 입력과 출력 값의 단계를 무한대로 확장하면 다층 신경망회로의 중간층(은닉층)으로도 해석될 수 있으며 이로써 은닉층에 대한 분석에 용이한 도구가 될 수 있을 것이다.

References

- [1] Takahiro Haga, "An application of the (p,q) -logic to the synthesis of the p -valued logical networks and the s - (p,q) -logical completeness" Information Sciences 115, pp.165-185, 1999
- [2] Igor Aizenberg, "Solving The XOR and Parity n Problems Using a Single Universal Binary Neuron", Soft Computing, vol.12, No3, 2008.2
- [3] Igor Aizenberg, Shane Alexander, and Jacob Jackson "Recognition of Blurred Images Using Multilayer Neural Network Based on Multi-valued Neurons", 2011 41st IEEE International Symposium on Multiple-Valued Logic. 2011
- [4] Igor Aizenberg, Claudio Moraga, and Dmitry Paliy, "A Feedforward Neural Network based on Multi-Valued Neurons", Computational Intelligence, Theory and Applications. Advances in Soft Computing, XIV, Springer, 2005
- [5] Jong Joon Park, "The Capacity of Multi-Valued single layer CoreNet(Neural network) and Precalculation of its Weight Values", Journal of IKEEE, Vol.15 No4, pp.354-362, 2011.12
- [7] Jong Joon Park, "The Functions of Core-Net with 3-levels 2-layered Artificial Neural Networks", Institute of Industrial Technology, Seokyeong University, 25, pp.1-11, 2010
- [8] Jong Joon Park, "The Leveling Method for the Input Values of a Multi-valued Core-net Simulation", Institute of Industrial Technology, Seokyeong University, 26, pp.69-80, 2011.06
- [9] Jong Joon Park, Kandel, Abraham, Langholz, G. and L. Hawkes, "Neural Network Processing of Linguistic Symbols, Fuzzy Sets, Neural Networks, and Soft Computing", Ed. by R.R. Yager and L.A.Zadeh, Van Nostrand Reinhold, pp.265-284, 1994.
- [10] Jong Joon Park, "The Capacity of CoreNet: Multi-Level 2-Layer Neural Networks", The KIPS Transactions, Vol.6, No.8, pp.2098-2105, 1999
- [11] Harding, E.F., "The number of partitions of a set of N points in k dimensions induced. by hyperplanes", Proceedings of the Edinburgh Mathematical Society, Series II15, 285-289, 1967.7
- [12] Takahisa. Toda, On Partitioning Colored Points, <http://arxiv.org/abs/1011.3451v1>, http://arxiv.org/PS_cache/arxiv/pdf/1011/1011.3451v1.pdf, pp.2, 2010.10
- [13] Hirose, Yoshio, Yamashita, Koichi, and Hijiya, Shimpei, "Back-Propagation Algorithm Which Varies the Number of Hidden Units", Neural Networks, Vol.4, pp.61-66, 1991
- [14] Igor Aizenberg, "Learning Nonlinearly Separable mod k Addition Problem Using a Single Multi-Valued Neuron With a Periodic Activation Function", WCCI 2010 IEEE World congress on Computational Intelligence, pp.13-23, 2010 - CCIB, Barcelona, Spain

BIOGRAPHY

Jong Joon Park (Member)

1978 : BS degree in Physics,
Sogang University.

1980 : MS degree in Electronics
Engineering, Yonsei University.

1980~1990 : Assistant Professor in
the Dept. of Computer Science,
College of Industry, Chosun

University.

1994 : Ph.D in Computer Science, College of Art and
Science, Florida State University.

1995~Present : Professor in Computer Science,
Seokyeong University.