

단층 코어넷 다단입력 인공신경망회로의 처리용량과 사전 무게값 계산에 관한 연구

The Capacity of Multi-Valued Single Layer CoreNet(Neural Network) and Precalculation of its Weight Values

박종준*
Jong Joon Park*

Abstract

One of the unsolved problems in Artificial Neural Networks is related to the capacity of a neural network. This paper presents a CoreNet which has a multi-leveled input and a multi-leveled output as a 2-layered artificial neural network.

I have suggested an equation for calculating the capacity of the CoreNet, which has a p-leveled input and a q-leveled output, as $a_{p,q} = \frac{1}{2}p(p-1)q^2 - \frac{1}{2}(p-2)(3p-1)q + (p-1)(p-2)$. With an odd value of p and an even value of q, $(p-1)(p-2)(q-2)/2$ needs to be subtracted further from the above equation.

The simulation model 1(3)-1(6) has 3 levels of an input and 6 levels of an output with no hidden layer. The simulation result of this model gives, out of 216 possible functions, 80 convergences for the number of implementable function using the cot(x) input leveling method. I have also shown that, from the simulation result, the two diverged functions become implementable by precalculating the weight values. The simulation result and the precalculation of the weight values give the same result as the above equation in the total number of implementable functions.

요 약

인공신경망회로에서 아직도 안 풀리는 문제 중 하나는 회로의 처리용량에 관한 것이다. 본 논문은 인공신경망회로의 가장 기본이 되는 하나의 입력과 하나의 출력을 갖은 단층 다단 코어넷을 제안하고 그 처리 용량에 관한 수식을 유도하였다.

제안된 코어넷의 처리 용량으로 p단 입력과 q단 출력을 갖는 코어넷의 처리용량(구현 가능한 함수의 수)은 $a_{p,q} = \frac{1}{2}p(p-1)q^2 - \frac{1}{2}(p-2)(3p-1)q + (p-1)(p-2)$ 이며, 입력단 p 값이 짝수이고, 출력단 q가 홀수값이면 추가로 $(p-1)(p-2)(q-2)/2$ 만큼 감해진다. 입력 값으로 3단(level), 출력 값으로 6단을 갖는 1(3)-1(6) 모델을 시뮬레이션하여 분석한 결과, 총 216가지의 함수 조합에서 입력 레벨링 방법으로 cot(x)를 이용하여 82가지의 함수가 구현가능함을 보였다. 이 모델의 시뮬레이션 결과 80개의 함수가 수렴(구현 가능)하였고, 나머지 수렴되지 않은 함수 중에서 2개의 함수는 무게값 공간에서 무게값 좌표를 미리 계산하여 구현 가능함으로 나와, 총 82개의 구현 가능한 함수가 있음을 보였으며, 이는 위 코어넷 처리용량에 의한 계산 값과 일치하였다.

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

1. 서론

* 서경대학교 이공대학 컴퓨터학과(Computer Science, College of Natural Science & Engineering, Seokyeong University)

接受日: 2011年 12月 5日, 修正完了日: 2011年 12月 19日
掲載確定日: 2011年 12月 26日

VLSI 칩은 그 크기가 매우 제한되고, 입출력의 핀 수를 줄여야 하는 경우가 많다. 이러한 경우 다차원의 문제를 푸는 인공신경회로망의 구현에서 여러 개의 입력이 필요한 2치논리(binary logic) 보다는 입력

이나 출력 수를 줄일 수 있는 다치논리(multi-valued logic) 회로가 적합하다. 다치논리의 인공신경회로망 구현에서 가장 어려운 것은 주어진 문제, 즉 함수에 맞는 무게값 공간을 찾는 것이다. 다만 입출력 조합으로 이루어진 모든 함수의 구현은 불가능하며, 사실 인공신경망 회로로 구현 가능한 함수의 수는 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 입출력간 오차가 작아지면서, 충분히 작은 오차 범위로 수렴해야 한다.

다양한 연속 비선형 가운데층 활성화 함수를 갖고, 충분히 많은 수의 노드를 갖는 MLP를 사용한다면 하나의 가운데층만으로도 충분히 “범용 근사치” 속성을 갖게 할 수 있다.[1][2]

근래의 다치논리를 이용한 인공신경망회로는 복소수를 이용한 MVN(Mulgi-Valued Neurons) 연구가 많이 이루어졌으며, 구현 가능한 함수의 증가에도 많은 진보가 있었다.[3][4][5] 위 MVN을 신경망회로에서는 무게값 표현으로 복소수를 이용하였고, 다단계 문턱함수(multiple-valued threshold functions) 특성을 갖는 활성화 함수(activation function)를 사용하였다.

문헌[3]은 MVN을 이미지 인식에 적용한 예로 흐린 이미지를 스펙트럼 분석과 주파수 도메인에서 분석하여 인식률을 높이고자 하였다. 문헌[4]와 [5]는 MLMVN(MultiLayer Multi-Valued Neurons)를 이용한 다층 다단신경망회로에 관한 것으로 복소수 무게값을 다층으로 확장하였고, 이를 Parity-n 문제와 두 나선형 자료 분류(two spirals), “sonar” 벤치마크 및 Mackey-Glass 시계열 분석에 적용하였다. MVN은 비선형 자료의 분류를 비롯한 구현 가능한 함수의 수를 증가시키는 면에서는 장점이나, 복소수 계산과 여러 단계의 활성화 함수를 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 변환과정이 필요하다.

본 논문은 단층(입출력 2 layer)이라는 제한적인 모델이지만 신경망회로의 처리용량을 수식으로 유도하고 시뮬레이션하여 비교 검증하였다. 제시된 코어넷을 합성하여 2개 이상의 다단 입력과 출력으로 모델을 구성하면 MVN과 비슷한 계산량으로 구현 가능한 함수의 영역을 넓힐 수 있을 것으로 예측된다.

코어넷(CoreNet:다단 2층 신경망회로)이란 인공신경망회로의 가장 작은 단위인 하나의 입력과 하나의 출력을 가지고 중간층이 없는 가장 기본적인 신경망

회로로서 은닉층이 있는 다층 신경망 회로의 분석을 용이하게 하기 위함이다.

본 논문은 p레벨(level:단계)을 갖는 입력 노드와 q레벨의 출력 노드를 갖는 코어넷을 이용하여 입출력 레벨에 따른 조합 가능한 다단계 함수의 구현 가능성에 관한 것으로, 다단 입출력 값들로 조합 가능한 함수 중에서 가능한 많은 수의 인공신경망회로의 구현을 위하여 무게값 공간에서 구현 가능한 함수의 수를 찾는다. 이를 인공신경망회로의 시뮬레이션 결과 수렴되는 함수와 비교 하였으며, 무게값 공간에 분할 영역이 있으나 그 영역이 너무 작아 인공신경망회로에서 수렴(구현)이 안 되는 함수는 그 영역의 중심 무게값을 미리 계산으로 구하여 구현할 수 있음을 보였다. 입력 값의 레벨링 방법으로 무게값의 분할 영역을 넓히기 위하여 $\cot(x)$ 를 이용하였다.

II. 인공신경망회로와 다단계법

1. 코어넷의 정의

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

정의. (코어넷, CoreNet):

코어넷은 하나의 입력 노드를 갖는 입력층과 하나의 문턱값(threshold value)을 포함한 하나의 출력 노드로 된 출력층으로 구성된 2층 신경망회로로서 입력 노드와 출력노드는 그 표현 값으로 다단계 값을 가진다.

코어넷의 구조는 그림 1과 같다. 입출력 값의 표현을 위한 다단계 값은 식 (1)과 (15)에 따른다. 시그모이드(Sigmoid) 활성화 함수로부터 q 레벨의 k 번째 출력값을 다음과 같이 쓸 수 있다.

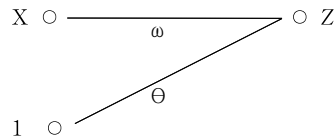


Fig. 1. CoreNet: a Neural Network, with an input and an output

그림 1. 코어넷: 하나의 입력과 하나의 출력을 갖는 신경망회로

$$Z = \frac{1}{1 + \exp[-(\sum_{i=1}^n w_{ji} X_i + \theta_j)]} \tag{1}$$

$$= \frac{k}{q}$$

여기서 Z 는 출력값, X_i 는 입력값, w 는 무게값, θ 는 문턱값이다. 이를 θ 에 대하여 역변환 하면 식 (2)와 같다.

$$\theta = -Xw + \ln \frac{k}{q-k}, \text{ 여기서 } 1 \leq k \leq q-1 \tag{2}$$

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. 코어넷의 처리용량은 무게값 공간에서 그림 2와 식 (4), (5)에 의하여 나누어지는 부분 즉, 분할 가능한 영역(segment)의 수이다.

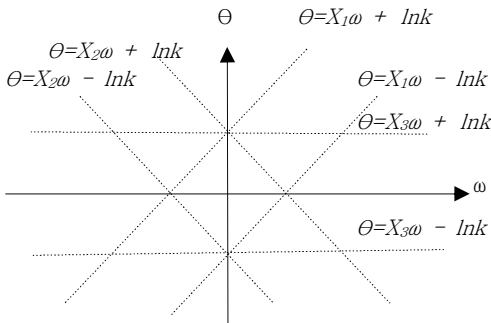


Fig. 2. Separable lines and sectors in weight space for the model of 1(3)-1(3)

그림 2. 모델 1(3)-1(3)의 무게값 공간 분할선과 분할 영역

예로서, 입력 레벨 수가 3이고 출력 레벨이 3일 때, 그림 2에 보이는바와 같이 무게값 공간을 나눌 수 있는 영역의 수(각 선을 연장하여 나뉘어지는 수)는 17이다. 또한 입출력 각 3단으로 가능한 함수의 수는 $q^p = 3^3 = 27$ 가지이다. 이 코어넷으로는 17가지 함수만 구현 가능하며, 나머지 10가지 함수는 구현 불가능하다.

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 함수

의 부분집합이다.

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

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. 이에 는 $p(q-1)$ 개의 분할선이 나오며, 여기서 $p, q \in \mathbf{N}$, $p \geq 1$ 이며 $q \geq 2$, \mathbf{N} 은 자연수 집합이다. 이 때 코어넷의 용량 $a_{p,q}$ 은 다음 두 가지 경우의 식으로 표현 된다.

(1) $p \geq 1$ 와 $q \geq 2$, p 가 짝수이거나 q 가 홀수 일 때,

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

(2) $p \geq 1$ 와 $q \geq 2$, p 가 홀수이고 q 가 짝수 일 때.

$$a_{p,q} = \frac{1}{2}[p(p-1)q^2 - (3p-1)(p-2)q - (p-1)(p-2)(q-4)] \tag{5}$$

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

$$a_{p,q} = h_{p(q-1)}^{(2)} - O_{pq} - O_e - Q_{pq} \tag{6}$$

여기서, O_{pq} 는 p 개의 선들이 중첩된 한 점을 지날 때 일반분포의 수로부터 감소되는 영역의 수이며, Q_{pq} 는 q 개의 선들이 서로 평행으로 되어 감소되는 영역의 개수이다. 중첩점 O_{pq} 의 개수는 $q-1$ 개 존재하며, Q_{pq} 의 평행 선 그룹은 p 개가 존재한다. O_e 는 p 가 홀수이고 q 가 짝수 일 때 w 축을 중심으로 추가로 발생하는 중첩점에 의한 감소 영역의 수이다.

(1) 2차원 평면상에서 세 선 이상 한 점 교차나 두 선 이상 평행이 없는 n 개의 선으로 구분할 수 있는 영역의 수는 다음과 같다.[9][10]

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

$$= \binom{n}{0} + \binom{n}{1} + \binom{n}{2}$$

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

$$= \frac{1}{2}n^2 + \frac{1}{2}n + 1 \tag{7}$$

(2) O_p 를 p 개의 선들이 한 점을 겹쳐 지날 때 일반분포의 수로부터 감소되는 영역의 수라고 하자. 한 점을 교차하는 $p-1$ 개의 선이 있을 때 p 번째 새로운 선이 이 점을 교차할 때 감소되는 영역의 수는 $p-1$ 개이다. 즉 기존 점을 교차하지 않거나 다른 선과 평행하지 않다면, 이 선은 $p-1$ 개의 기존 선을 교차하므로 새로이 분리 생성되는 영역은 p 개가 된다. 따라서 기존 점을 통과하게 되면 두 영역을 통과하여 두 개의

새로운 영역이 생성되어 p-2개의 영역이 감소된다. 앞에서 정의된 코어넷에서 식 (1)에 따라 θ 축 절편 값들은 $Z_i=(2i-1)/2q$, $1 \leq i \leq q$ 이며, $\ln(k/(q-k))$ 를 Z_i 값으로 바꾸면 q-1개의 점들이 θ 축과 교차한다. 따라서 p개의 선이 한 점을 통과하여 이 q-1개의 p그룹에 의하여 감소되는 총 영역의 수는 다음과 같다.

$$O_p = \sum_{i=3}^p (i-2) = \frac{1}{2}(p-1)(p-2), i \geq 3, \quad (8)$$

영역이 감소되는 것은 세 중첩점부터 시작하므로 $i \geq 3$ 이며, q 레벨의 분할선은 q-1개이므로, 이들 중첩점들에 의하여 감소되는 영역의 수는 식 (8)과 같다.

$$O_{pq} = \sum_{i=1}^{q-1} O_p = \frac{1}{2}(p-1)(p-2)(q-1) \quad (9)$$

(3) 입력 레벨 p가 홀수이면 $\cot(p*\pi/2p)=\cot(\pi/2)=0$ 이 되어 분할선 기울기가 0이 되고, 출력 레벨 수 q가 짝수일 때, (q-1)은 홀수가 되어 한 선은 ω 축을 지나게 된다. 이 경우 추가로 ω 축을 따라서 p개의 선이 한 점을 지나는 점(p 중첩점)이 (q-2)개 존재하게 된다. 따라서 p가 홀수이고 q가 짝수이면 다음 O_e 만큼의 수가 추가로 감소된다. p가 짝수이거나 q가 홀수이면 O_e 값은 0이 된다.

$$O_e = \begin{cases} \sum_{i=1}^{q-2} O_p = \frac{1}{2}(p-1)(p-2)(q-2), \\ \text{for } p \text{ is odd and } q \text{ is even} \\ 0, \text{ otherwise} \end{cases} \quad (10)$$

(4) Q_q 는 q개의 선들이 서로 평행으로 되어 감소되는 영역의 개수라고 하자. 기존의 q-1개의 평행선이 있고, 새로이 추가되는 q번째 선이 기존 선과 평행하지 않다면 교차되는 선은 q-1개이므로, 늘어나는 분리영역의 수는 q개이다. 그러나 이 q번째 추가선이 q-1개의 선과 다시 평행하게 되면 교차되는 선은 없으며 하나의 영역만 추가된다. 즉 평행으로 추가되는 선에 의하여 감소되는 영역의 수는 q-1개이다.

$$Q_q = \sum_{i=2}^{q-1} (i-1) = \frac{1}{2}(q-1)(q-2), i \geq 2, \quad (11)$$

평행선은 두 개로부터 시작하므로 $i \geq 2$ 이다.

이 평행 선 그룹은 그림 2와 4, 식 (1)과 (2)에서 보듯이, 기울기 X_i 가 $(2i-1)/2p$ 이며, i 는 $1 \leq i \leq p$ 이므로 이 그룹은 p개가 존재한다. 따라서 총 감소되는 영역의 수 Q_{pq} 는 다음과 같다.

$$Q_{pq} = \sum_{i=1}^p Q_q = \frac{1}{2}p(q-1)(q-2), \quad (12)$$

(5) p 단계값을 갖는 하나의 입력과, q 단계값을 갖는 하나의 출력으로 이루어진 코어넷은 한 점을 p개의 선들이 지나가고, 이러한 점들이 q개 있으며, q개의 선들이 서로 평행하고, 이러한 평행 그룹이 p개 존재한다(그림 4 참조). 따라서 이 코어넷에서 분리된 영

역의 갯수는 일반분포값 $h_n^{(2)}$ (2차원에서 n개의 선, $n=p(q-1)$)에서 O_{pq} 와 O_e , Q_{pq} 값을 뺀 수이다. 따라서, p가 짝수이거나 q가 홀수 일 때,

$$a_{p,q} = h_{p(q-1)}^{(2)} - O_{pq} - Q_{pq} \quad (13)$$

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

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

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

p가 홀수이고 q가 짝수 일 때,

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

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

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

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

$$= \frac{1}{2}[p(p-1)q^2 - (3p-1)(p-2)q - 2(p-1)(p-2)(q-4)] \quad (14)$$

3. 모델 1(3)-1(6)과 NMLGR(New Multi-Level Grading Rule)에 의한 2층 다단계 신경망회로

다단계(multi-level) 인공신경망회로에서 각 단계(레벨)의 값을 표현하기 위하여 다단계법(MLGR)이 제안되었다.[8]

가. 입력단 $\cot(x)$ 레벨링 방법

입력값의 레벨링 방법으로 식 (15)와 같은 단계치를 사용하였다(그림 3).

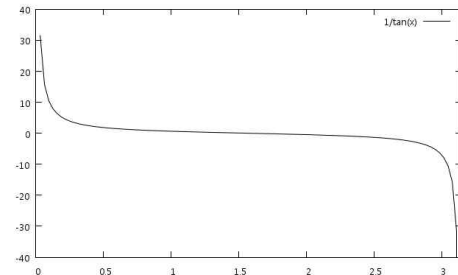


Fig. 3. Graph of $\cot(x)=-\tan(x-1/2\pi)$, $0 < x < \pi$

그림 3. $\cot(x)=-\tan(x-1/2\pi)$, $0 < x < \pi$ 그래프

$$x_i = \cot\left(\frac{2i+1}{2p}\pi\right), \text{ 여기서 } 0 \leq i < p. \quad (15)$$

이는 되도록 무계값 공간에서 분리되는 영역의 크

기가 고르게 분포되도록 하기 위함이며, 기존 [0,1]의 균등 레벨링 방법에 비하여 무게값 공간 영역의 크기가 커졌다.

입력 레벨링 함수로 $\tan(x)$ 를 이용하면 $\tan(\pi/2)$ 이 무한대가 되어 무게값 공간에서 각도 90도에 해당하는 세로축 부근의 분할 공간이 극히 좁아져 중간 레벨값($\sim \pi/2$)이 포함된 함수는 구현이 불가능하다. 따라서 전체 구현 가능한 함수의 수도 크게 감소된다.[11]

나. 출력단 레벨링 방법

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

$$\frac{2(k-1)+1}{2L} \tag{16}$$

따라서 최대 유효범위는 $-1/2L \sim +1/2L$, 즉, $1/L$ 이다. 여기서 유효범위란 해당 레벨 값을 기준으로 $\pm 1/2L$ 의 오차 범위 이내를 의미하며 전체의 퍼센트로 표현한다. 활성화 함수의 출력이 0과 1 사이에 있으므로 이 범위는 [0, 1]이다.[6][7][8]

다. 1(3)-1(6) 모델

1(3)-1(6) 모델은 3단계 값을 갖는 하나의 입력 노드와 6단계 값을 갖는 하나의 출력 노드로 구성된다(그림 1). 이 모델에 의한 함수는 다음과 같이 표현되며,

$$f: \{x_1, x_2, x_3\} \rightarrow \{z_1, z_2, z_3, \dots, z_6\} \tag{17}$$

$X_i \in \{x_1, x_2, x_3\}$, $|X_i| = 3$ 이며 $Z_j \in \{z_1, z_2, z_3, z_4, z_5, z_6\}$, $|Z_j| = 6$ 이다. 출력값 Z_j 는 식 (18)과 같다.

$$Z_j = \frac{1}{1 + \exp[-(\sum_{i=1}^n w_{ji} X_i + \theta_j) / \theta_0]} \tag{18}$$

이 함수의 입출력 레벨로 조합 가능한 함수의 수는 6^3 으로 216가지이다. 이 중에서 구현 가능한 함수의 최대 수는 최대 무게값 공간 분할선의 수 $15(=3*5)$ 개가 되며, 2차원 공간에서 이 선들이 임의로 배치되어 분할하는 영역의 수는 $\binom{15}{0} + \binom{15}{1} + \binom{15}{2} = 121$ 개이다. 즉

임의로 배치된 15개의 선으로 분할 할 수 있는 최대 영역의 수는 121개이다. 여기서 $\binom{a}{b}$ 는 a 개 중 b 개를 취하는 조합의 수이다. 그림 4에서 보듯이 이 모델에서는 3개씩 5쌍의 평행선이 있으므로, $82(=a_{3,6})$ 개의

분할 영역이 나온다.

III. 시뮬레이션 결과와 분석

모델 1(3)-1(6)을 역전파 알고리즘을 기반으로 다단계 입출력 2층 코어넷 신경망회로로 구현하였다. 각 레벨의 유효영역은 최대 유효영역인 100%로 정하고, 활성화 함수 Θ_0 는 0.11, 학습률(learning rate) η 는 0.16, 모멘텀(momentum) α 는 0.15, 오류 교정 최대 반복횟수는 200만회로 정하였다.

1. 모델 1(3)-1(6)의 시뮬레이션 결과

표 1에서 함수번호(“=== 0.dat ===” 형식)와 그 다음 훈련에 사용된 표본 입력 값들이 다음 3줄(3단)에 나와 있다. 0번 함수(0.dat, 0)에서, 훈련 데이터 0번 표본값으로 입력 값 0과 목표치 0.083, 1번 표본값으로는 입력 값 1.732($=\cot(\pi/6)=\tan(\pi/3)$), 그리고 2번 표본값에서는 입력 값으로 -1.732($=\cot(5\pi/5)=\tan(2\pi/3)$)와 같이 주어졌다. “Layer 1:” 다음에는 훈련된 결과 무게값 좌표이며, 이 무게값 좌표로 계산된 출력 이 “output”으로 나와 있다.

Table 1. Simulation results of model 1(3)-1(6)

표 1. 모델 1(3)-1(6) 시뮬레이션 결과

```

=== 0.dat===
0: 0.000 0.083
1: 1.732 0.083
2: -1.732 0.083
Layer 1: w00=-0.0407453 Theta=-0.253791
sample output target
0 0.09053 0.083
1 0.04980 0.083
2 0.15901 0.083
0.out: Total number of iteration is 50
Normalized system error 0.00143976 <= 0.00347222
Successful Exit with individual error.

=== 1.dat===
0: 0.000 0.083
1: 1.732 0.083
2: -1.732 0.250
Layer 1: w00=-0.0553341 Theta=-0.271355
sample output target
0 0.07821 0.083
1 0.03429 0.083
2 0.16860 0.250
1.out: Total number of iteration is 676
Normalized system error 0.0017763 <= 0.00347222
Successful Exit with individual error.

...
=== 45.dat===
0: 0.000 0.250
1: 1.732 0.250
2: -1.732 0.583
Layer 1:
    
```

```
w00=-0.0547396 Theta=-0.0707514
sample output target
0 0.34452 0.250
1 0.18166 0.250
2 0.55447 0.583
45.out: Total number of iteration is 2000000
Normalized system error 0.00269482 <= 0.00347222
...

=== 160.dat===
0: 0.000 0.750
1: 1.732 0.417
2: -1.732 0.750
Layer 1:
w00=-0.05331 Theta=0.0707514
sample output target
0 0.65548 0.750
1 0.45110 0.417
2 0.81497 0.750
160.out: Total number of iteration is 2000000
Normalized system error 0.00269482 <= 0.00347222
...
```

```
=== 214.dat===
0: 0.000 0.917
1: 1.732 0.9172: -1.732 0.750
Layer 1: w00=0.26159 Theta=0.534749
sample output target
0 0.99232 0.917
1 0.99987 0.917
2 0.67752 0.750

=== 215.dat===
0: 0.000 0.917
1: 1.732 0.917
2: -1.732 0.917
Layer 1: w00=-0.100453 Theta=0.355767
sample output target
0 0.96210 0.917
1 0.83923 0.917
2 0.99197 0.917
215.out: Total number of iteration is 9
Normalized system error 0.00261852 <= 0.00347222
Successful Exit with individual error.
```

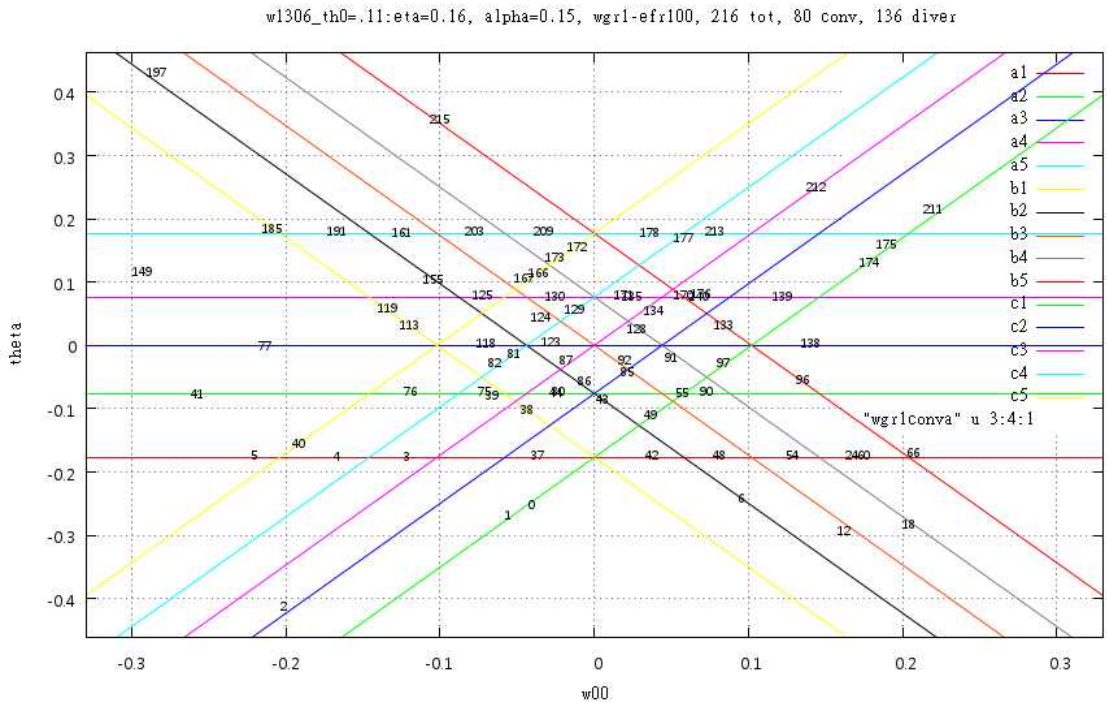


Fig. 4. Functions and their coordinates for the converged ones in the 1(3)-1(6) model

그림 4. 모델 1(3)-1(6)의 각 수렴함수 무게값 공간($a_{3\sigma}=82$ 개의 영역) 좌표와 함수 번호

그림 4는 모델 1(3)-1(6)의 시뮬레이션 결과 무게값 공간에서 각 함수가 차지하는 영역을 함수 번호와 함께 좌표로 나타낸 것이다.

그림 5는 그림 4의 중간 좌측부분을 확대한 것으로 시뮬레이션에서 수렴되지 못한 두 함수 160번(그림 상단 원 표시)과 45번(그림 하단 원 표시) 함수의 위

치를 가리킨다. 좌표에 의한 무게값 계산을 위하여 다음과 같이 선택한다.

함수 160번과 45번 두 함수가 무게값 공간에서 차지하는 분할영역 내 임의의 무게값 좌표로 그림 위쪽 삼각형 내 좌표(w, θ), 즉 함수 f_{160} 의 좌표는 $(-0.051, 0.081)$, 그림 아래쪽 삼각형 내 좌표($w,$

theta)로 함수 f_{45} 의 좌표의 점으로 (-0.051, -0.081)를 선택하여 식 (18)에 대입하여 입출력 값을 계산한다 (표 3 참조).

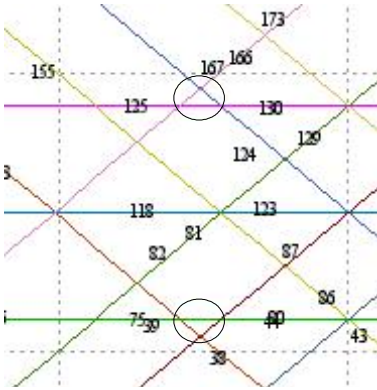


Fig. 5. Enlarged sectors for the function number 45 and 160
 그림 5. 시뮬레이션 수렴 불가 영역인 함수번호 45, 160번 영역

2. 모델 1(3)-1(6)의 시뮬레이션 결과

표 1은 이 모델 1(3)-1(6)의 시뮬레이션 출력 결과의 일부이며, 혼련에 사용된 입력 값들(혼련 값 번호, 입력 값, 목표 값), 무게값(w00와 Theta), 시뮬레이션 결과 계산되어 나온 값들(sample 번호, 출력 값, 목표 값), 그리고 수렴하는데 소요된 반복계산 횟수(iteration)들이 나와 있다. 여기서 편의상 함수번호는 0번부터 q^p-1 까지로 목표 값(target value)에 따라 붙여졌다. 즉 순서대로 0(000₆)번부터 215(555₆)까지 q-진법을 10진수로 표시하였다. 표 2는 실험 결과 수렴된 함수, 즉 구현 가능한 함수들만의 출력력으로서 수렴하기까지의 반복 오차계산 횟수를 보여주고, 그림 4는 무게값 공간에서 수렴 함수들의 무게값 좌표들과 해당 함수번호들을 보여준다.

유효영역의 범위를 축소하면 오차 계산 반복횟수는 늘고, 수렴 함수들의 갯수는 감소하는 반면에 함수의 무게값 좌표들이 무게값 공간의 분할 영역에서 비교적 중앙쪽으로 가까이 간다. 반면에 유효영역을 100%로 설정하면 무게값 좌표들은 이웃 영역과의 경계선 부근에 위치하게 된다.

표 2는 각 수렴된 함수(“함수번호.out”)들의 반복계산 횟수를 나타낸다. 표 2의 시뮬레이션 결과에서 보듯이, 총 216가지의 조합 가능한 함수 중에서 80개의 함수가 수렴되었으며(표 2 하단 참조), 나머지 136개의 함수는 발산하여 구현이 불가능하다. 그림 4에서 공간 분할선은 3쌍의 5평행선으로 모두 15개이

Table 2. Converged function numbers and their iterations of model 1(3)-1(6)

표 2. 모델 1(3)-1(6) 시뮬레이션에서 수렴된 함수 번호와 혼련 반복횟수

| fn.# | iter. | fn.# | iter. | fn.# | iter. | fn.# | iter. | fn.# | iter. |
|--|-------|---------|-------|---------|-------|---------|-------|---------|-------|
| 0.out | : 50 | 1.out | :676 | 2.out | : 7 | 3.out | : 44 | 4.out | : 40 |
| 5.out | : 30 | 6.out | : 18 | 12.out | : 61 | 18.out | : 44 | 24.out | : 42 |
| 30.out | : 21 | 37.out | :158 | 38.out | : 51 | 39.out | : 8 | 40.out | :193 |
| 41.out | : 39 | 42.out | : 89 | 43.out | : 85 | 44.out | : 47 | 48.out | : 43 |
| 49.out | :189 | 54.out | : 55 | 55.out | : 19 | 60.out | :282 | 66.out | : 56 |
| 75.out | :233 | 76.out | : 36 | 77.out | : 10 | 80.out | : 11 | 81.out | : 69 |
| 82.out | :255 | 85.out | : 22 | 86.out | : 67 | 87.out | : 93 | 90.out | : 91 |
| 91.out | : 21 | 92.out | : 10 | 96.out | :224 | 97.out | : 84 | 102.out | : 33 |
| 113.out | :125 | 118.out | : 12 | 119.out | :175 | 123.out | : 4 | 124.out | : 4 |
| 125.out | :104 | 128.out | :152 | 129.out | : 4 | 130.out | :282 | 133.out | :240 |
| 134.out | : 37 | 135.out | : 71 | 138.out | : 16 | 139.out | : 58 | 140.out | :119 |
| 149.out | : 1 | 155.out | : 10 | 161.out | :282 | 166.out | : 54 | 167.out | : 75 |
| 170.out | :187 | 171.out | : 19 | 172.out | :199 | 173.out | : 10 | 174.out | : 6 |
| 175.out | :131 | 176.out | : 52 | 177.out | : 49 | 178.out | :183 | 185.out | : 41 |
| 191.out | : 41 | 197.out | : 11 | 203.out | : 56 | 209.out | : 20 | 210.out | : 1 |
| 211.out | : 41 | 212.out | : 82 | 213.out | : 28 | 214.out | : 16 | 215.out | : 9 |
| The # of run:216 tot, Converge:80, Diverge:136 | | | | | | | | | |

다. 이 3쌍의 5평행선으로 분할되는 영역의 수는 모두 82개이다. 즉 총 82개로 식 (14)에 의하여 $p=3$ 과 $q=6$ 을 대입한 $a_{3,6}=82$ 와 일치한다. 이 82개의 구현 가능한 함수 중에서 시뮬레이션 결과 실제로 오차값이 충분히 작도록 수렴한(구현 가능한) 함수는 최대 80개로 나왔다. 나머지 2개 45번 함수와 160번 함수는 그림 4와 5에서 보듯이 그 크기가 매우 작은 분할 영역들로 이곳의 무게값들을 역전파 알고리즘으로는 찾을 수 없었다(그림 5). 이는 여러 모멘텀과 학습률, θ_0 값들을 달리 하고 충분히 큰 오차수정 반복횟수(200만)를 주어도 같은 결과로 나왔다.

3. 무게값 좌표의 사전 계산법에 의한 출력 값

그림 5는 그림 4에서 중앙에 근접한 상,하 두 작은 삼각형 영역 부분을 확대한 것이다(그림 5에서 원으로 표시). 이 두 작은 삼각형은 분명 분할 영역임에도 불구하고 수렴된 함수가 없었다. 영역내의 좌표값 (-0.051, 0.081)과 (-0.051, -0.081)로 함수의 출력 값을 계산할 수 있다.

그림 5에서 미리 계산된 무게값과 식 (18)에 의한 함수 입력과 출력 값을 계산한 결과 표 3과 같다. 이 표에서 보듯이, 목표 값과 출력 값의 오차가 오차 범위($\pm 0.0833=1/12$)를 벗어나므로 이 모델의 시뮬레이션에서는 수렴되지 못했다. 그러나 그림 5에서 미리 추출된 좌표값을 식 (18)에 대입하여 계산한 값의 오차범위는 0.019~0.074로 최대 오차 범위 0.083($=1/12=1/(2*6)$)보다 적다. 따라서 이 두 함수

Table 3. Outputs of the simulation and the calculated values of the diverged two functions of number 45 and 160

표 3. 시뮬레이션에서 수렴 안 된 두 함수 45번과 160번에 대하여 시뮬레이션 출력 값과 미리 계산된 무게값에 의한 계산 값

| 표본 값 | | 함수# 45(f_{45})=113 ₆ | | |
|------|--------|---------------------------------------|---------|--------|
| 번호 | 입력 | 목표 값 | 출력 값 | 계산 값 |
| 0 | 0.000 | 0.250(1) | 0.34452 | 0.3237 |
| 1 | 1.732 | 0.250(1) | 0.18166 | 0.1766 |
| 2 | -1.732 | 0.583(3) | 0.55447 | 0.5167 |
| 표본 값 | | 함수# 160(f_{160})=424 ₆ | | |
| 0 | 0.000 | 0.750(4) | 0.65548 | 0.6762 |
| 1 | 1.732 | 0.417(2) | 0.45110 | 0.4833 |
| 2 | -1.732 | 0.750(4) | 0.81497 | 0.8234 |

는 구현 가능한 함수임을 알 수 있다. 그림 5에서 보듯이 이 두 함수가 차지하는 무게값 공간의 분할영역이 매우 작기 때문에 시뮬레이션 프로그램에서는 적절한 무게값을 찾지 못하였다. 이 결과는 문턱값, 모멘텀, 학습률 변수를 달리하고, 오차수정 반복회수를 충분히 크게(200만회) 해도 같은 결과로 나왔다.

이 두 함수를 포함하여 모델 1(3)-1(6)에서 구현 가능한 함수의 수는 82개로 본 논문에서 유도한 식 (4), (11)에서 계산한 $a_{3,6} = 82$ 와 일치한다.

IV. 결론

본 논문에서는 2층 다단계 신경망회로망인 코어넷의 처리용량과 관련하여, p단 입력과 q단 출력을 갖는 신경망회로의 처리용량을 구하는 식을 유도하였으며, 이를 3레벨 입력과 6레벨 출력인 1(3)-1(6) 모델로 시뮬레이션하여 결과를 비교, 분석하였다. 입력값 레벨링 방법으로 $\cot(x)$ 각도를 이용하여 구현 가능한 함수를 최대화 시켰다.

시뮬레이션 실험 결과 제시된 코어넷 처리용량 식에서 계산된 값(82)보다 2개 적은 80개의 함수가 수렴되었다. 수렴되지 않은 두 개의 함수는 무게값 공간에서 차지하는 영역이 매우 작다. 비슷한 크기의 또 다른 두 함수 영역은 수렴되었으나 이들 두 함수는 여러 매개변수들을 바꾸어도 수렴 수는 늘지 않는다. 이 두 함수는 무게값 공간의 좌표를 찾아 계산하면 구현 가능한 함수임이 나타난다.

본 논문에서 제시된 2층 다단계 코어넷의 처리용량을 계산하는 식 (13)과 (14)는 모든 입력과 출력단

의 레벨 수(p와 q)에 적용할 수 있다(모델 1(p)-1(q)). 그러나 그림 5에서 언급되었듯이 시뮬레이션 실험에서는 수렴되지 못하는 함수가 다수 존재한다. 즉, 레벨값이 커질수록 해당 함수가 차지하는 분할 영역이 매우 작아져 역전과 알고리즘으로 계산하는데 한계가 있음을 보인다.

실험 결과, 입력 레벨링 방법에 따라서 함수의 출력(함수번호)이 무게값 공간에서 차지하는 영역의 크기와 위치가 다르며, 무게값 공간에서는 분할 영역이 존재하여도 훈련 시뮬레이션에서 수렴되지 못하는 함수가 많이 나온다. 따라서 다치논리를 이용한 정밀한 신경망회로의 설계에서는 입출력 레벨링에 관한 다양한 방법이 필요하다.

코어넷은 문제(함수)의 종류에 따라 VLSI와 같이 최소한의 입력 핀 수나 구조로 이루어진 다단 입력으로 최대한의 학습 기능을 구현하는데 효율적인 방법이 될 수 있을 것이다. 또 입력과 출력 값으로 다단계 값을 가지므로, 이를 오차 범위내 계산 가능한 큰 값으로 확장하면 다층 신경망회로의 중간층(은닉층)으로도 해석될 수 있으며 이를 이용하면 최적의 구현 함수와 노드 수, 구현 불가능한 함수를 찾는 데 이용될 수 있을 것이다.

또한 제시된 코어넷을 여러개 합성하여 2개 이상의 다단 입력과 다단 출력 모델로 확장하면 구현 가능한 함수의 수도 크게 증가 할 것으로 예상된다.

참고문헌

[1] Bishop, C.M. "Neural Networks for Pattern Recognition", *Oxford: Oxford University Press*, pp.121-126, 1995.
 [2] Ripley, B.D. "Pattern Recognition and Neural Networks", *Cambridge:Cambridge University Press*, pp.173-180, 1996.
 [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] Igor Aizenberg and Claudio Moraga, "Multilayer feedforward neural network based on multi-valued neurons(MLMVN) and a backpropagation learning algorithm", *Soft Comput* 11, pp.160-183, 2007
- [6] 박종준, "코어넷(3단 2층 신경망회로)의 구현 가능 함수에 관한 연구", *서경대학교 산업기술연구소 논문집*, 25, pp.1-11, 2010
- [7] Park, Jong Joon, 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.
- [8] Park, Jong Joon, "The Capacity of CoreNet: Multi-Level 2-Layer Neural Networks", *한국정보처리학회 논문지 제6권 제8호*, pp.2098-2105, 1999
- [9] 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 III*5, 285-289, 1967.7
- [10] 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
- [11] 박종준, "2층 코어넷 다단입력 인공신경망회로에서 입력값 레벨링에 관한 연구", *서경대학교 산업기술연구소 논문집*, 26, pp.69-80, 2011.06
- [12] 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
- [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

 저 자 소 개

박종준 (정회원)


1978년 : 서강대학교 물리학과 졸업 (이학사)

1980년 : 연세대학교 대학원 전자공학과 (공학석사)

1980년 9월~2001년 8월: 조선대학교 산업대학 전자계산학과 교수

1994년 : 전자계산학과, Florida State University, (Ph.D)

1995년 3월~현재 : 서경대학교 이공대학 컴퓨터과 학과 교수

<주관심분야> Artificial Neural Network, Web Programming, Internet, 정보처리, 정보보안