

계단형 불연속 함수의 근사화를 위한 새로운 모듈형 신경회로망 학습 알고리즘

正會員 이 혁 준*

A New Modular Neural Network Training Algorithm for Step-like Discontinuous Function Approximation

Hyukjoon Lee* *Regular Member*

※이 논문은 1997년도 광운대학교 교내 학술 연구비에 의하여 연구되었음.

요 약

다층 퍼셉트론이 모든 종류의 함수를 근사화할 수 있다는 것은 이미 증명된 바가 있다. 그러나 불연속 함수의 경우, 오류 역전파와 같은 기존의 훈련 알고리즘으로는 국소 최소점으로서의 수렴 등의 문제로 인하여 높은 정확도를 갖는 함수 근사화가 실질적으로 불가능한 경우가 많다. 본 논문에서는 불연속 함수의 근사화를 위한 새로운 훈련 알고리즘을 소개한다. 제안하는 알고리즘은 모듈 방식으로, 각 신경망 모듈은 전체 훈련데이터 입력 공간중 일부분에 한하여 국부적으로 훈련되고 훈련 결과에 따라 점진적으로 그 수가 증가하며, 최종 단계에서 단일 신경망으로 통합된다. 각 모듈이 담당하는 학습 데이터의 입력공간은 입력공간 창(window)에 의해 점진적으로 팽창하며 그 내부의 훈련 데이터는 독립적으로 정규화된다. 신경망 모듈들을 합치는 과정은 훈련을 통하지만 오류 역전파 과정을 포함하지 않기 때문에 매우 빠르게 진행된다. 실험결과를 통해서 이 알고리즘의 성능이 오류 역전파 알고리즘과 같은 기존의 훈련 알고리즘에 비하여 정확도와 속도면에서 우수함을 보인다.

ABSTRACT

Theoretically, a multi-layered feedforward network has been known to be able to approximate a continuous function to an arbitrary degree of accuracy. However, these networks fail to approximate discontinuous functions when they are trained by well-known training algorithms. This paper presents a training algorithm which doesn't only train the network, but it also constructs the network incrementally while the training proceeds. The final network consists of one or more modules, which are trained in a sequential order within subspaces of the input space, and is trained very rapidly once all modules are trained and merged. The experimental results of applying this method indicates the proposed training algorithm is superior to traditional ones such as backpropagation.

*광운대학교 컴퓨터공학과
論文番號:97394-1030
接受日字:1997年 10月 30日

I. 서 론

다층 퍼셉트론(Multi-layer Perceptron, MLP)의 함수 근사화 능력에 대한 연구는 여러 학자들에 의하여 상당히 깊이 연구되어 왔다. Hecht-Nielsen이 주장한 바에 의하면 Kolmogorov의 고전적인 수학이론은 모든 종류의 연속 함수 $f: [0, 1]^n \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ 를 3개의 층 - n 개의 노드를 포함하는 입력층, $2n + 1$ 개의 노드를 포함하는 은닉층, 그리고 m 개의 노드를 포함하는 출력층 (그림 1) - 으로 이루어진 신경회로망으로 표현할 수 있다는 것을 암시한다고 한다⁽⁵⁾⁽¹⁰⁾. 그는 또한 Fourier 급수 이론을 이용하여 모든 $\epsilon > 0$ 과 L_2 함수 $f: [0, 1]^n \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ 에 대하여 함수 f 를 ϵ 평균 제곱 오차(Mean Squared Error, MSE) 범위내로 근사화할 수 있는 3층 퍼셉트론이 존재함을 증명하였다⁽⁶⁾. Lapedes와 Farber는 sigmoidal 유니트들로 이루어진 두 개의 은닉층에 의해 생성되는 국소 돌출부(local bumps)를 이용하여 함수들을 근사화할 수 있음을 보였다⁽¹¹⁾. Cybenko는 모든 연속 함수(continuous function)는 하나의 은닉층을 갖는 신경망으로 근사화할 수 있다는 것을 증명하였다⁽²⁾. Hornik, Stinchcombe, White 등은 스퀴싱(squashing) 노드 함수를 사용하는 은닉 노드만 충분히 많이 존재한다면 하나의 은닉층을 갖는 MLP로 어떠한 Borel measurable 함수도 근사화가 가능함을 보였다⁽⁸⁾. 그러나 위의 결과들은 단지 MLP의 함수 근사화 능력을 이론적으로 분석하였을 뿐, 모든 함수의 근사화가 가능한 훈련 알고리즘을 제시하지는 않았다.

MLP의 훈련 알고리즘으로 가장 널리 알려진 것은 오류 역전파(backpropagation, BP) 알고리즘이다. 그러나 BP 알고리즘으로 MLP를 훈련할 경우 모든 함수를 근사화할 수 있는 것은 아니다. 이러한 함수의 대표적인 예로 불연속 함수(discontinuous function)와 꺾인점이 포함된 연속 함수 등을 들 수 있다. 그리고 이러한 함수들이 많이 다루어지는 응용 분야로는 컴퓨터비전 분야의 3차원 표면 복구(3D surface reconstruction) 등이 대표적이다. 여기서는 미지의 불연속 에지(edge)를 포함하는 depth data로부터 3차원 표면을 찾아내는 문제를 다룬다. MLP의 불연속 함수 근사화 능력에 관한 연구 결과는 이론적으로나 실험적으로나 거의 존재하지 않는다. 다만 Sontag이 단일 은

닉층 신경망으로 불연속 함수를 근사화하는 것은 매우 어렵다는 것을 언급한 정도이다⁽¹⁷⁾.

한편, 뉴런과 비슷한 단순한 형태의 노드들로 구성된 회로망 구조중에서 불연속 함수의 근사화에 적합한 것들이 있다. 예를 들면, 보간법(interpolation)이나 신호 분해(decomposition)을 위한 Hopfield 네트워크가 그중 하나이다⁽⁷⁾⁽¹⁸⁾. Koch는 이 Hopfield 네트워크를 일반화하여 희박한(sparse) 데이터로부터 불연속성을 보존하면서 완곡 표면(smooth surface)을 복구해주는 신경회로망을 개발하였다⁽⁹⁾. Marroquin이 개발한 하이브리드 아날로그/디지털 회로망은 회로에서 사라지는 전력을 에너지 범함수(functional)로 사용한다. 표면의 깊이 또는 높이 f_i 는 4방향 연결(4-connected) 회로망의 노드 i 의 전압으로 표현되며, 노드들은 서로 스위치와 저항으로 연결되어 있다. 저항들은 인접한 노드들의 전압차의 한계를 결정하는 역할을 담당하며 스위치는 점프 불연속성(jump discontinuity)을 허용하는 메카니즘을 제공한다⁽¹³⁾. Harris가 개발한 Coupled Depth/Slope 모델은 저항과 감산자(subtractor)로 상호 연결된 P 개의 층으로 이루어져 있으며 P 차 미분을 표현할 수 있다⁽⁴⁾. 그러나 위에서 언급한 모델들은 함수값(화소값)이 유한개의(finite) 화소 위치에 한하여 정의되는 이산(discrete) 도메인의 경우에만 적용할 수 있다. 따라서 보간(interpolation)이 불가능하며 n 차원 입력 공간(input space) 함수를 다룰 수 없는 것이 단점이다.

불연속 함수를 다루는 일반적인 접근 방법중의 하나는 regularization 기술을 사용하는 것이다⁽¹⁹⁾. 불연속 함수의 근사화는 ill-posed 문제의 하나이며 따라서 well-posed 문제의 조건(criteria) - 유일 해가 반드시 존재하고 이는 초기 데이터에 연속적으로 의존한다 - 중 한가지 이상에 대하여 만족하지 못한다⁽¹⁾. Ill-posed 문제를 다루는 방법의 핵심은 적당한 사전 지식을 이용하여 합당한 해(solution)의 범위에 한계를 둬으로써 문제를 well-posed 문제의 형태로 변환하는 것이다. Regularization에서는 사전 지식을 활용하여 가능한 해의 범위를 한정짓는 variational principle의 형태 또는 해의 공간의 통계적인 특성의 형태로 변환시키는 방법이 많이 쓰인다. Grimson이 처음 소개한 후, 초기 비전(early vision) 문제에서 자주 사용되어 온 완곡 강요 조건(smoothness constraint)을 통한 regulariz-

ation이 그중 한 예이다⁽³⁾⁽¹⁵⁾.

신경 회로망에서는 Rumelhart등이 가중치 감쇠 (weight decay) 항을 목적 함수에 추가하는 방법을 제안한 바있다⁽¹⁶⁾. Moody는 regularization 파라미터 값으로부터 feed-forward 망의 실질적인 파라미터의 갯수를 결정하는 방법을 개발하였다⁽¹⁴⁾. Poggio등은 Radial Basis Functions을 일반화하여 두 개의 층으로 이루어진 regularization 망을 유도해 냄으로써 regularization에 기초한 함수 근사화의 이론적인 틀을 만들었다. Mao등은 신경망에 적용되는 regularization을 Type I (network-inherent), Type II(algorithm-inherent), Type III(explicitly-specified)의 세 종류로 분류하였다⁽¹²⁾.

본 논문에서는 불연속 함수의 근사화를 위하여 기존의 BP 알고리즘을 개선한 훈련 알고리즘을 소개한다. 이 알고리즘은 회로망의 내적 표현(internal representation)이 형성되는 과정을 조정하여 불연속성 부근에서의 근사화의 오차를 줄이는 방법을 사용한다. 위에서 언급한 Mao의 regularization 분류법에 의하면 이 알고리즘은 Type I과 Type II에 속한다.

본 논문의 구성은 다음과 같다. 제2장에서는 다층 퍼셉트론을 이용하여 불연속 함수를 근사화하는데 따르는 문제점을 살펴보고, 제3장에서는 불연속 함수의 근사화를 위한 새로운 훈련 알고리즘을 소개한다. 제4장에서는 불연속 함수를 근사화한 실험 결과를 통하여 새로운 훈련 알고리즘의 성능을 검증하고, 끝으로 제5장에서는 결론을 맺는다.

II. 시그모이드 유닛을 이용한 불연속성의 근사화

2층 퍼셉트론의 함수 근사화 능력은 이미 많은 연구 결과를 통하여 잘 알려져 있으나 실제로 계단형 불연속성(step-like discontinuity)을 갖는 함수나 또는 구간별 연속(piecewise continuous) 함수를 근사화하기 위한 훈련은 쉽게 이루어지지 않는 경우가 많다. 예를 들면, BP와 같은 기존의 훈련 알고리즘은 충분한 수의 은닉층 노드로도 10개의 등차 계단 함수(step function)와 같은 함수도 근사화하지 못한다(그림 6). 이것은, 훈련 알고리즘의 일반화 특성(generalization ability)이 압도적으로 큰 영향을 미치기 때문으로, 계단들의 중간을 통과하는 완곡한 함수로 근사화 되고

만다. 이런 경우, MSE의 크기가 매우 작기 때문에 가중치 벡터가 전역 최소점을 향해 움직이는 것이 거의 불가능하다. 즉, MSE 만으로는 불연속 함수의 근사화 정도를 표현할 수 없다는 것을 알 수 있다. 그러므로, MSE 외에 목적 함수의 불연속성 주위에서의 급격한 변화 등의 국소 정보를 적절히 사용될 수 있는 방법이 필요하다. 본 논문에서는 이러한 방법을 소개한다. 설명과 이해의 편의상 특별한 경우를 제외하고는 모든 함수는 1차원 입력 함수로 가정한다.

한 개의 스텝 불연속성은 급격히 증가하는 시그모이드 유닛을 한 개로 근사화 할 수 있다(그림 1). 시그모이드 함수의 입력을 $z=c_1(x+c_2)$ 라고 하면 c_1 가 무한대로 접근함에 따라 시그모이드 함수 $f(z)=1/(1+e^{-z})$ 는 $x=c_2$ 에서 가파른(steep) 계단 함수로 접근한다.

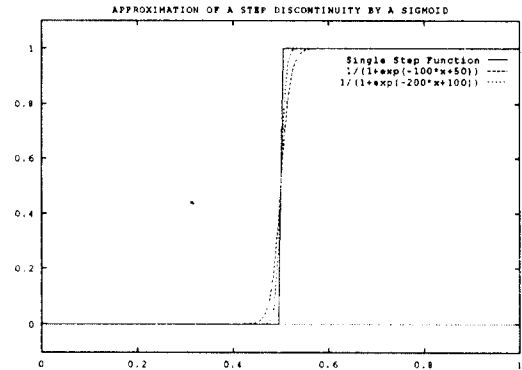


그림 1. 가파른 시그모이드를 이용한 계단형 불연속성의 근사화

Fig. 1 Approximation of a step-like discontinuity by steep sigmoids

그림 1은 시그모이드 함수가 c_1 값이 증가함에 따라 계단 불연속 함수를 더욱 가깝게 근사화하는 것을 보여주고 있다. 한 개의 불연속성을 갖는 구간별 연속 함수 f 는 연속 함수인 f_c 와 스텝 불연속 함수인 f_d 로 분리할 수 있고, $f=f_c+f_d$ 가 된다. 앞서 설명한 바와 같이 f_c 와 f_d 는 각각 두 개의 시그모이드, 즉 s_c 와 s_d 로 충분히 근사화 할 수 있다. 이 경우 f 의 근사 함수는 $s=s_c+s_d$ 가 된다. 다수의 다양한 크기의 스텝 불연속성을 갖는 함수는 같은 수의 가파른 시그모이

드의 선형 조합(linear combination)으로 근사화 할 수 있다(그림 2). 임의의 높이의 불연속성은 시그모이드 활성 함수를 갖는 뉴런의 출력값을 확대 또는 축소함으로써 근사화할 수 있다. 출력 뉴런의 입력 z_0 는 다음의 식에 의해 계산된다.

$$z_0 = \sum_{i=1}^L w_i y_i + \theta \quad (1)$$

이때 L 은 은닉 뉴런의 갯수, w_i 는 은닉 뉴런과 출력 뉴런간의 연결 가중치, y_i 는 은닉 뉴런의 출력, θ 는 문턱값을 나타낸다. 그러므로 각 은닉 뉴런의 시그모이드는 w_i 에 의하여 크기가 조정된다.

결과적으로 시그모이드 활성 함수를 사용하는 MLP는 은닉 뉴런의 수만 충분하다면 계단형 불연속성을 갖는 모든 함수를 근사화할 수 있어야 한다. 그러나, 기존의 BP 알고리즘에 의해 훈련할 경우 모든 가중치는 크기가 작은 난수(random number)를 초기치로 하여 점차 그값이 변하도록 되어 있는데 입력층과 은닉층 사이의 가중치가 충분히 커지지 못하면 불연속성을 근사화하지 못하게된다. 이때, 불연속성을 전혀 근사화 하지 못해도 불연속성 주위에서의 MSE의 크기가 전체 근사화 영역내의 MSE에 미치는 영향이 극히 작을 수 있다. 다음 절에서는 불연속성 근처에서의 오차의 기여도를 선별적으로 증가시켜서 불연속성을 보다 정확히 근사화 하는 알고리즘을 소개한다.

III. Attentive Modular Construction and Training(AMCT)

이 알고리즘의 주 내용은 전체 입력 공간을 다수의 부분공간(subspace)으로 나누어 개별적, 부분적인 근사화를 수행하고 그 결과를 통합하여 전체 함수를 근사화 하는 것이다. 훈련 과정은 모듈 방식에 기반을 두고 있다. 최종적인 신경망은 다수의 모듈로 구성되며 이들은 각자 할당된 부분공간 내에서 독립적으로 함수 근사화를 수행한다. 각 부분공간 내에서는 독립적으로 데이터를 정규화하며 각 부분공간의 위치와 크기는 모듈의 훈련 결과에 따라서 적응적으로 결정된다. 훈련된 모듈들은 출력 뉴런들을 최종적인 신경망의 출력 뉴런에 모두 연결시킴으로서 통합된다. 통합 연결 가중치는 델타 학습 규칙(delta rule)에 의하여 적응적으로 결정된다.

전체 훈련 과정은 d 입력 노드와 소수의 노드로 이루어진 한 개의 은닉층을 갖는 모듈로부터 시작된다. 이 모듈은 한두개의 불연속성만을 포함하고 있는 함수와 같은 매우 단순한 형태의 함수만을 근사화 할 수 있으며, 이 모듈로 근사화가 가능한 부분은 전체 입력공간의 일부에 한정된다. 각 모듈의 근사화 영역은 입력 공간 창(window)에 의해서 결정 된다. 즉, 입력 공간 창은 내부에 포함되는 함수의 극소 복잡도가 충분히 낮은 정도, 즉 0~1개의 불연속성을 포함할 정도로 부터 시작하여 모듈이 감당할 수 없는 정도의 복잡한 함수를 포함할 때까지 확장되고, 입력 공간 창 내의

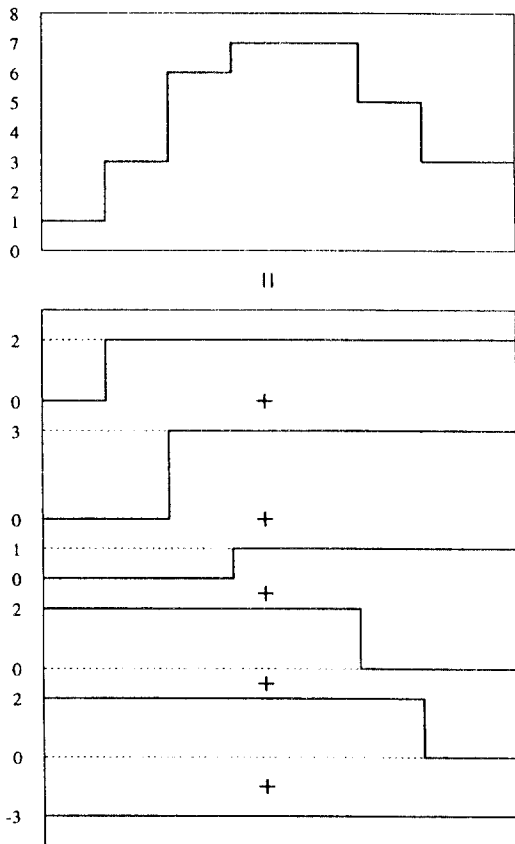


그림 2. 계단형 불연속 함수의 분해
Fig. 2 Decomposition of an arbitrary step-like discontinuous function

함수의 복잡도가 현재의 모듈이 근사화할 수 없는 정도가 되면 새 모듈이 추가된다. 입력 공간 창 내의 함수의 복잡도가 모듈의 능력을 초과함은 MSE의 급격한 상승으로부터 알 수 있다. 새로운 모듈 추가 이전의 마지막 창 확장전의 모듈은 이미 높은 정확도의 근사화를 이룬 상태이며, 입력층-은닉층 연결 가중치는 불연속성의 근사함수 $f(z) = 1/(1 + e^{-z})$, $z = c_1(x + c_2)$ 의 c_1, c_2 값을 구한 상태이다. 따라서, 이들 연결 가중치를 저장하여 모듈 통합시에 사용 한다. 추가된 모듈의 훈련은 이전 모듈의 마지막 확장 영역으로부터 시작되며, 약간의 여유 영역을 주위에 추가하여 불연속성이 영역내의 한쪽에 치우쳐 위치하는 것을 방지한다. 그림 3은 1차원 입력 공간 창의 확장 과정을 보여준다.

2차원 이상의 입력 공간의 확장은 1차원의 경우보다 다양하게 이루어질 수 있다. 가장 간단한 예로, 입력 공간이 직사각형의 2차원 평면인 경우, 입력 공간 창 역시 직사각형 2차원 평면으로, 창의 한쪽 변을 전체 입력 공간의 한쪽변과 같게 놓고 창의 다른 한쪽 변은 전체 입력 공간의 나머지 한쪽변을 따라 확장하는 띠(strip)의 형태로 하거나 또는 이의 수직 방향으로 확장하는 형태로 할 수 있다. 또한, 입력 공간 창의 모양을 정사각형으로 하여 임의의 방향으로 확장하는 것도 가능하다.

목표 함수값은 각 영역 내에서 독립적으로 정규화되므로 불연속성의 위치는 그대로 유지되는 반면에 그 크기는 확대 된다. 따라서 불연속성 근처에서의 오차가 증가하게 되고 모듈내에서 전파되는 오차의 크기도 증가하므로 결과적으로 모듈의 훈련도 쉬워진다. 이렇게 생성된 가파른 시그모이드의 크기는 본래의 것 보다 확대된 것이지만 모듈 통합시에 훈련에 의하여 다시 축소된다.

훈련된 모듈들은 최종 입력 공간의 확장이 더 이상 불가능하거나 최대 모듈수에 도달하게 되었을 때 통합된다. 모듈들이 통합되는 과정은 다음과 같다(그림 4 참조).

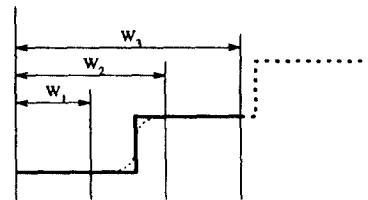
- (1) 모든 모듈들의 은닉 뉴런들은 은닉층을 구성한다.
- (2) 모든 모듈내의 입력층과 은닉층 사이의 문턱값을 포함한 연결 가중치들의 저장된 값을 복사한다.
- (3) 통합 신경망의 은닉층과 출력층 사이의 연결 가중치의 초기치는 부호(+/-)는 모듈로부터 복

사하고 크기는 $[0, 1]$ 사이의 난수값으로 한다.

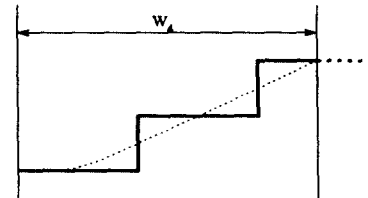
- (4) 모든 모듈내의 은닉층과 출력층간의 문턱값은 폐기하고 통합 신경망의 은닉층-출력층 문턱값은 $[-1, 1]$ 구간내의 난수값으로 초기화 한다.
- (5) 통합 신경망의 은닉층과 출력층 사이의 연결 가중치를 델타 학습 규칙을 이용하여 구한다.

단계 (5)의 자세한 내용은 아래에서 자세히 설명한다.

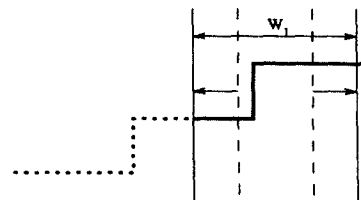
앞서 설명한 바와 같이 통합 신경망의 입력 노드들과 은닉 노드들간의 연결 가중치는 모듈들로부터 복사된다. 이 값들은 이미 각 모듈의 훈련 과정에서 최적화되어 저장된 값이므로 바뀔 필요가 없다. 그러나 은닉 노드들과 출력 노드간의 연결 가중치의 경우는



(A) WINDOW EXPANDED TWICE FOR THE FIRST MODULE AND TRAINING IS SUCCESSFUL



(B) WINDOW EXPANDED AGAIN AND THE FIRST MODULE FAILS TO APPROXIMATE



(C) REINITIALIZE WINDOW FOR THE SECOND MODULE TO THE LAST EXPANSION PLUS AND MINUS ALLOWANCE

그림 3. 입력 공간 창의 확장 과정: 실선으로 표시된 부분은 창내부에 포함되는 함수의 일부이다.

Fig. 3 Explanation of how windows are maintained: Only the solid line portion of function is visible for the given window

다르다. 이들은 스텝 불연속성의 높이와 지향성(orientation)을 나타낸다. 그러나, 각 모듈의 훈련 과정에서 생성되는 값은 독립적 정규화로 인하여 높이 정보를 잃게 된다. 계단을 오르는 방향을 나타내는 지향성에 -1을 곱하는 것은 계단 불연속성을 입력 공간을 기준으로 반사 영상을 얻는 것과 같다. 그러므로 은닉 노드와 출력 노드 사이의 연결 가중치가 음수이면 step-up 불연속성은 step-down 불연속성이 된다. 즉, 하나의 불연속성이 근사화되는 데는 두가지의 경우로 나누어 지는데 이중 어느것이 모듈에 의하여 선택될 것인가는 모듈의 초기 연결 가중치와 훈련과정에 의하여 결정되므로 사전에 알 수 없다. 각 계단 불연속성의 높이는 은닉 뉴런과 출력 뉴런 사이의 연결 가중치를 훈련시킴으로써 근사화한다. 이 훈련 과정은 오류를 역전파하지 않으므로 단시간 내에 종료된다. 문턱값 유니트와 출력 뉴런간의 연결 가중치는

$$T = -\frac{1}{2} \left(\max_x - \min_x \{z_o(x)\} - \min_x \{z_o(x)\} \right) \quad (2)$$

```

/* Initialize the first window to be of zero
size. */
/* Initialize the first module. */
N_modules <- 0;
Do
  Do
    /* Expand the window and train the
current module. */
    Expand_window();
    flag <- BP_train();
  Until flag = fail OR
    window_size = training_set_size
  if flag = fail
    /* Freeze the current module and
activate a new module. */
    N_modules <- N_modules + 1;
  end_if
Until N_modules = max. no. of modules OR
  current window covers the entire
  training set
/* Merge all modules & train entire network */
    
```

그림 4. Attentive 모듈 통합과 훈련 알고리즘
Fig. 4 Attentive Modular Construction and Training Algorithm

이며, 여기서 $z_o(x)$ 는 출력 노드의 총입력값에서 은닉층 문턱값을 제외한 값이다. 즉,

$$z_o(x) = \sum_{i=1}^L v_i f_i(x) \quad (3)$$

이며, 여기서 $f_i(x)$ 는 i 번째 은닉 노드의 활성화 함수값이고 v_i 는 전역 최소점에서 은닉층-출력층 연결 가중치이다. 이론적으로는 T 는 위의 식(2)에 의하여 정해지지만, 실제로는 v_i 와 함께 훈련을 통하여 정한다.

은닉층과 출력층간의 연결 가중치를 훈련한 다음 마지막으로 통합된 망 전체를 기존의 BP 알고리즘을 사용하여 잠시 훈련한다. 이것은 통합 신경망의 미세 조정을 위한 것이다. 그림 4는 지금까지 설명한 알고리즘을 정리한 것이며 그림 5는 이것을 그림으로 나타낸 것이다.

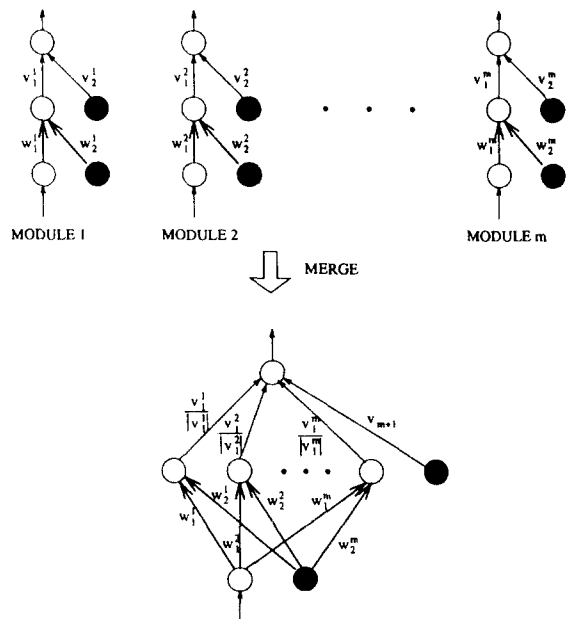


그림 5. 모듈의 통합 과정: 굵은 화살표는 통합과정에서 그 값이 복사되는 부분이며 검은색 원은 문턱값을 나타낸다.

Fig. 5 Construction of Network from Modules: Large arrows represent the connection weights copied in the merge process and the filled circles represent threshold units

IV. 실험 및 고찰

AMCT의 불연속 함수 근사화 능력을 테스트하기 위해 다음의 1차원과 2차원 함수들을 대상으로 근사화를 시도하였다.

- (1) 1차원 등변 10계단 함수
- (2) 1차원 비등변 6계단 함수
- (3) 2차원 등변 10계단 함수
- (4) 2차원 등변 피라미드형 5계단 함수
- (5) 불연속면과 곡면을 포함하는 2차원 함수

입력과 출력은 모두 [0, 1] 구간내로 정규화 하였다. 훈련속도를 향상시키기 위해서 [-0.001, 0.001] 범위의 균등 분포(uniform distribution) 노이즈를 목표 함수값에 더하였다. AMCT와 BP의 훈련 시간에 대한 성능을 비교하기 위해 AMCT의 훈련중에 연결 가중치 경신 회수의 총합을 구하였다.

4.1 1차원 등변 10계단 함수

훈련 데이터는 200개의 입출력쌍으로 10개의 높이와 간격이 같은 계단으로 이루어진 함수를 나타낸다(그림 6). 입력 공간 창은 크기가 0.08이며 x축 상의 원점으로부터 시작하여 x의 값이 증가하는 방향으로 확장하도록 했다. 각 모듈은 한 개의 은닉 노드로 이루어진 단일 은닉층을 갖는다. 그림 6(a)는 AMCT를 이용한 근사화 결과이다. 그림 7(a)는 기존의 BP를 사용한 결과로 불연속성을 전혀 근사화하지 못하고 있어서 이전의 AMCT의 근사화 성능과 크게 차이남을 쉽게 알 수 있다. 그림 6(b)와 그림 7(b)는 각각 AMCT와 BP의 훈련중의 MSE의 변화를 나타낸다. 특히, 그림 6(b)에서 피크 부분은 입력 공간 창내에 2개의 불연속성이 포함됨으로 인하여 MSE가 상승하는 것을 나타낸다. 마지막 피크는 통합망의 훈련과정을 나타낸다. AMCT를 사용했을 경우의 총 모듈의 수는 9개로 불연속성의 갯수와 동일하다. 표 1은 AMCT와 BP의 훈련 결과를 요약한 것이다. 표에서 알 수 있듯이 두 결과 간에 MSE의 차이는 그리 크지 않다. BP에 의한 근사화가 이루어지지 않았으므로 훈련 속도 등의 성능 비교는 별로 의미가 없으나 훈련이 종료된 시점에서 연결 가중치의 총 경신 회수는 AMCT가 BP보다 작았다.

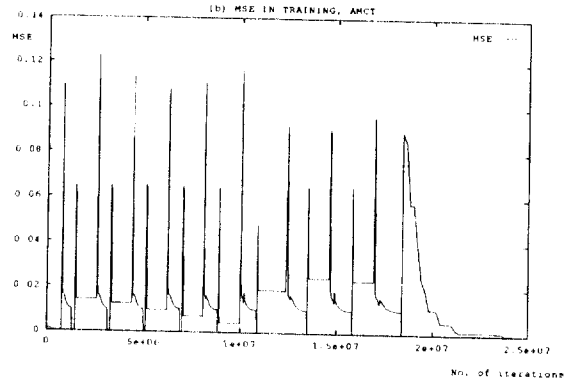
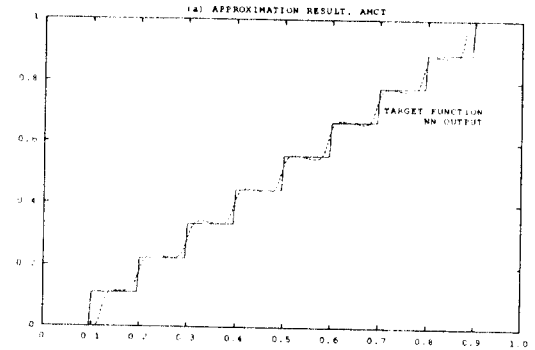
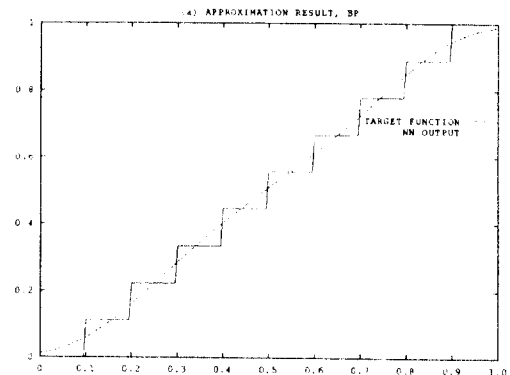


그림 6. AMCT에 의한 등변 10계단 함수의 근사화 결과와 훈련중 MSE의 변화

Fig. 6 Approximation of regular 10 steps by AMCT and change in MSE during training



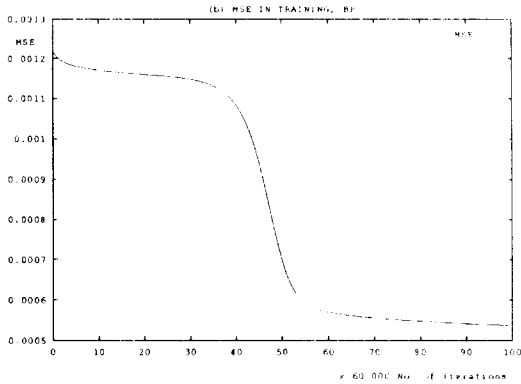


그림 7. BP에 의한 등변 10계단 함수의 근사화 결과와 훈련중 MSE의 변화

Fig. 7 Approximation of regular 10 steps by standard BP and change in MSE during training

표 1. AMCT와 BP의 1차원 등변 10계단 함수 근사화에서의 성능 비교

Table 1. Comparison between AMCT and BP for regular 10 steps

Learning Rules	AMCT	Back-prop	Improvement
MSE	0.000364	0.000537	47.52%
No. of Weight Changes	139,147,516	186,000,000	33.67%
No. of Iterations	24,339,379	6,000,000	

4.2 1차원 비등변 6계단 함수

이 실험에서 사용된 훈련 데이터는 100개의 샘플로 구성되어 있고 6개의 서로 다른 높이와 상호간 거리를 갖는 계단을 포함한다. 각 모듈의 입력 공간 창의 크기와 확장 방향 및 각 모듈의 구조는 4.1절의 경우와 같다. 그림 8은 AMCT를 이용한 근사화 결과이며

표 2. AMCT와 BP의 1차원 비등변 6계단 함수 근사화에서의 성능 비교

Table 2. Comparison between AMCT and BP for irregular 6 steps

Learning Rules	AMCT	Back-prop	Improvement
MSE	0.002242	0.003229	44.23%
No. of Weight Changes	55,056,520	88,000,000	59.83%
No. of Iterations	10,794,130	4,000,000	

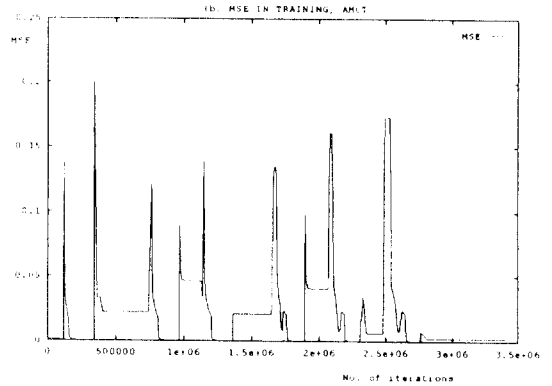
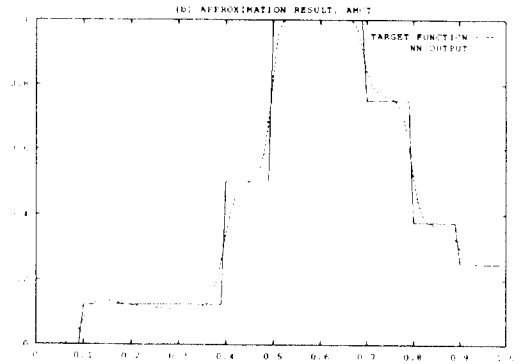
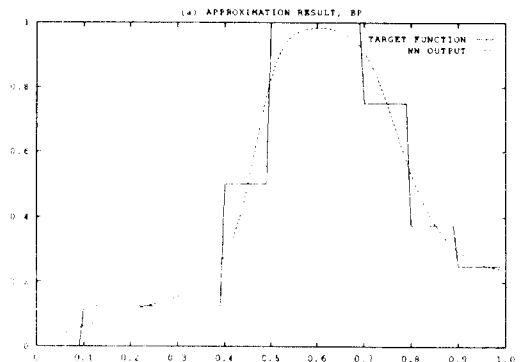


그림 8. AMCT에 의한 비등변 6계단 함수의 근사화 결과와 훈련중 MSE의 변화

Fig. 8 Approximation of irregular 6 steps by AMCT and change in MSE during training



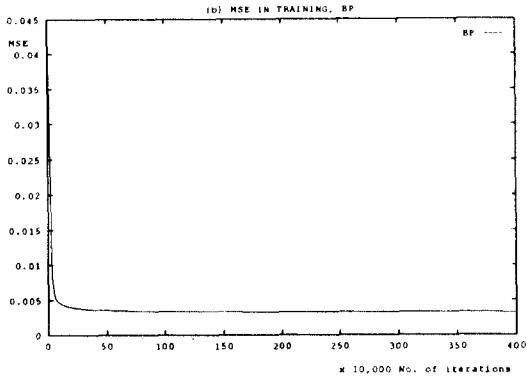


그림 9. BP에 의한 비등변 6계단 함수의 근사화 결과와 훈련중 MSE의 변화
 Fig. 9 Approximation of irregular 6 steps by standard BP and change in MSE during training

그림 9는 BP를 이용하여 근사화한 결과이다. 표 2는 AMCT와 BP의 훈련 결과를 요약한 것이다. 4.1절의 실험에서와 마찬가지로 두 결과 간에 MSE의 차이는 적은 편이었다.

4.3 2차원 등변 10계단 함수

이 실험은 AMCT의 2차원 불연속 함수에 대한 근사화 능력을 검증하기 위한 것이다. 훈련 데이터의 샘플수는 2,000개(100×20 격자점)이며, 격자점간 간격은 x방향으로 0.01, y방향으로 0.05이다. 각 모듈의 입력 공간 창은 계단을 오르는 방향(x방향)의 직각으로 뻗는 띠(strip)의 형태로 하였다. 최초의 입력 공간 창의 폭은 0.07(7개의 격자점을 포함)이었으며 확장 방향은 띠의 폭이 늘어나는 방향이다. 각 모듈의 구조는 입력층 노드 2개, 은닉층 노드 1개, 출력층 노드 1개로 각 모듈은 한 개의 불연속성을 근사화하도록 하였다. 훈련에 의하여 생성된 모듈의 수는 9개로 불연속성의 수와 같다. 그림 10(a), (b), (c)는 각각 목표 함수, AMCT를 이용한 근사화 결과, BP를 이용한 근사화 결과로, AMCT의 근사화 결과가 BP보다 우수함을 쉽게 알 수 있다. 표 3은 훈련 결과의 비교로 역시 AMCT의 우수함을 잘 나타내고 있다.

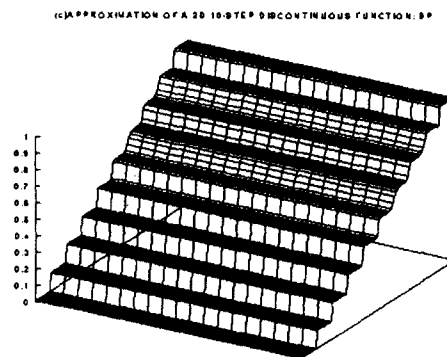
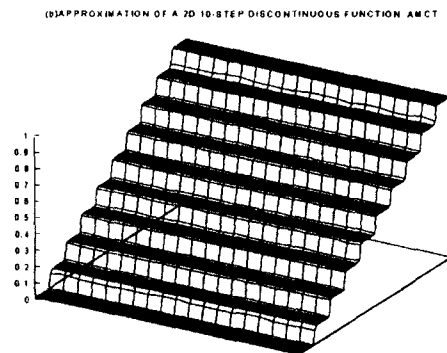
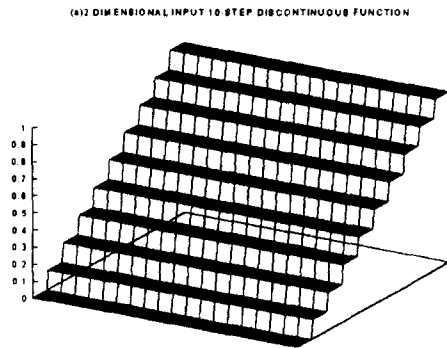


그림 10. AMCT와 BP를 이용한 2차원 등변 10계단 함수의 근사화 결과
 Fig. 10 Approximation of 2D regular 10 steps by AMCT and BP

표 3. AMCT와 BP의 2차원 등변 10계단 함수 근사화에서의 성능 비교

Table 3. Comparison between AMCT and BP for 2D regular 10 steps

Learning Rules	AMCT	Back-prop	Improvement
MSE	0.000011	0.000048	436.3%
No. of Weight Changes	1,259,514,800	76,200,000,000	6049.9%
No. of Iterations	179,242,400	600,000,000	

4.4 2차원 등변 피라미드형 5계단 함수

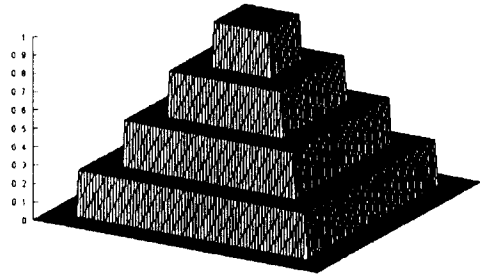
이 실험에서 사용된 함수는 3차원 표면 복구 알고리즘에서 자주 사용되는 화상속의 물체와 비슷한 것으로, 서로 4개의 수직으로 교차하는 불연속성이 5개 층을 이루어 피라미드 형태를 형성한다. 훈련 데이터 샘플수는 10,000개로 100×100 크기의 격자 영역을 이룬다. 각 불연속성은 x축 또는 y축에 수평한 방향으로 위치하며 층간 간격은 0.1 또는 10 데이터 포인트이다. 각 모듈은 4개의 불연속성으로 이루어진 피라미드의 한 개층을 근사화하는 것을 가정하여 입력 공간차를 근사화할 수 있는 구조를 갖도록 했다. 입력 공간창의 크기는 8×8 격자점 크기이며, 시작 위치는 (0.5, 0.5)으로 하였다. 각 모듈은 2개의 입력 노드, 4개의 은닉 뉴런을 갖는 은닉층과 단일 출력 노드로 구성되었다. 특히, 근사화된 4개의 층을 합할 경우 본래의 피라미드의 복원이 가능하지만 각각의 불연속성의 합을 구해서는 같은 결과를 얻을 수 없다. 따라서, 여기서는 통합 신경망을 구성할 때 모듈의 출력 노드들로 제2의 은닉층을 형성하도록 하였으며 제 1, 제2 은닉층간의 연결 가중치는 모듈의 은닉층과 출력층간의 연결 가중치로 초기화 하였다.

표 4. AMCT와 BP의 2차원 등변 피라미드형 함수 근사화에서의 성능 비교

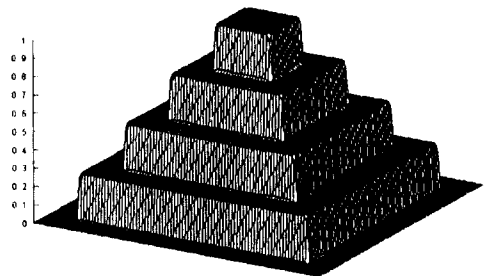
Table 4. Comparison between AMCT and BP for 2D regular pyramid

Learning Rules	AMCT	Back-prop	Improvement
MSE	0.0001	0.000642	642.0%
No. of Weight Changes	2,086,619,296	60,500,000,000	2899.4%
No. of Iterations	219,843,488	1,000,000,000	

2 DIMENSIONAL INPUT 5 STEP DISCONTINUOUS FUNCTION



APPROXIMATION OF A 5-STEP DISCONTINUOUS FUNCTION AMCT



APPROXIMATION OF A 5 STEP DISCONTINUOUS FUNCTION BP

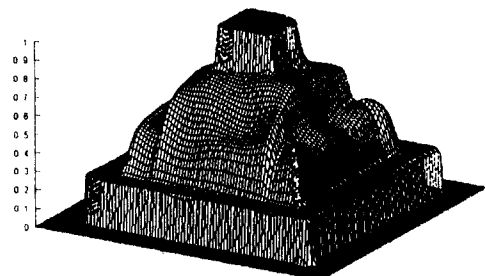


그림 11. AMCT와 BP를 이용한 2차원 등변 피라미드형 함수의 근사화 결과

Fig. 11 Approximation of 2D regular pyramid by AMCT and BP

4.5 불연속면과 곡면을 포함하는 2차원 함수

이 실험에서는 3면의 불연속면과 1면의 완곡면으로 이루어진 2차원 함수의 근사화를 시도했다. 훈련 데이터는 40×40 크기의 입력 공간 내의 샘플들로 이루어져 있다. 그림 12(a)는 목표 함수를 도식한 것이다. 이 실험은 불연속성과 완곡면이 모두 포함된 함수에 대한 AMCT의 근사화 능력을 테스트하기 위한 것이다. 따라서, 여기서는 AMCT의 근사화 결과만을 다루었으며, BP와의 성능 비교는 하지않았다. 그림 12(b)는 근사화 결과로, AMCT의 완곡 표면과 불연속 표면을 함께 포함한 함수에 대한 근사화 능력을 알 수 있다.

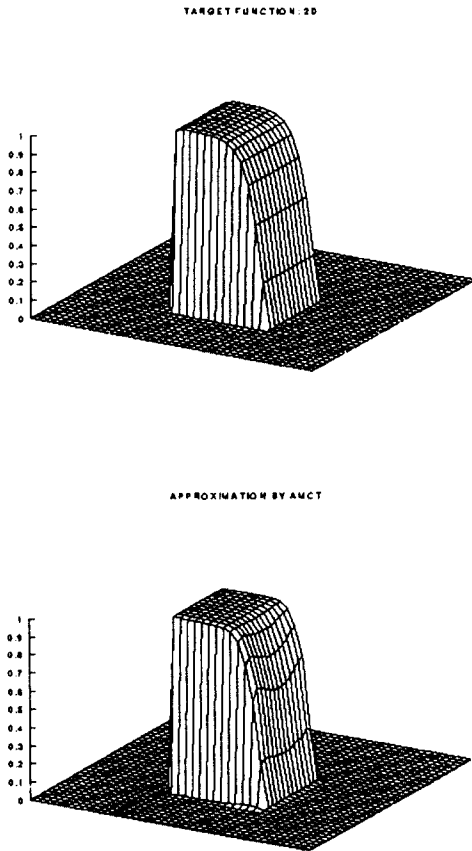


그림 12. AMCT에 의한 불연속면과 곡면을 포함하는 2차원 함수의 근사화 결과
 Fig. 12 Approximation of a 2D function including discontinuity and curved surface by AMCT

V. 결 론

기존의 BP 알고리즘으로 훈련한 다층퍼셉트론을 universal approximator로서의 함수 근사화 능력에도 불구하고 다수의 불연속성을 갖는 함수의 근사화에 한계를 가지고 있다. 본 연구에서 언급한 BP의 몇가지 문제점으로는 (1) 특히, MSE의 크기가 작은 경우 국부 최소점으로부터 탈출이 불가능한 점 (2) 훈련 알고리즘의 국부적 제어 기능의 부재로 인하여 국부 근사화(local approximation)가 불가능한 점 (3) 은닉 노드나 연결선(connection)등의 자유 파라미터의 수를 조절할 수 없음 등이 있다. 위에서 언급한 문제점중 첫 번째 항목에 의한 결과로는 불연속성에서의 스텝의 높이가 작을 경우, 목표 함수의 전역적 형태를 잘 따라가는 근사 함수를 얻을 수는 있으나 불연속성들은 무시한다는 것이다. 불연속성은 급격히 증가하는 시그모이드에 의하여 근사화가 가능하다. 그러므로 이론적으로 입력 공간내의 불연속성의 수 만큼의 시그모이드가 필요하다. 그러나, 이 숫자에 대한 사전 지식을 가질 수 없다. 또한, BP 알고리즘은 은닉 노드의 수를 적응적으로 조절하는 기능도 없다. 높은 정확도를 갖는 불연속 함수의 근사화는 ill-posed 문제에 속한다. ill-posed 문제를 풀기 위해서는 문제를 well-posed 상태로 변환해야 한다.

Regularization은 사전 지식을 이용하여 가능한 해의 범위에 제한을 가하는 방법이다. 이 논문에서는 다수의 제한적인 근사화 능력만을 갖는 신경망 모델을 이용하여 국부적 근사화를 실현하고 이들을 통합하여 최종 신경회로망을 구성하는 알고리즘에 기반한 regularization의 방법을 소개하였다. 이 알고리즘은 기존의 BP 알고리즘 보다 근사화 오차와 훈련 속도 면에서 우수한 성능을 보이며, 불연속성의 개수등에 영향을 받지 않는다. 몇가지의 실험을 통하여 AMCT와 BP의 불연속 함수 근사화에 있어서의 성능이 우수함을 보였다. 그러나 MSE를 비교해보면 AMCT의 MSE가 BP의 MSE보다 결코 현저하게 낮지 않음을 알 수 있다. 예를 들면, 표 4의 MSE 값들은 크기가 너무 작아 비교의 의미가 별로 없다. 그러나, 근사화 결과로부터 오차, 즉 $\{e_i = (o_i - t_i) : i = 1, \dots, n\}$ 를 검사해보면 불연속성 주위에서 AMCT에 의한 이 오차의 크기는 BP에 비하여 현저히 낮음을 알 수 있다.

본 논문에서 소개한 AMCT 알고리즘은 표면 복원, 자동 제어등과 같이 시스템의 특성이 불연속성을 포함하는 함수의 형태인 문제를 많이 다루는 분야에서 유익하게 응용될 수 있을 것이다. 앞으로 해결되어야 할 문제로는 국부 근사화를 위한 입력 공간 영역의 확장 방법을 모든 문제에서 일반적으로 적용될 수 있도록 하는 것이다.

참 고 문 헌

1. M. Bertero, T. Poggio, and V. Torre. "Ill-posed problems in early vision," Proceedings of IEEE, vol. 76, pp. 221-250, 1988.
2. C. Cybenko, "Approximation by superpositions of a sigmoidal function," Math. Control, Signals and Systems, vol. 2, pp. 303-314, 1989.
3. W. Grimson, From images to Surfaces: A computational study of the human early visual system, MIT Press, Cambridge, MA, 1981.
4. J. G. Harris, "The coupled depth/slope approach to surface reconstruction," Technical Report, vol. 908, MIT AI Lab, 1986.
5. R. Hecht-Nielsen. Neurocomputing, Addison-Wesley, Reding, Messachusetts, 1989.
6. R. Hecht-Nielsen. "Theory of the back-propagation neural network," In Proceedings of the International Conference on Neural Networks, June 1989.
7. J. Hopfield and D. Tank, "Computing with neural circuits: A model," Sciences, vol. 235, pp. 1226-1227, March 1986.
8. K. Hornick, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," Neural Networks, vol. 2, pp. 359-366, 1989.
9. C. Koch, J. Marroquin, and A. Yuille. "Analog "neuronal" networks in early vision," Proc. Natl. Acad. Sci., vol. 83, pp. 4263-4267, June 1986.
10. A. N. Kolmogorov. "On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition," Doki. Akad. Nauk USSR., vol. 114, pp. 953-956, 1957.
11. A. Lapedes and R. Farber. "How neural nets work," In D. Anderson, editor, Neural Information Processing Systems, pp. 442-456, New York, 1988. American Institute of Physics.
12. J. Mao and A. K. Jain. "Regularization techniques in artificial neural networks," In Proceedings of World Congress of Neural Networks, July 1993.
13. J. Marroquin, S. Mitter, and T. Poggio. "Probabilistic solution of ill-posed problems in computational vision," J. of the American Statistical Association, vol. 83, no. 397, pp. 76-89, March 1987.
14. J. Moody. "The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems," In J. Moody, S. Hanson, and R. Lippmann, editors. Advances in Neural Information Processing Systems 4, San Mateo, CA, 1992. Morgan Kaufmann Publishers.
15. T. Poggio, V. Torre, and C. Koch, "Computational vision and regularization theory," Nature, vol. 317, pp. 638-643, 1985.
16. D. Rumelhart, G. Hinton, and R. Williams. "Learning internal representations by error propagation," In Parallel Distributed Processing: Exploration in the microstructure of cognition, vol. 1, pp. 318-362, MA, 1986.
17. E. Sontag. "Capabilities of four-vs. three-layer nets, and control applications," In Proceedings of conference on Information Science and Systems, Johns Hopkins University, March 1991.
18. D. Tank and J. Hopfield, "Simple "neural" optimization networks: An a/d converter, signal, decision circuit, and a linear programming circuit," IEEE Trans. on Circuits and Systems, vol. 33, no. 5, pp. 533-541, May 1986.
19. A. Tikhonov and V. Arsenin. Solutions of ill-posed problems, Winston Sons, Washington D.C., 1977.



李 赫 峻(Hyukjoon Lee) 정회원

1987년 8월:미국 University of Michigan, Ann Arbor 학사(전산학)

1989년 6월:미국 Syracuse University 석사(전산학)

1993년 12월:미국 Syracuse University 박사(전산학)

1994년 3월~1996년 2월:삼성전자(주) 멀티미디어연구소 선임연구원

1996년 3월~현재:광운대학교 컴퓨터공학과 전임강사
※주관심분야:신경회로망, 인공지능, 컴퓨터그래픽스, 가상현실, 분산컴퓨팅