

사전 정보를 이용한 다층신경망의 설계

(A Design Method for a New Multi-layer Neural Networks Incorporating Prior Knowledge)

金 丙 鎬*, 李 志 弘**

(Byoung Ho Kim and Ji Hong Lee)

要 約

본 논문은 주어진 학습패턴의 분포를 이용하여 다층신경망을 설계하는 방법에 대하여 기술한다. 신경망의 크기와 연결강도 등을 포함한 신경망의 구조는 주어진 학습패턴의 특징점 추출에 의해 초기화 된다. 이러한 신경망은 개선된 오차역전달 알고리즘에 의해 학습된다. 결론적으로 제안된 방법은 무작위로 초기화한 다층신경망의 학습에 비해 학습속도 개선의 장점을 가진다. 제안된 방법의 효용성을 보이기 위해, 2차원 연속함수의 근사화에 적용한 결과를 제시하였다.

Abstract

This paper presents the design consideration of the MFNNs(Multilayer Feed forward Neural Networks) based on the distribution of the given teaching patterns. By extracting the *feature points* from the given teaching patterns, the structure of a network including the network size and interconnection weights of a network is initialized. This network is trained based on the modified version of the EBP(Error Back Propagation) algorithm. As a result, the proposed method has the advantage of learning speed compared to the conventional learning of the MFNNs with randomly chosen initial weights. To show the effectiveness of the suggested approach, the simulation result on the approximation of a two dimensional continuous function is shown.

1. 서 론

인공신경망이 등장한 이래, 다층신경망은 이론 및

응용 분야에 걸쳐 다양하게 연구되고 있는데, Rumelhart등^[1]에 의해 오차역전달(error backpropagation)방법이 고안되면서 초기에 문제시 되었던 exclusive OR 문제는 물론, 복잡한 비선형 함수도 모사할 수 있게 되었으며, 각종 시스템 모델링, 결함 검출, 인식 및 제어등의 분야에 널리 이용되고 있다. 또 Nielsen^[2]은 다층신경망의 한계 및 근사능력의 해석에 대해 하나씩의 입력층, 출력층 및 히든(hidden)층을 갖는 인공신경망이 히든(hidden)층의 유닛의 개수를 늘려감에 따라 어떤 함수도 주어진 오차 이내로 근사시킬 수 있음을 보였으며, Stinchcombe

*正會員, 金烏工科大學校 電子工學科

(Dept. of Elec. Eng., Kumoh Nat'l Univ. of Technology)

**正會員, 金烏工科大學校 電子制御工學科

(Dept. of Control Eng., Kumoh Nat'l Univ. of Technology)

接受日字 : 1993年 1月 28日

등^[3,5]과 Hornik^[4]등도 이러한 내용을 보다 수학적으로 증명하였다.

최근에는 주어진 학습 패턴을 효과적으로 구현하기 위한 최적의 다층신경망에 대한 연구가 중요한 연구 과제로 수행되고 있는데, Sartori등^[6]은 학습해야 하는 입출력쌍 즉, 학습 패턴이 사전에 정해져 있는 경우에 신경망의 히든(hidden)유닛 하나하나가 입출력쌍에 대응되도록 시그모이드(sigmoid)함수의 기울기, 중앙값등을 계산해 내고, 이들 시그모이드(sigmoid)함수의 조합에 의해 가우스 함수꼴의 출력 특성^[7]을 내도록 구성하여, 주어진 입출력쌍의 정보로부터 히든(hidden)-출력층 사이의 연결강도를 결정함으로써, 학습이 전혀없이 주어진 입출력 대응을 오차없이 구현할 수 있는 다층신경망을 제안하였다.

그런데 Sartori등의 방법은 학습이 불필요한 대신 학습용으로 주어 입출력 쌍만큼의 히든(hidden) 유닛이 필요하게 되어 학습 패턴이 광범위한 경우 시스템 구성이 대단히 복잡해지게 되므로 비효율적이며, 학습 포인트에서는 오차없는 재생이 보장되는 반면 그 사이의 위치에서는 오차를 예상하기 힘든점등의 단점이 있다. 이러한 문제점에 착안하여 Sartori^[6]의 무학습 개념을 이용하여 불필요한 학습과정을 줄이고, 추가적인 학습 과정을 통해 한층 더 학습 효과를 개선할 수 있는 구조의 다층신경망을 생각해 보았다.

본 논문에서는 주어진 학습 패턴의 분포 특징을 참조한 다층신경망 설계 방법을 제안한다. 먼저 주어진 학습 패턴을 조직적으로 분류하여 학습 패턴을 이루는데 중요한 역할을 하는 특징점들을 찾아내고, 이러한 특징점들을 활용하여 Sartori등의 방법으로 계수들을 계산에 의해 구한 다음, 선택된 특징점들에 대한 대응을 수행하도록 다층신경망의 연결 강도 및 관련 계수들을 초기화한 후, 그 이외의 포인트들에 대해서는 추후 학습을 통해 주어진 학습 패턴에 더욱 근사한 대응을 수행하도록 한다. 즉, 다층신경망의 크기, 즉 층의 갯수 및 각 층의 프로세싱 유닛의 갯수등과 각 프로세싱 유닛의 활성화함수의 비선형도 등을 전체 학습 패턴중에서 선별된 특징점들에 의해 결정하며, 이렇게 결정된 초기상태에서 전체 학습 패턴에 의한 학습 과정을 거쳐 원하는 목적을 수행하도록 한다. 여기서 특징점이란 주어진 학습 패턴을 구성함에 있어 중요한 역할을 하는 부분으로 primary points, secondary points 및 boundary points로 분류되며, 학습의 초기상태를 가급적 해 근처에 위치시키는 정보로 이용된다. 특징점에 대한 구체적인 설명은 2장에서 하기로 한다. 제안된 신경망은 [8]에

서 제안된 바 있는 1 차원 학습 패턴(입력이 하나인 경우)에 적용된 이론을 더욱 확장하여 학습 패턴의 특징점 위치에 종형(bell type)의 가우스 함수를 배치함으로써 다차원 학습 패턴(입력이 여러개인 경우)에 대해서도 한층 더 학습 효과를 개선할 수 있도록 하였다. 이번에는 입력이 하나인 경우와는 달리, 입력이 여러개인 경우이기 때문에 완전히 다른 과정을 통해 특징점을 추출하게 되어 [8]에서의 결과(입력이 하나인 경우)로써 입력이 여러개인 경우로 바로 확장할 수가 없다. 즉 [8]에서 다룬 경우와 목적을 공유할 뿐 신경망의 구조, 신호처리 방법, 특징점들의 의미, 추출방법 등에 있어서 상당한 차이가 있다. 또한 학습 법칙은 [1]의 유도과정을 이용하여 유도할 수 있으며, 히든(hidden)층에 곱하기 유닛이 등장함에 따라 기존의 학습 법칙과는 약간 다르게 표현되며, 그 결과식은 2장에서 제시한다. 결론적으로 본 논문에서는 다층신경망의 크기와 근사화 능력이 근본적으로 특징점들의 분포 및 갯수에 의해 상당부분 결정된다는 관찰을 근거로, 학습 패턴에 따라 특징점을 적절히 선정하여, 보다 효율적으로 원하는 학습 패턴의 근사화를 이룰 수 있음을 보인다.

이어서 2장에서는 다층신경망의 설계에 관한 알고리즘을 설명하고, 3장에서는 적용예를, 4장에서는 검토 및 결론을 맺도록 하겠다.

II. 신경망의 설계

일반적으로 다층신경망을 이용하는 많은 경우에는 학습을 시키기 위해 입출력 집합이 미리 주어져 있게 된다. 이러한 경우에 무계획하게 학습시키는 것보다 학습용으로 주어진 입출력 패턴의 특징점들, 예를 들어 입출력 패턴중 유사한 패턴, 변화량이 일정한 패턴, 급격히 변화하는 패턴, 초기값, 마지막 값 등의 분포 특징들을 적절히 이용한다면 보다 효과적인 학습을 기대할 수 있다. 이는 사람의 경우에 학습해야 할 내용이 많이 주어져 있는 경우, 학습할 내용을 사전에 유사한 내용, 주기적인 내용, 아주 판이하게 다른 내용등으로 분류하여 적절한 순서를 정하고, 각각의 특징에 맞게 학습의 방법을 정한다는 직관적인 방법과도 기본적인 개념을 공유한다고 볼 수 있다. 이러한 개념을 근간으로 학습 입력이 하나인 경우, 즉 $R^1 \rightarrow R^1$ 대응에 대해서는 [8]에서 제안된 바 있다.

본 논문에서는 주어진 학습 입력이 두 개 이상인 경우, 즉 학습 패턴이 2 차원 이상인 경우($R^m \rightarrow R^n$)에 대해 한층 더 학습 효과를 개선할 수 있는 신경망 설계 알고리즘에 대해 생각해 보았다. 학습 패

턴이 1 차원에서 2 차원으로 확장됨에 따라 신경망의 구성상 크게 다른점은 곱하기 유닛이 나타난다는 점이며, 따라서 곱하기 유닛이 존재하는 층에서는 기존의 오차역전달(error backpropagation) 학습 법칙이 수정된 형태로 적용된다. 여기서는 주어진 학습 패턴의 특징점을 이용하여 신경망을 설계하게 되는데, 이번에는 입력이 하나인 경우와는 달리, 입력이 여러개인 경우이기 때문에 완전히 다른 과정을 통해 특징점을 추출하게 된다. 여기서 특징점(feature point)이란 주어진 학습 패턴을 이루는데 중요한 역할을 하는 점들로 나중에 설명하게 될 인공신경망 설계의 근간을 이루는 정보로서 다음 두가지 측면에서 그 유용성이 있다. 첫째, 시그모이드(Sigmoid)함수는 단조변화(단조증가 또는 단조감소)함수이고, 연결강도와 바이어스(bias)등은 이 함수의 높이, 기울기, 위치 등을 결정하지만 궁극적으로 단조변화함수란 특징을 바꾸지는 못한다. 그러므로 학습 패턴의 변화율의 부호가 변하는 부분은 여러개의 시그모이드(Sigmoid)함수가 조합되어야만 그러한 부분을 담당(cover)할 수 있게 된다. 이와같은 근거로 학습 패턴의 변화율의 부호가 변하는 부분을 특징점으로 잡고그 점에서 이웃하는 부분을 단조증가 또는 단조감소의 특성을 갖는 히든(hidden)유닛이 대응하도록 신경망을 초기화 한후, 전체적으로 학습과정을 수행하면 보다 효과적인 학습이 이루어지는 장점이 있다. 둘째, 선별된 특징점의 개수와 특징점으로 분류된 학습 포인트의 입출력 데이터(data)로부터 신경망의 크기와 연결강도등의 관련 계수들을 각각 결정할 수 있다.

그러면 학습 입력이 두 개 이상인 경우에 대해 학습 패턴의 특징점들을 추출하는 방법, 추출된 특징점들에 의한 신경망의 크기 결정 및 각 층의 연결강도와 바이어스(bias) 등 계수의 초기화 방법에 대해 기술하고, 수정된 형태의 학습 법칙을 소개한다.

1. 특징점 추출 방법

입력의 개수를 m, 출력의 개수를 n이라 하자. 즉, 학습을 위해서 m 차원 입력 벡터와 n 차원 출력 벡터 사이의 대응이 p 개의 학습 패턴으로 주어지고, 이러한 전체 학습 패턴은 각 성분별로 (p₁, ..., p_k, ..., p_m)개의 학습 입력 패턴을 가지고 있다고 가정한다. 그러면 각 성분별로 (j₁, ..., j_k, ..., j_m)번째의 학습 입력에 의해 주어지는 전체 학습 패턴의 (j₁, ..., j_k, ..., j_m)번째 학습 입력을 다음과 같이 나타내고,

$$v(j_1, \dots, j_k, \dots, j_m) = (v_1(j_1), \dots, v_k(j_k), \dots, v_m(j_m))$$

$$j_1=1, \dots, p_1, \dots, j_k=1, \dots, p_k, \dots, j_m=1, \dots, p_m$$

(j₁, ..., j_k, ..., j_m)번째 학습 입력에 대응하는 n 차원 학습 출력을

$$d(j_1, \dots, j_k, \dots, j_m) = (d_1(v(j_1, \dots, j_k, \dots, j_m)), \dots, d_n(v(j_1, \dots, j_k, \dots, j_m)))$$

라고 하자. 이때 d(j₁, ..., j_k, ..., j_m)의 차수는 출력 유닛의 개수와 같고, 각 성분 방향의 k 번째 입력 패턴은 식(1)과 같이 일정한 간격으로 분포되어 있으며,

$$v_k(j_k+1) - v_k(j_k) = v_k(j_k+2) - v_k(j_k+1), 1 \leq k \leq m \quad (1)$$

식(2)와 같이 크기에 따라 순서가 매겨져 있다고 가정한다.

$$v_k(j_k+1) > v_k(j_k), 1 \leq k \leq m \quad (2)$$

실사 데이터가 이러한 형태가 아니더라도 정리하여 이러한 규칙으로 재배열하는 일은 대부분의 경우에 그리 어려운 일이 아니므로, 무리한 가정이 아니라고 할 수 있다. 그러면 다음의 검사하는 과정을 통해 각 성분의 학습 입력 패턴이 무학습 신경망을 구성하기 위해 필요한 특징점인가를 확인하고, 전체 학습 입력 공간내에서 특징점들의 집합(σ)을 추출한다. 효과적인 설명을 위해 m=2, n=1 인 그림 1의 예에 대해 alto리즘을 적용하면서 설명한다.

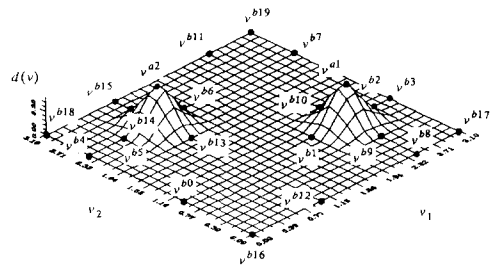


그림 1. R²→R¹ 학습패턴의 특징점 추출에 : m=2, n=1의 경우

Fig. 1. Extracting feature points of a training pattern: m=2, n=1 case.

(s0) σ를 공집합으로 한다.

(s1) 주어진 학습 패턴에 대해 m 차원 입력 벡터의 각 성분 방향으로 δ 값 이상의 변화를 동시에 만족하는 점들을 찾는데, 여기서는 δ를 0으로 하여 학습 패턴의 변화율의 부호가 변하는 점(극점)에 해당

하는 학습 포인트를 찾는다. 즉 $1 \leq r \leq n$ 의 r 에 대해 다음의 조건쌍을 만족하는 점 $(v_1(j), \dots, v_k(j), \dots, v_m(j_m))$ 들을 찾는다.

$$\begin{aligned} d_r(v_1(j_1), \dots, v_k(j_k+1), \dots, v_m(j_m)) - d_r(v_1(j_1), \dots, v_k(j_k), \dots, v_m(j_m)) &\geq \delta \text{ and} \\ d_r(v_1(j_1), \dots, v_k(j_k), \dots, v_m(j_m)) - d_r(v_1(j_1), \dots, v_k(j_k-1), \dots, v_m(j_m)) &< \delta \\ \text{for all } 1 \leq k \leq m & \quad (3\text{-}(a)) \end{aligned}$$

또는

$$\begin{aligned} d_r(v_1(j_1), \dots, v_k(j_k+1), \dots, v_m(j_m)) - d_r(v_1(j_1), \dots, v_k(j_k), \dots, v_m(j_m)) &< \delta \text{ and} \\ d_r(v_1(j_1), \dots, v_k(j_k), \dots, v_m(j_m)) - d_r(v_1(j_1), \dots, v_k(j_k-1), \dots, v_m(j_m)) &\geq \delta \\ \text{for all } 1 \leq k \leq m & \quad (3\text{-}(b)) \end{aligned}$$

이 점들을 $v^q = (v_1^q, \dots, v_k^q, \dots, v_m^q), v^{q2} = (v_1^{q2}, \dots, v_k^{q2}, \dots, v_m^{q2}), \dots$ 라 한다. 이렇게 선정된 점들을 *primary points*라 하고, 집합 ϕ 에 포함시킨다. 이 점들은 각 입력성분 방향으로 기울기의 변화가 어느 정도(δ) 이상이거나 부호가 바뀌는 점들을 의미한다. 이때 δ 값은 특징점을 판별하는데 있어서 중요한 판별치이고, 특징점은 δ 에 의해 추출되는데, 아직까지 δ 를 선정하는 일반적인 방법은 연구되지 않았다. 그러나 직관적으로 δ 에 따라 특징점의 개수가 작거나, 많이 나타나게 된다. 즉 δ 를 크게 잡으면 특징점의 개수는 작고, δ 를 작게 잡으면 특징점의 개수는 많아지게 된다. 그림 1의 경우에는 v^1, v^{2} 의 위치가 선정된다.

(s2) primary point $v_{wq} = (v_1^q, \dots, v_k^q, \dots, v_m^q), q=1, 2, \dots$ 의 성분중 하나인 v_k^q 를 제외한 나머지 성분 $v_i^q, i \neq k$ 을 고정시키고, 전체 학습 패턴에 대해 $v_k = v_k^q$ 로 표시되는 단면을 따라 변화율이 δ 이상인 점들을 찾는다. 즉, 고정된 k 에 대해 식(3)의 조건을 만족시키는 점을 찾는다. 그림 1에서는 v^1 점에 대해 $k=1$ 인 경우에 v^1, v^{2} 점이 찾아지고, v^{2} 인 경우에 v^{6}, v^{7} 점이 찾아진다. 같은 과정을 $k=2$ 에 대해 수행하여 $v^{9}, v^{10}, v^{13}, v^{14}$ 등의 점들을 찾는다. (s2) 과정을 통해 찾은 점들을 *secondary points*라 하고, 집합 ϕ 에 포함시킨다. 단, secondary points 추출시 항상 나타나게 되는 primary point는 secondary points에서 제외한다.

(s3) primary point를 포함하는 각 성분 방향의 초기점과 마지막점 및 전체 학습패턴의 각 모서리점들을 포함시킨다. 즉 그림 1에서는 v^{1} 점에 대해 $k=1$ 인 경우에, v^{6}, v^{7} 점이 찾아지고, v^{12} 인 경우에 v^{14}, v^{17} 점이 찾아진다. 같은 과정을 $k=2$ 에 대해 수행하여 $v^{18}, v^{11}, v^{12}, v^{15}$ 등의 점들을 찾고, 전체 학습 패턴의 각 모서리점 $v^{16}, v^{17}, v^{18}, v^{19}$ 을 포함시킨다. (s3) 과정을 통해 찾은 점들을 *boundary points*라 하고, 집합 ϕ 에 포함시킨다. 이러한 점들

은 학습패턴의 시작과 끝에 관한 정보를 의미하므로 이점들을 특징점에 포함시키지 않으면 초기화된 신경망의 시작점과 끝점에서 큰 오차가 유발된다. 이러한 측면에서 학습 패턴의 시작점과 끝점, 즉 boundary points들이 특징점으로써 꼭 필요한 것이다.

2. 신경망의 크기, 각 층의 특성 및 관련계수의 초기화

본 절에서는 1 절에서 찾아진 특징점들을 이용하여 신경망의 크기를 결정하고, 각 층의 특성 및 신경망의 각 층에 관련된 계수들의 초기화에 대해 설명한다.

먼저 신경망의 크기는 특징점들에 의해 다음과 같이 결정된다. 1절에서 각 성분별로 찾아진 특징점들의 갯수를 각각 $\tilde{p}_1, \dots, \tilde{p}_m$ 라 하면, 입력층은 m 개, 출력층은 n 개로 결정되고, 1번 히든(hidden)층과 2번 히든(hidden)층은 각각 m 개의 그룹(group)으로 구성되는데, 1번 히든(hidden)층의 각 그룹(group)은 $\tilde{p}_m - 1$ 개, 2번 히든(hidden)층의 각 그룹(group)은 \tilde{p}_m 개의 유닛을 갖게 되고, 3번 히든(hidden)층은 식(4-a)에 의해 유닛의 갯수 및 번지가 결정된다.

$$\begin{aligned} \tilde{p} &= (\tilde{j}_1 - 1) \left(\prod_{k=2}^m \tilde{p}_k \right) + (\tilde{j}_2 - 1) \left(\prod_{k=3}^m \tilde{p}_k \right), \\ & \quad (\tilde{j}_{m-1} - 1) \tilde{p}_m + \tilde{j}_m \quad (4\text{-}(a)) \\ 1 \leq \tilde{j}_1 \leq \tilde{p}_1, 1 \leq \tilde{j}_2 \leq \tilde{p}_2, \dots, 1 \leq \tilde{j}_{m-1} \leq \tilde{p}_{m-1}, 1 \leq \tilde{j}_m \leq \tilde{p}_m \end{aligned}$$

여기서 $\tilde{p}_1, \tilde{p}_2, \dots, \tilde{p}_{m-1}, \tilde{p}_m$ 는 각 성분 방향으로 찾은 특징점들의 갯수이며, 결국 2번 히든(hidden)층의 유닛의 갯수가 되고, $\tilde{j}_1, \tilde{j}_2, \dots, \tilde{j}_{m-1}, \tilde{j}_m$ 는 각각 2번 히든(hidden)층의 1, 2, ..., $m-1, m$ 그룹(group)에 있는 유닛의 번지를 나타낸다. 그리고 \tilde{p} 는 곱하기 유닛의 번지를 나타내며, 다음 범위의 값을 갖는다.

$$\tilde{p} = 1, 2, \dots, \prod_{k=1}^m \tilde{p}_k \quad (4\text{-}(b))$$

이러한 일반적인 학습 패턴($R^m \rightarrow R^n$)에 대한 다층신경망 그림 2와 같은 구조를 갖는다.

다음은 신경망의 각 층의 특성에 대해 설명한다. 입력층 유닛 및 출력층 유닛의 특성은

$$f(u) = u \quad (5)$$

이고, q 번째 학습 입력에 대한 1번 히든(hidden)층, 2번 히든(hidden)층 그리고 출력층의 총화(net total)입력은

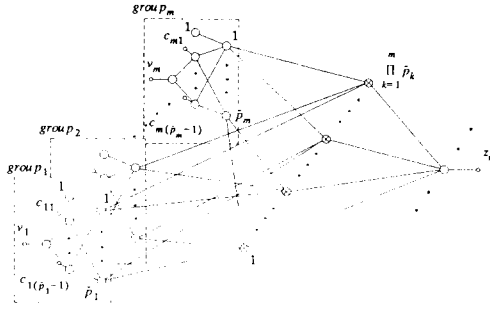


그림 2. 다차원 신경망 ($R^m \rightarrow R^n$)의 일부
Fig. 2. A part of multi-dimensional network: $R^m \rightarrow R^n$ case.

$$net_{qi} = \sum_i w_{ji} o_{qi} \tag{6}$$

이며, 1 번 히든(hidden)층의 활성화함수는

$$f(u, c_j, s_j) = \frac{1}{1 + e^{-s_j(u - c_j)}} \tag{7}$$

이다. 여기서 s_j 는 시그모이드(sigmoid)함수의 기울기를 의미하며, 입력층과 1번 히든(hidden)층 사이의 연결강도에 의해 등가적으로 보정된다. c_j 는 시그모이드(sigmoid)함수의 위치를 결정한다.

2번 히든(hidden)층은 식(7)을 선형조합한 형태의 가우스(gauss)함수특성을 갖고, 3번 히든(hidden)층은 곱하기 유닛으로 구성되며, 곱하기 함수특성을 갖는다. 특히 입력이 두개인 경우 3번 히든(hidden)층은 그림 3의 2 차원 가우스(gauss)함수 특성을 갖는다.

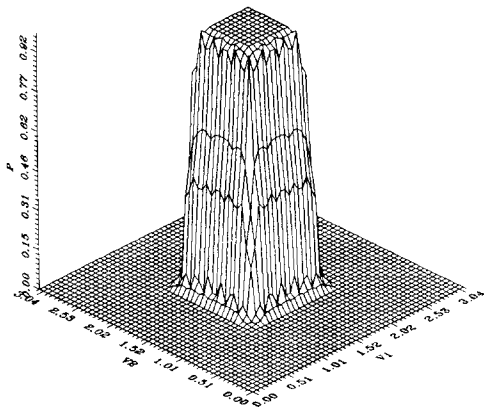


그림 3. 2차원 가우스(gauss) 함수의 예
Fig. 3. A two dimensional Gaussian-type function.

마지막으로 신경망의 초기화 방법에 대해 설명한다.

(i0) 1 절에서 각 성분별로 찾아진 점들의 개수를 $\bar{p}_1, \dots, \bar{p}_m$ 라 한다.

(i1) 1번 히든(hidden)층의 시그모이드(sigmoid)함수의 계수 설정 집합 ϕ 의 원소인 점(각각 m 차원 벡터)들의 j 번째 성분만을 골라내어 크기 순으로 재배열하여 이들을 $\bar{v}_j(1), \bar{v}_j(2), \dots, \bar{v}_j(\bar{p}_j)$ 라 하고, 1 번 히든(hidden)층의 j 그룹(group)의 시그모이드(sigmoid)함수의 중앙값을 다음 식에 의해 구한다.

$$c_{jk} = \frac{1}{2} [\bar{v}_j(k) + \bar{v}_j(k+1)], 1 \leq j \leq m, 1 \leq k < \bar{p}_j \tag{8}$$

시그모이드(sigmoid)함수의 기울기(s)는 궁극적으로 가우스(gauss)함수의 기울기에 관계하여 신경망의 초기 출력을 결정하는데, 몇번의 시도에 의해 특징점 위치에서 정확한 입출력 대응을 수행하도록 값을 정한다.

(i2) 입력층과 1번 히든(hidden)층 사이의 연결강도 W^1 의 초기 설정 : 1번 히든(hidden)층은 m 개의 그룹(group)으로 구성되는데, 각 그룹(group)은 $\bar{p}_m - 1$ 개의 유닛을 갖게 되고, 입력층과 1번 히든(hidden)층 사이의 연결강도는 다음과 같이 설정한다.

$$w_{ji}^{1,1}, \dots, w_{ji}^{1,m} = 1.0, \text{ for all } j, i \tag{9}$$

(i3) 2번 히든(hidden)층과 1번 히든(hidden)층 사이의 연결강도 W^2 의 초기 설정 : 2번 히든(hidden)층은 m 개의 그룹(group)으로 구성되는데, 각 그룹(group)은 \bar{p}_m 개의 유닛을 갖게 되고, 1번 히든(hidden)층과 2번 히든(hidden)층 사이의 연결강도는 다음과 같이 설정한다.

$$w_{ji}^{2,1}, \dots, w_{ji}^{2,m} = \begin{cases} 1.0 & \text{if } j = i + 1 \\ -1.0 & \text{if } j = i \\ 0.0 & \text{if } j \neq i, j \neq i + 1 \end{cases} \tag{10}$$

(i4) 2번 히든(hidden)층과 3번 히든(hidden)층 사이의 연결 강도 W^3 및 바이어스(bias)의 초기 설정 : 2번 히든(hidden)층과 3번 히든(hidden)층 사이의 연결강도는

$$w_{\bar{p}_j}^3 = 1.0 \tag{11}$$

로 하고, 그 이외의 경우는 연결하지 않는다. 바이어

스(bias)는 각 그룹(group) 2번 히든(hidden)층의 1번째 유닛에만 1.0 으로 연결한다.

(i5) 3 번 히든(hidden)층과 출력층 사이의 연결강도 W^4 의 초기 설정 : 편의상 하나의 출력인 경우를 가정하여 설명한다. (i4) 까지의 과정에 의해 3번 히든(hidden) 층의 각 유닛은 특징점들의 집합 ϕ 의 원소 중 하나의 원소가 입력 되었을때만 1의 출력을 내게 된다. 또한, 집합 ϕ 의 학습 패턴에 의해 특징점들의 각 입력에 대한 출력 $d(r)$ 가 이미 정해져 있으므로, ϕ 중의 하나의 원소를 입력으로 했을때, 1 을 출력하는 유닛과 출력 유닛과의 연결강도를 그 입력에 대한 출력값으로 초기화한다. 수식으로 표현하면 다음과 같다.

$$w_{r\bar{p}}^4 = d(r), \quad r=1,2,\dots,\prod_{k=1}^m \bar{p}_k \quad (12)$$

결과적으로 c_k 는 1번 히든층에 배치된 시그모이드(sigmoid)함수의 위치를 결정하고, W^1 은 시그모이드(sigmoid)함수의 기울기를 보정하는 역할을 하며, W^2 는 1 번 히든(hidden)층의 시그모이드(sigmoid)함수를 조합하여 2 번 히든(hidden)층이 가우스(gauss)함수 특성을 갖도록 한다. W^3 은 2 번 히든(hidden)층의 출력이 1 인 번지만 곱해지도록 초기화하여 결국 m 차원 가우스(gauss)함수의 특성을 만들어 내는 역할을 한다. 그리고 W^4 는 특징점으로 분류된 데이터(data)의 출력에 해당하므로 시그모이드(sigmoid)함수의 기울기를 크게(100 이상) 선택하면, 초기에 특징점의 위치에서는 정확한 입출력 대응이 이루어지게 된다. 이렇게 초기화한 후 학습을 수행하면 보다 효율적으로 원하는 학습 결과를 얻을 수 있다.

3. 신경망의 학습 법칙

학습 법칙은 [1]의 유도과정을 이용하였으며, 유도 과정은 생략하고, 그 결과식을 다음에 기술하겠다.

패턴 p 의 j 번째 출력 유닛의 기대값을 t_{pj} , 실제 출력값을 o_{pj} 라 하면, 패턴 p 에 의한 오차를

$$E_p = \frac{1}{2} \sum_j (t_{pj} - o_{pj})^2 \quad (13)$$

으로 하고, 전체 오차를

$$E = \sum_p E_p \quad (14)$$

라 하면, 각 층의 연결강도의 학습 법칙은 다음과 같

이 나타낼 수 있다.

$$\Delta_p w_{jk} = \eta \delta_{pj} o_{pk} \quad (15-(a))$$

$$\delta_{pj} = \begin{cases} (t_{pj} - o_{pj}) f_j'(net_{pj}) & \text{for output units} \\ \prod_{k=1}^{j-1} w_{jk} o_{pk} f_j'(net_{pj}) \sum_k \delta_{pk} w_{kj} & \text{for multiplying units} \\ f_j'(net_{pj}) \sum_k \delta_{pk} w_{kj} & \text{for other units} \end{cases} \quad (15-(b))$$

그리고 이 학습 법칙을 초기화 과정에서 연결된 부분에만 적용하게 된다.

다음장에서는 예제를 통해 제안된 알고리즘을 적용하여 설계한 신경망의 학습 효과에 대해 알아 본다.

III. 적용 예

본장에서는 그림 4의 학습패턴(입력이 두 개, 출력이 하나인 경우)에 대해 II 장의 설계과정을 적용하여 신경망을 설계한 후, 시뮬레이션한 결과에 대해 고찰한다.

학습 패턴은 다음 식으로 하였다.

$$z = (\sin(2x)\cos(x) + x/3)(\sin(2y)\cos(y) + y/3), \quad 0 \leq x \leq \pi, \quad 0 \leq y \leq \pi \quad (16)$$

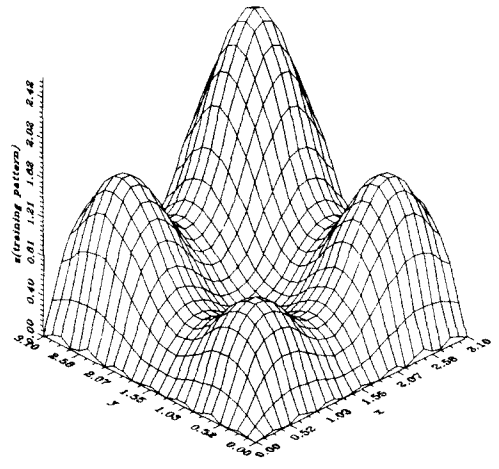


그림 4. 학습 패턴의 예 : $m=2, n=1$ 의 경우
Fig. 4. A training pattern : $m=2, n=1$ case.

이 학습 패턴에 대해서 δ 를 0 으로 하여 찾아진 특징점은 primary point 9 개, boundary point 16 개로 모두 25 개이며, 특징점들의 방향 벡터, 즉 x, y 축 방향에 크기순으로 재배열된 특징점들의 방

향 벡터 성분의 개수는 각각 x 축 방향으로 5 개, y 축 방향으로 5 개이다. 이러한 특징점들을 이용하여 설계한 신경망은 그림 5 와 같다.

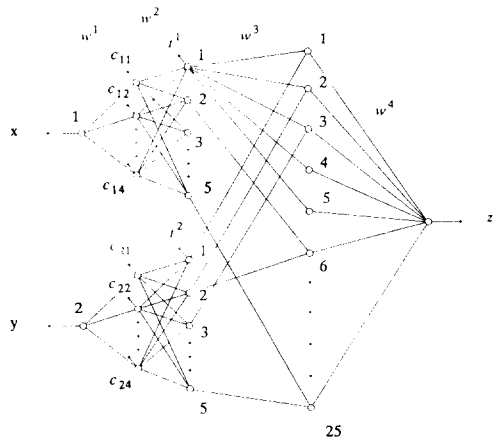


그림 5. $R^2 \rightarrow R^1$ 학습 패턴의 신경망
: $m=2, n=1$ 의 경우
Fig. 5. Designed network : $m=2, n=1$ case.

그림 5 에서 보는 바와 같이 신경망은 입력층 1 개, 히든(hidden)층 3 개, 출력층 1 개의 5 층 신경망으로서, 입력층 유닛 2 개, 1 번 히든(hidden)층 유닛 4 개, 2 번 히든(hidden)층 유닛 5 개, 3 번 히든(hidden)층 유닛 25 개, 출력층 유닛 1 개로 구성되었다. 그리고 각 층의 관련 계수는 \parallel 장의 초기화 과정에 의해 초기화 하였다.

특히 기울기(s)는 기울기(s)의 변화에 의한 무학습 출력을 그림 6-(a),(b)와 같이 관찰한 후, 10 으로 결정하였다.

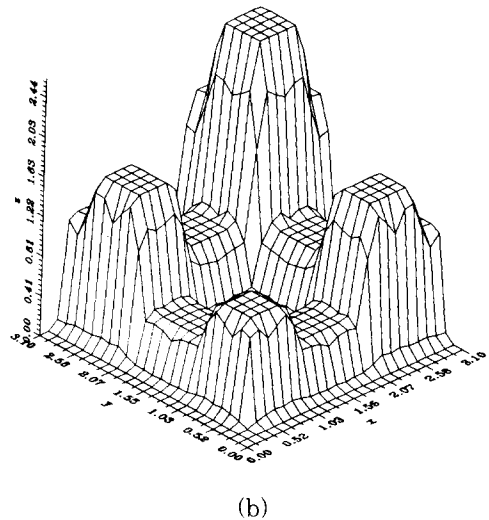
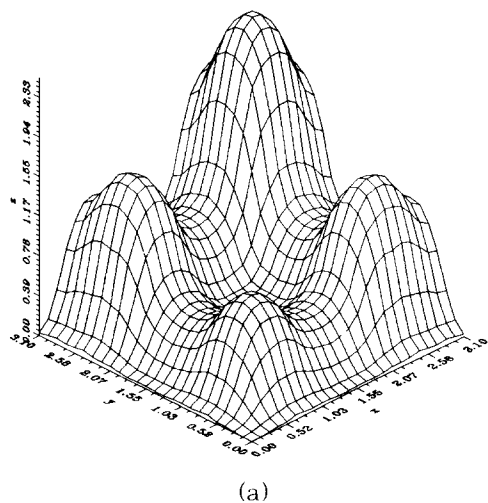


그림 6. 기울기 변화에 의한 무학습 신경망의 출력
(a) $s=10$ (b) $s=100$

Fig. 6. Effects of slope at initialization stage with (a) 10 and (b) 100.

다음은 오차역전달(error backpropagation) 학습 과정을 수행(학습율 0.0015)한 시뮬레이션 결과에 대해 살펴본다. 오차역전달(error backpropagation) 학습 과정에 의한 학습 진도를 그림 7 에 보이고, 300 번 학습후의 학습 결과를 그림 8-(a),(b)에, 오차(Error) 평면을 그림 9-(a),(b) 에 보였다. 이때 비교대상 신경망은 입력층 1 개, 히든(hidden)층 3 개, 출력층 1개의 5층 신경망으로서, 1번 히든

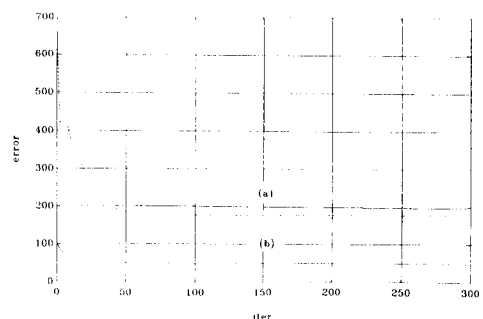
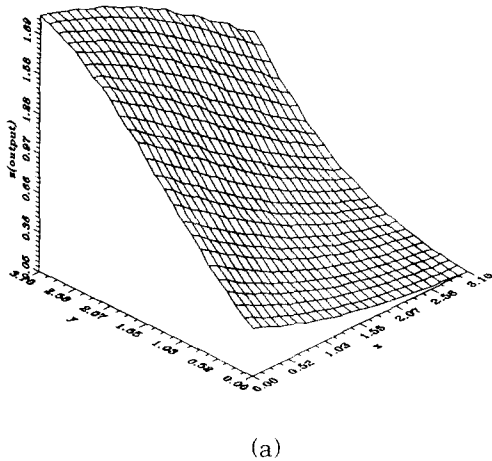


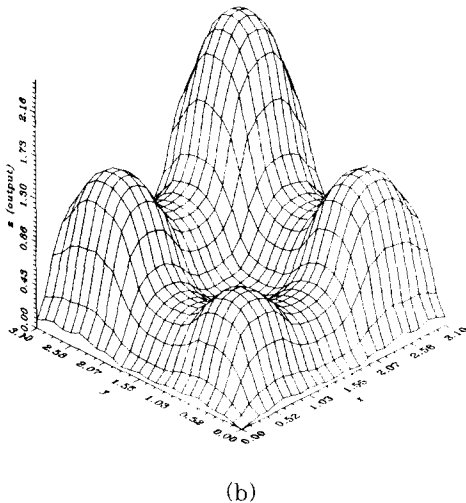
그림 7. 그림 4의 학습 패턴에 대한 학습 과정
(a) 무작위 초기화 방법
(b) 제안된 초기화 방법

Fig. 7. Error history for the training pattern of Fig 4.
(a) random initialization method,
(b) proposed initialization method.

(hidden)층의 활성화함수는 (7)식으로 표현되는 시그모이드(sigmoid) 함수이고, 나머지 층은 선형조합 함수 특성을 가지며, 입력층 유닛 2개, 1번 히든(hidden)층 유닛 25개, 2번 히든(hidden)층 유닛 15개, 3번 히든(hidden)층 유닛 10개, 출력층 유닛 1개로 구성하였고, 시그모이드(sigmoid)함수의 초기 기울기는 10, 바이어스(bias)는 1.0, 각 층간의 연결 강도는 0과 1.0 사이값을 무작위(random)하게 초기화한 후, 동일한 학습 과정을 수행하였다.



(a)



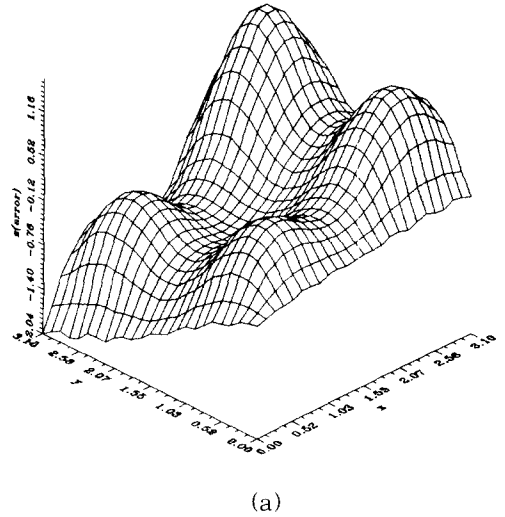
(b)

그림 8. 300번 학습후의 신경망의 출력

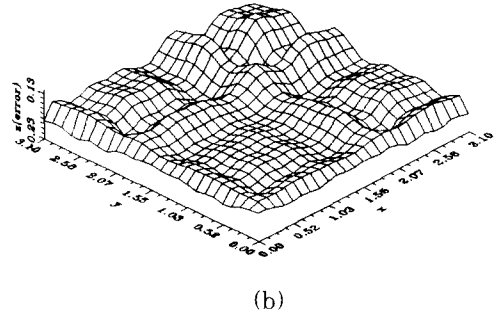
- (a) 무작위 초기화 방법
- (b) 제안된 초기화 방법

Fig. 8. Output of the network after 300 iterations of learning.

- (a) random initialization method.
- (b) proposed initialization method.



(a)



(b)

그림 9. 300번 학습후의 오차 평면

- (a) 무작위 초기화 방법
- (b) 제안된 초기화 방법

Fig. 9. Difference between the output of the network and original training pattern after 300 iterations of learning.

- (a) random initialization method.
- (b) proposed initialization method.

그림 7에서 (a)와 (b)를 비교해 보면, (b)의 경우가 (a)의 경우보다 빨리 학습 진도가 개선됨을 확인할 수 있다. 또한 그림 8에서 (a)와 (b)를 비교해 보면 (a)의 경우 거의 학습이 되지않고 있음을 확인할 수 있고, (b)의 경우 작은 횟수(300번)의 학습과정에 의해 거의 원래 학습 패턴(그림 5)에 근사하고 있음을 볼 수 있다. 한편, 그림 9의 오차 평면을 살펴보면, (a)의 경우 거의 학습이 되지않았기 때문에 오차가 오히려 학습 패턴과 유사하게 나타나고 있으며, (b)의 경우 주로 특징점으로 선정되지 못한 부분에서 학습 오차가 약간 남아 있음을 알 수 있다. 이러한 문제는 궁극적으로 특징점을 추출하는 과정에

있어서 특징점 판별식(학습 패턴의 가속도 성분 등)을 적절히 추가적으로 적용하여 특징점의 갯수를 증가시키고, 또 학습 횟수를 증가시키면 잔류 오차를 더욱 줄일 수 있다고 판단된다. 또한 특징점의 갯수가 많아지면, 시스템 구성이 복잡해지게 되므로, 적정의 특징점들로 신경망을 구성하는 방법이 요구되고, 특히 특징점들이 밀집하게 분포될 경우 적절히 선별하는 등의 방법도 효과적으로 이용될 수 있겠다. 아울러 입력이 여러개인 경우, 제안된 신경망이외의 방법으로도 같은 개념의 근사화를 얻을 수 있겠으며, 최종적으로 얻어진 오차를 더욱 줄이기 위해 부가적인 프로세싱 유닛을 적절히 추가시키는 방법 등도 효과적으로 적용될 수 있겠다.

IV. 검토 및 결론

학습을 위한 입출력쌍($R^m \rightarrow R^n$)이 주어져 있는 경우 입출력 쌍들의 사전 정리와 분포 특징을 이용하여 적절한 신경망의 크기와 관련 계수들의 초기 상태를 결정하는 방법 및 학습이 효과적으로 이루어질 수 있는 구조의 다층신경망을 제안하였다.

본 논문에서는 먼저 주어진 학습용 입출력 쌍으로부터 학습 패턴을 이루는데 중요한 역할을 하는 특징점들을 추출해내고, 이들로부터 다층신경망의 프로세싱 유닛의 비선형성에 관계되는 계수들과 연결강도 등의 값을 학습없이 계산에 의해 구하여 추출된 특징점들을 입력과 출력으로 대응시킨다. 그후 전체 학습 패턴에 대해 학습을 수행하여 주어진 전체 입출력대응을 수행할 수 있도록 하였다. 제안된 초기화 방법은 다층신경망의 크기와 연결강도 등의 관련 계수들을 사전에 결정할 수 있고, 제안된 신경망은 효과적인 초기값 설정에 의해 기존의 무작위 초기화 방법에 의한 다층신경망에 비해 학습 속도가 대단히 빠를 뿐만 아니라, Sartori가 제안한 무학습 신경망에 비해 작은 수의 유닛을 갖는 구조로써 어느 정도의 오차 한계내로 주어진 학습 패턴의 근사화를 수행할 수 있다는 장점이 있다. 결국 off-line으로 신경망의 구조 및 초기화에 역점을 두어 추후 on-line으로 학습을 효율적으로 수행하게 된다.

또한 학습 패턴에 노이즈(Noise)가 존재할 경우에 대해서는 특징점이 많이 추출될 수 있는데, 이러한 경우 filter technique을 사용하여 적당히 Smoothing시키면 본 논문의 방법이 효과적으로 이용될 수 있다. 그리고 학습 패턴의 특징점이 불규칙적으로 분포할 경우에는 재정리 하여 규칙적인 패턴으로 근사화시키는 등의 방법을 사용하는 연구가 진행 중에 있으

며, 최종적으로 얻어진 오차를 더욱 줄이기 위해서 어떤 계수들을 갖는 프로세싱 유닛을 어떤 위치에 추가해야 하는지와 신경망 설계에서의 추가된 계산량과 개선된 학습과정에서의 이득을 종합하여 기존의 방법과 비교하는 등의 추가적인 연구가 필요하다고 판단된다.

參考文獻

- [1] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing*, vol. 1, pp. 318-362, MIT press, 1986.
- [2] R. H. Nielsen, "Theory of the Backpropagation Neural Network," *Proc. Int. Joint Conf. Neural Networks*, vol. 1, pp. 593-606, 1989.
- [3] M. Stinchcombe and H. White, "Universal Approximation Using Feedforward Networks with Nonsigmoid Hidden Layer Activation Functions," *Proc. Int. Joint Conf. Neural Networks*, vol. 1, pp. 613-617, 1989.
- [4] K. Hornik, M. Stinchcombe and H. White, "Multi-layer feedforward networks are universal approximators" *Proc. Neural Networks*, vol. 2, pp. 359-366, 1989.
- [5] M. Stinchcombe and H. White, "Approximating and Learning Unknown Mappings Using Multilayer Feedforward Networks with Bounded Weights," *Proc. Int. Joint Conf. Neural Networks*, vol. 3, pp. 7-16, 1990.
- [6] M. A. Sartori and P. J. Antsaklis, "Implementations of Learning control Systems Using Neural Network," *IEEE Control Systems*, vol. 12, no. 2, pp. 49-57, Apr., 1992.
- [7] S. Lee and R. M. Kil, "A Gaussian Potential Function Network with Hierarchically Self-Organizing Learning," *Neural Networks*, vol. 4, pp. 207-224, 1991.

[8] 이 지홍, “ $R^l \rightarrow R^l$ 대응을 수행하는 다층신경망의 효율적 학습을 위한 초기상태 결정 방법 ” 대한전자공학회지, 3월호, 1993.

著 者 紹 介



金 丙 鎬(正會員)
1966年 2月 18日生. 1989年 2月
금오공과대학교 전자공학과 졸업
(학사). 1989年 1月 ~ 1993年 3
月 금성정밀(주) 근무. 1992年 3
月 ~ 현재 금오공과대학교 산업
대학원 전자공학과 석사과정 재학
중. 주관심분야는 로봇틱스, 신경회로망, 최적제어
등임.

李 志 弘(正會員) 第 30卷 B編 第 6號 參照
현재 금오공과대학교 전자제어공학
과 조교수