신경망의 기하학적 설계와 이의 Point Location Problem에의 적용

조 성배[o], 김 진형

한국과학기술원 전산학과 인공지능연구실

A Geometric Design of Neural Network and Its Application to Point Location Problem

Sung-Bae Cho and Jin H. Kim

Artificial Intelligence Lab., Dept. of Computer Science, KAIST

요 약

최근에 단순한 기능을 하는 처리기들의 대규모 상호연결을 통하여 문제를 해결하고자 하는 신경망이 기존 계산방식의 한계를 극복할 수 있는 대안의 하나로 제안되어, 패턴인식이나 컴퓨터 시각과 같이 그 처리과정을 알고리즘화 하기 어려운 분야에서 크게 각광받고 있다. 하지만 현재까지 개발된 많은 신경망 방식들은 문제를 해결하는 과정에서 주어진 사전지식을 거의 사용하지 못하는 문제점이 있다. 본 논문에서는 기존 신경망 기법의 문제점을 보완하기 위하여 주어진 문제의 기하학적인 구조를 반영하는 신경망의 기하학적 설계방법을 제안하고 이를 계산기하학의 대표적인 문제중의 하나인 Point Location Problem에 적용하여 그 유용성을 입증한다. Point Location Problem을 제안한 기하학적 신경망으로 해결하는 경우에 전처리 시간 $O(n\log n)$, 저장공간 $O(n)$, 그리고 질의응답 시간 $O(1)$의 복잡도를 나타냄을 볼 수 있다.

I. 서 론

기존의 계산이 절차적인 순서에 의한 알고리즘에 근거하여 문제를 해결하는 방법이라면, 신경망(Neural Networks)은 단순한 기능을 하는 처리기들의 대규모 상호연결을 통하여 문제를 해결하고자 하는 새로운 계산모형이다. 최근에 이러한 신경망이 기존 계산방식의 한계를 극복할 수 있는 대안의 하나로 여러가지 응용분야에 적용되고 있다. 신경망의 가장 큰 장점이라고 하면 주어진 문제에 대한 정확한 해결방법을 모르는 경우에도 자체적으로 학습능력을 사용하여 문제의 특성에 맞는 답을 제공해 준다는 것이다. 이러한 특성으로 인하여 패턴인식이나 컴퓨터 시각과 같이 그 처리과정을 알고리즘화 하기 어려운 분야에서 크게 각광받고 있다.

하지만 신경망의 최대장점인 학습능력은 오히려 가장 큰 단점이 되기도 한다. 왜냐하면 이를 사용하면 답은 얻을 수 있을지 몰라도 어떻게 하여 그와같은 결과가 나왔는지에 대하여 알 수 없는 경우가 많고, 또 어떤 경우에는 문제에 대한 사전지식(a priori knowledge), 예를 들면 문제의 기하학적인 구조등이 명백한 경우에도 이를 신경망의 문제해결 과정에서 이용할 수 없기 때문이다. 따라서 신경망 자체에 대한 좀 더 정형화된 해석방법이 절실한 실정이며 이와 같은 해석이 신경망 기법내에서 사용되어질 수 있어야 할 것이다.

본 논문에서는 이를 위하여 신경망의 작동원리를 계산 기하학적인 입장에서 분석하여 신경망내의 정보가 어떻게 처리되는가를 알아보고 이를 이용하여 단순한 학습방법이 아닌 문제의 패턴공간(pattern space)을 기하학적으로 고려하는 새로운 신경망 기법을 제안한다. 또 제안된 방법이 일반적인 문제, 특히 계산 기하학 [2, 5, 8]에서 다루는 많은 문제에 어떻게 사용되어질 수 있는가를 알아보기 위해서 Point Location Problem에 적용하여 본다. 그리고 나서 신경망 방법을 사용하였을때 필요한 전처리 시간(preprocessing time), 질의응답 시간(query time) 및 저장공간의 복잡도(storage complexity)를 분석하여 기존의 방법들과 비교한다.

II. 기존 신경망 방식의 학습

신경망의 기본 아이디어는 데이타로 주어진 입력과 출력의 쌍에 존재하는 사상(mapping)을 자체의 적응성(adaptability)에 의하여 생성해 내는데 있다. 이때, 신경망이 제대로 동작하기 위하여 연결강도(connection weight)를 반복법에 의해 조정해 나가는 것을 학습이라고 하는데, 패러미터를 추정하는 전통적인 방법으로는 실제 출력과 원하는 출력 사이의 차이를 최소화 시키는 것으로 여러가지 유용한 최적화 기법이 사용될 수 있으며, 그 중에서 가장 널리 사용되는 것으로는 경사정보(gradient information)를 이용하는 Backpropagation을 들 수 있다.

Backpropagation은 지도 학습(supervised learning) 알고리즘으로서 이상적인 출력과 실제 출력사이의 평균 제곱 오차에 해당하는 비용함수의 값을 최소화하기 위하여 경사추적 방법을 사용한다 [6, 7, 11]. 이것은 결국 신경망의 전체 오차를 최소화하도록 하는 연결강도를 구하기 위해 다음과 같은 반복식을 적용하는 것이다.

$$w_{n+1} = w_n + \alpha(\partial E/\partial w_n)$$

이때 α는 학습률이라 한다.

III. 기하학적 신경망의 관련연구

II장에서 기술한 바와 같이 신경망의 문제해결 방식은 간단한 수식을 반복적으로 적용하기만 하면되는 일종의 경사추적 기법이기 때문에 그 해결방법이 명확하지 않은 문제를 처리하는데 새로운 가능성을 제시하고 있다. 하지만 이러한 학습에 의존하는 방법은 문제의 복잡도에 따라서 그 학습시간이 급격히 증가하는 문제가 있으며, 최근 연구결과[1]에 의하면 신경망이 올바로 작동하도록 연결강도의 값을 결정하는 것은 *NP-complete* 문제라는 것이 증명되었다.

이제 신경망의 한 노드가 하는 기능을 살펴보자. 신경망의 한 노드는 그 입력으로 들어오는 값에 적절한 가중치를 곱해서 모두 더한후 이것이 자신의 임계치보다 크면 +1을, 작으면 -1을 출력하는 임계함수 기능을 한다고 볼 수 있다. 즉,

$$y = f(\Sigma^{N-2} w_i x_i - \theta)$$

단, $f(x) = +1,\ \text{if } x > 0$

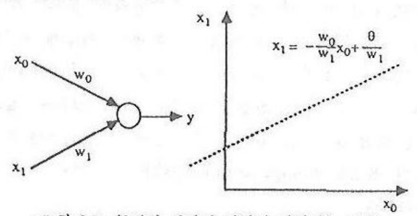
$f(x) = -1,\ \text{if } x \le 0.$

이때, 두개의 입력을 받아서 출력하는 노드의 경우에는 위의 식을 x_1에 대하여 정리하여 다음과 같은 식을 유도할 수 있다.

$$w_0 x_0 + w_1 x_1 - \theta = 0$$

$$x_1 = -\frac{w_0}{w_1} x_0 + \frac{\theta}{w_1}$$

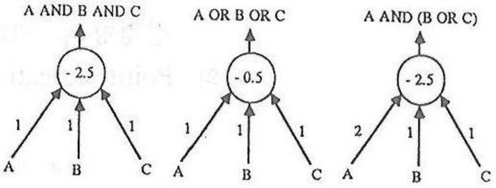
이 식은 <그림 3.1>과 같이 이차원 공간을 이분하는 직선의 식임을 쉽게 알 수 있다.



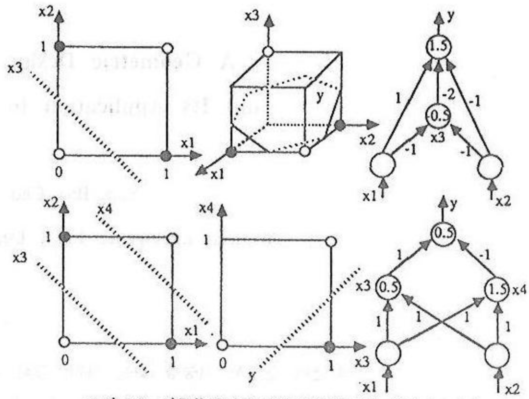
<그림 3.1> 두개의 입력에 의하여 결정되는 직선

기존 신경망기법의 단점을 보완하기 위하여 신경망의 설계나 학습단계에 기하학적인 개념을 도입하고자 하는 연구가 많이 진행되고 있는데 [3, 9, 10, 13, 14], 이것들은 기본적으로 신경망의 한 노드가 하는 기능이 N차원 공간상의 초평면을 나타낸다는 것에 착안한 것으로 대표적인 것들을 연대순으로 소개하면 다음과 같다.

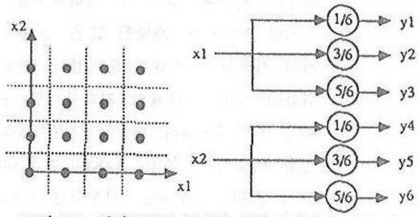
첫째로 Wieland등 [14]은 초평면 분할(Hyperspace Partitioning) 기법을 도입하여 신경망의 능력이나 구조를 기하학적으로 분석하고자 시도하였는데, 그 결과로 초평면 분할을 하는 노드가 임의의 부울 논리식(Boolean logic expression)을 구현할



<그림 3.2> 신경망 노드에 의한 논리함수의 생성



<그림 3.3> 정규분할법에 의한 XOR문제의 해결에 2가지



<그림 3.4> 패턴공간의 분석을 통한 신경망의 설계 예

수 있음을 보였다. <그림 3.2>는 연결강도의 값과 임계치를 적절히 조정하여 AND, OR, 그리고 이들이 결합된 형태의 부울식을 구현할 수 있음을 보여준다. 이 연구는 또한 신경망의 일반화 능력이 수학적으로는 결국 임의의 구간값을 보간(Interpolation)하는 것이라는 것을 밝혔다. 이와 같은 사실들은 매우 기초적인 것이기는 하지만 신경망의 능력을 기하학적으로 해석하고자 했던 첫 시도라는 점에 의의가 있다.

다음으로 Rujan등 [10]은 정규 분할(Regular Partitioning) 방법을 고안하였는데, 이것은 한 층의 노드가 가능한 최대로 입력패턴을 나눌 수 있도록 하는 병렬 분할면(*Parallel Cutting Plane*)을 만든후 다음 층의 노드로 좀 더 세분하여 나누는 일종의 Greedy 알고리즘이다. 이것은 최대 축소원리(Maximal Contraction Principle)를 사용한 것이라 할 수 있는데, <그림 3.3>은 XOR문제에 대한 작동 예를 보여주고 있다.

마지막으로 Ramacher등 [9]은 문제에 따른 패턴공간을 기하학적으로 분석하여 신경망의 구조를 결정하는 방법을 제안하였다. 이것은 주어진 문제의 패턴들을 나눌 수 있는 최소의 분리평면을 구한후 이것이 좌표축과 교차하는 점을 이용하여 각 노드의 연결강도와 임계치를 결정하는 방법으로 <그림 3.4>는 16개의 점을 6개의 직선으로 나누고 이로부터 6개 노드의 임계값을 결정하는 예를 보여준다. 이 연구에서는 여기에서 중복된 정보를 제거하여 최적의 구조를 얻어내는 방법에 대해서도 언급하고 있다.

IV. 기하학적 신경망

이 장에서는 신경망을 기하학적으로 설계하고 학습시키는 새로운 방법에 대하여 기술한다. 이것은 풀고자 하는 문제에 대한 사전지식이 기하학적으로 분석가능할 경우, 그 정보를 이용하여 신경망을 단계적으로 구성하고자 하는 것이 기본 아이디어이다. 이를 위하여 다음과 같이 기하학적 신경망을 구성하는 노드들을 정의하기로 하자.

정의 1: (평면노드)
평면노드는 N개의 입력 x_i를 $f(\Sigma\ w_i x_i\ -\ \theta)$로 계산하여, 이 값이 0보다 크면 +1을 0보다 작으면 -1을 출력하는 노드이다. 이 때 이 노드의 기능은 N차원 공간을 두 영역으로 나누는 N차원의 초평면을 결정한다.

정의 2: (볼록노드)
볼록노드는 평면노드의 출력을 입력으로 받는 노드로 입력 값이 모두 1일때만 1을 출력하는 노드이며 입력평면들이 이루는 볼록영역을 결정한다.

정의 3: (오목노드)
오목노드는 볼록노드의 출력을 입력으로 받는 노드로 입력 값 중에서 하나라도 1인 것이 있으면 1을 출력하는 노드로 입력된 볼록영역들로 구성가능한 임의의 오목영역을 결정한다.

이상의 정의들로 부터 이 세가지 노드들을 적절히 조합시키면 임의의 결정경계선(decision boundary)을 나타낼 수 있음을 쉽게 알 수 있다. 이것을 정리하면 다음의 정리를 유도할 수 있다.

정리 1: 평면노드와 볼록노드, 그리고 오목노드의 층으로 이루어진 3층의 Feedforward 신경망은 모든 결정문제를 해결하는데 필요충분하다.

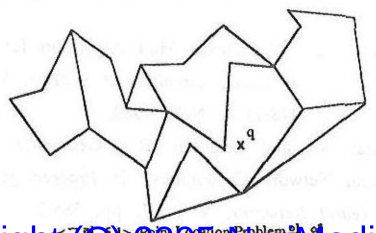
V. Point Location Problem에의 적용

Point Location Problem [2, 4, 5, 8, 12]은 n개의 vertex로 이루어진 planar straight-line graph(PSLG) G와 하나의 질의점(query point) q가 주어져 있을때, G에 의하여 생성된 planar subdivision의 어떤 영역에 q가 포함되는지를 결정하는 문제이다. <그림 5.1>은 한 예를 보여준다.

5.1 노드 패러미터의 결정

이 절에서는 Point Location Problem을 해결하는 기하학적 신경망을 설계하기 위해서 필요한 각 노드의 패러미터 값들을 결정해 보자. 여기에서 패러미터는 결국 각 노드에 입력되는 연결선의 연결강도와 각 노드의 임계치이다.

먼저 평면노드의 패러미터들을 결정하여 보자. 주어진 PSLG의



<그림 5.1> Point Location Problem의 예

각 edge에 대해서 하나의 평면노드가 할당된다. <그림 5.2>와 같이 두개의 점, (x_1, y_1)과 (x_2, y_2)를 지나는 직선의 식은 다음과 같이 정의된다.

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1) + y_1 \quad \text{----- (식 5.1)}$$

$$y = \frac{y_2 - y_1}{x_2 - x_1}x + \frac{x_2 y_1 - x_1 y_2}{x_2 - x_1} \quad \text{------- (식 5.2)}$$

그런데 <그림 5.3>과 같은 평면노드에서의 기능은 다음과 같다.

$$w_0 x + w_1 y - \theta = 0 \quad \text{------------------------ (식 5.3)}$$

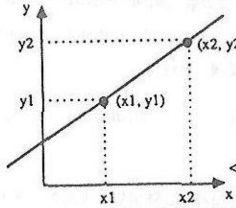
$$y = -\frac{w_0}{w_1}x + \frac{\theta}{w_1} \quad \text{-------------------- (식 5.4)}$$

(식 5.2)와 (식 5.4)를 연립하여 풀면 각 패러미터 w_0와 w_1 및 θ는 다음과 같이 구해질 수 있다.

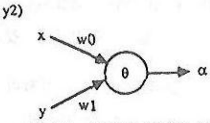
$$w_0 = y_1 - y_2$$
$$w_1 = x_2 - x_1$$
$$\theta = x_2 y_1 - x_1 y_2$$

이때, 두점의 x좌표가 일치하는 경우, 즉 입력된 선분이 수직일 때는 이러한 방법을 그대로 사용할 수 없는 문제가 발생한다. 이 경우에는 두점의 x좌표를 약간 변경시키는 방법을 사용하여 해결하면 된다.

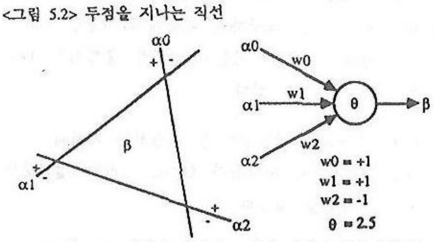
다음으로 볼록노드의 패러미터들을 구해보자. 하나의 볼록영역(convex region)을 결정하기 위하여 평면노드들의 출력을 입력으로 받아서 이루어지는 볼록영역에 대하여 하나의 볼록노드를 할당한다. 각 입력의 연결강도는 그것이 결정하는 직선의 어느 반공간에 의하여 볼록영역이 이루어 지는가에 따라서 결정된다. 즉, 볼록영역이 해당직선의 상위 반공간(upper half-space)에 속하는 경우의 연결강도는 +1로 설정하고, 하위 반공간(lower half-space)에 속하는 경우에는 -1로 만든다. 그리고 임계치 θ는 모든 입력이 1일때 1을 출력해야 하기 때문에 입력노드의 수에서 0.5를 뺀 값을 사용한다. <그림 5.4>는 한 예로 세개의 직선



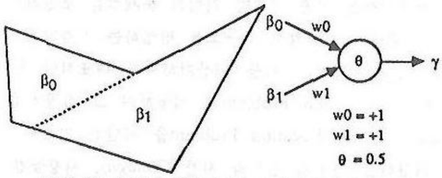
<그림 5.2> 두점을 지나는 직선



<그림 5.3> 두개의 입력을 갖는 신경망의 노드



<그림 5.4> 세개의 직선이 모여서 하나의 볼록영역을 이루는 경우의 예



<그림 5.5> 두개의 볼록영역을 모아서 하나의 영역을 이루는 경우의 예

이 모여서 하나의 볼록영역을 이루는 경우에 각각의 패러미터 값을 어떻게 설정하여야 하는지를 보여준다

마지막으로 오목노드의 패러미터 값을 결정해 보자 오목노드는 볼록노드의 출력을 입력으로 받아서 그 중에 하나라도 1인 경우에 1을 출력하는 노드인데, 이것은 주어진 PSLG에서 하나의 subdivision에 해당하게 된다 패러미터 값으로는 각각의 연결강도를 모두 +1로 설정하고 임계치는 0 5로 설정하면 된다 <그림 5 5>는 두개의 볼록영역을 모아서 하나의 영역을 나타내는 경우를 예로 보여주고 있다

5 2 알고리즘

앞 절에서 정의한 노드들의 패러미터를 가지고 Point Location Problem을 푸는 알고리즘을 기술하면 다음과 같다

알고리즘 1 Point Location Problem을 위한 기하학적 신경망
- 입력 Planar Straight-Line Graph G, 질의점 q
- 출력 q가 포함되는 subdivision의 인덱스

1 기하학적 신경망의 구성
- a 각 선분에 대하여 하나의 평면노드를 결정
- b a 결과 생성된 볼록영역에 대해 하나의 볼록노드를 결정
- c 각 영역을 이루는 볼록노드를 모아서 오목노드를 결정

2 질의점 q에 대해서 각 층의 노드들을 순차적으로 수행

이 알고리즘은 주어진 PSLG에 대해서 기하학적 신경망을 구축하고 질의점 q가 입력되었을때 어떤 오목노드의 출력이 1로 되는지를 조사함으로써 이 점이 어떤 subdivision에 속하는지를 알 수 있게 한다

이 알고리즘의 복잡도를 분석해 보면 먼저 기하학적 신경망을 구축하는데 필요한 저장공간은 평균적인 경우에 입력된 선분의 수에 비례하기 때문에 $O(n)$이 됨을 쉽게 알 수 있다 또 전처리 과정에서 주어진 임의의 영역을 PSLG의 각 선분으로 분할하는 과정을 decomposition 알고리즘 [4]에 의하여 자동화하는 경우에 triangulation에 소요되는 $O(n\log n)$이 필요하기 때문에 다음의 소정리와 같은 복잡도를 갖게 된다.

소정리 1 알고리즘 1의 전처리 시간은 $O(n\log n)$이고, 필요한 저장장소의 복잡도는 $O(n)$이다

이 신경망에 임의의 질의점 q가 입력되면 세개의 층에 대해서 일정한 계산시간을 요구하기 때문에 질의응답 시간의 복잡도는 $O(1)$이 된다 따라서, 이상의 결과를 종합하면 다음과 같은 정리를 이끌어 낼 수 있다

정리 2 Point Location Problem을 해결하는 기하학적 신경망은 전처리 시간 $O(n\log n)$, 저장공간 $O(n)$, 그리고 질의응답 시간 $O(1)$의 계산 복잡도를 갖는다

VI 결 론

본 논문에서는 기존 신경망 기법의 문제점을 보완하기 위하여 주어진 문제의 기하학적인 구조를 반영하는 신경망의 기하학적 설계방법을 제안하고 이를 계산기하학의 대표적인 문제중의 하나인 Point Location Problem에 적용하여 그 유용성을 입증하여 보았다 Point Location Problem을 제안한 기하학적 신경망으로 해결하는 경우에 전처리 시간 $O(n\log n)$, 저장공간 $O(n)$, 그리고 질의응답 시간 $O(1)$의 복잡도를 나타내었다 이로 부터 문제의 구조가 기하학적으로 분석가능한 경우에 제안된 기하학적 신경망 방식이 효과적임을 알 수 있다

참고 문헌

[1] 조 성배, 김 진형, "인공 신경망의 계산 복잡도," 한국정보과학회 가을 학술발표 논문집, pp. 315-318, Vol. 16, No 2, 1989

[2] Aggarwal, A , *Computational Geometry*, lecture notes, MIT, 1988

[3] Brady, M , Raghavan, R , and Slawny, J , "Gradient Descent Fails to Separate," in *Proceedings of Int Conf Neural Networks*, Vol I, pp 649-656, 1988

[4] Edahiro, M , Kokubo, I , and Asano, T., "A New Point-Location Algorithm and Its Practical Efficiency - Comparison with Existing Algorithms," *Research Memorandum RMI 83-04*, Department of Mathematical Engineering and Instrumentation Physics, Univ. of Tokyo, 1983

[5] Lee, D I , and Preparata, F P , "Computational Geometry - A Survey," *IEEE Trans Computers*, Vol C-33, No 12, pp 1072-1101, Dec 1984

[6] Lippmann, R P , "An Introduction to Computing with Neural Nets," *IEEE ASSP Magazine*, pp 4-22, April 1987

[7] Minsky, M , and Papert, S , *Perceptrons - An Introduction to Computational Geometry*, MIT Press, Cambridge, MA, 1987

[8] Preparata, F P , and Shamos, M , *Computational Geometry An Introduction*, Springer-Verlag, New York, 1985

[9] Ramacher, U , and Wesseling, M , "A Geometrical Approach to Neural Network Design," in *Proceedings of IEEE Int Joint Conf Neural Networks*, Vol II, pp 147-153, 1989

[10] Rujan, P , "A Geometric Approach to Learning in Neural Networks," in *Proceedings of IEEE Int Joint Conf Neural Networks*, Vol II, pp 105-109, 1989

[11] Rummelhart, D E , and McClelland, J L (Eds), *Parallel Distributed Processing Explorations in the Microstructure of Cognition*, MIT Press, Cambridge, MA, 1986

[12] Toussaint, G T , "Computational Geometry Recent Developments," in *New Advances in Computer Graphics*, Earnshaw, R A , and Wyvill, B (Eds), Springer-Verlag, 1989

[13] Wennmyr, E , "A Convex Hull Algorithm for Neural Networks," *IEEE Trans Circuits and Systems*, Vol 36, No 11, pp 1478-1484, Nov 1989

[14] Wieland, A , and Leighton, R , "Geometric Analysis of Neural Network Capabilities," in *Proceedings of Int Conf Neural Networks*, Vol III, pp 385-392, 1987