韓國 OA學會 論文誌
第3卷 第1號, 1998. 3.

論文 98-3-1-2-2

대표 패턴을 사용한 가변 기울기 역전도 알고리즘의 점진적 학습방법

심범식*, 윤충화**

The Incremental Learning Method of Variable Slope Backpropagation Algorithm Using Representative Pattern

Beom-Sik Shim*, Chung-Hwa Yoon**

요 약

역전도 알고리즘은 연관 기억장치, 음성 인식, 패턴인식, 로보틱스등 여러 응용 분야에 다양하게 사용되고 있다. 그러나 새로운 학습 패턴을 추가적으로 학습시키려면 이전에 학습했던 모든 패턴과 추가되는 패턴을 갖고 처음부터 새로운 학습을 수행하여야 한다. 이는 패턴의 개수가 점차 늘어날수록 학습에 소요되는 시간이 기하 급수적으로 길어지는 결과를 초래하게 된다. 따라서 주기적으로 다량의 데이터를 추가로 학습을 할 경우에 이러한 점진적 학습은 반드시 해결해야 할 문제점으로 간주된다. 본 논문에서는 기존의 신경망 구조는 그대로 유지하면서 대표 패턴을 추출해 추가 학습을 수행하는 방법을 제안하고 제안된 기법의 효율성을 위해 기계 학습 분야의 벤치마크로 많이 사용되는 Monk's data와 Iris data에 적용해 보았다.

Abstract

The Error Backpropagation algorithm is widely used in various areas such as associative memory, speech recognition, pattern recognition, robotics and so on. However, if and when a new learning pattern has to be added in order to drill, it will have to accomplish a new learning with all previous learning pattern and added pattern from the very beginning. Somehow, it brings about a result which is that the more it increases the number of pattern, the longer it geometrically progress the time required by learning. Therefore, a so-called Incremental Learning Method has to be solved the point at issue all by means in case of situation which is periodically and additionally learned by numerous data. In this study, not only the existing neural network construction is still remained, but it also suggests a method which means executing through added learning by a model pattern. Eventually, for a efficiency of suggested technique, both Monk's data and Iris data are applied to make use of benchmark on machine learning field.

---

* 마산전문대학 사무자동화과 조교수
** 명지대 컴퓨터 공학과 부교수

논문접수 : 98.2.23. 심사완료 : 98.3.20.

I. 서 론

신경회로망(neural network)의 이론은 폰 노이만(Von Neumann)형 컴퓨터와 같이 1940년대에 발표되었으나 민스키(Minskey)와 페퍼드(Papert)가 퍼셉트론의 한계를 지적함으로써 침체되었다가 홉필드(Hopfield)의 연상 메모리 모델의 제시로 신경회로망에 대한 연구가 활기를 띠기 시작했다. 또한 다층 신경회로망은 퍼셉트론의 한계를 극복하기 위하여 제안되었고 특히 PDP(Parallel Distributed Processing)그룹에 의해 폭넓은 연구가 진행되었다[1].

신경회로망의 구조나 적용 분야에 따라 여러 가지 학습 알고리즘이 연구되었으나 일반적으로 역전도 알고리즘을 이용한 다층 퍼셉트론 모델이 가장 널리 사용되고 있다[2]. 역전도 알고리즘은 연관 기억 장치(Associative Memory), 음성 인식, 패턴 인식, 로보틱스등 여러 응용 분야에 다양하게 사용되고 있다. Minskey와 Pepert에 의해 지적되었던 비선형 분리 문제(Non-linearly Separable Problem)를 은닉 뉴런층을 이용하여 해결할 수 있다는 장점외에도, 알고리즘이 간결하고 안정성(Satablity)이 수학적으로 증명되어 있다는 점을 들 수 있다. 그럼에도 불구하고 계속 많은 논문들이 역전도 알고리즘에 대해 발표되는 주된 이유는 입력층의 뉴런 개수와 저장하려는 패턴의 개수가 증가할수록 학습 시간이 많이 소요된다는 사실 때문이다.

기존의 신경회로망 모델은 새로운 학습 패턴을 추가적으로 학습시키려면 이전에 학습했던 모든 패턴과 추가되는 패턴을 갖고 처음부터 새로운 학습을 수행하여야 한다. 이는 패턴의 개수가 점차 늘어날수록 학습에 소요되는 시간이 기하 급수적으로 길어지는 결과를 초래하게 된다. 따라서 주기적으로 다량의 데이터를 추가로 학습을 할 경우에 이러한 점진적 학습은 반드시 해결해야 할 문제점으로 간주된다.

신경회로망의 분야에서는 점진적 학습기능에 대한 연구가 발표된바 있으며, 대부분 초기에 사용하였던 신경망의 구조를 변경하는 방법을 사용한다. 추가 패턴을 기존의 출력 클래스로 학습시키는 시도도 있으며, 또한 새로운 출력 클래스를 추가해서 학습시키는 시도도 있다. 그러나 본 논문에서는 기존의 신경망 구조는 그대로 유지하면서 추가 학습을 수행하는 방법을 추구하였다.

즉 초기에는 학습해야 할 모든 패턴들을 갖고 학습을 수행하고 학습 패턴을 클러스터링하여 대표 패턴을 추출한다. 이러한 학습 작업과 클러스터링 작업은 동시에 수행된다. 후에 추가적으로 학습해야 할 패턴이 생기면 이전에 추출하였던 대표 패턴과 추가 패턴만을 합하여 학습함으로써 학습에 소요되는 시간을 단축할 수 있게 된다. 물론 이때 기존에 학습되었던 초기 학습 패턴중

일부분은 없어지게 되는 현상이 발생할 것이며, 이를 최소화하는 파라미터(예를 들어 클러스터와 대표패턴의 개수등)의 선정이 중요한 관건이 될 것이다.

Ⅱ. 역전도 알고리즘

다층 신경회로망의 구조는 입력층과 출력층 그리고 그 사이에 하나 이상의 중간층(은닉층)으로 구성된다. 각 층은 뉴런으로 구성되고 각 층에 있는 뉴런은 인접층의 뉴런들과 연결되며 각 연결은 가중치를 갖는다.

가중치에 대한 오류의 최급강하법(gradient descent method)을 사용하여 반복적 계산으로 오류를 최소화하면서 학습하는 델타 규칙을 다층 신경망에 일반화시킨 오류 역전도 알고리즘은 각 노드에 제시된 패턴에 대하여 활성화 함수(Activation Function)를 이용하여 출력 (O_j)을 구하고, 요구되는 출력 (T_j)과 실제 출력 (O_j)과의 차이를 계산하여 이 차이를 출력층에서 은닉층을 거쳐 입력층으로 역전도 시키면서 층과 층 사이의 가중치(weight)를 갱신한다[3]. 그림 1은 다층 구조로 된 신경회로망을 나타낸다.

역전도 알고리즘에 쓰이는 활성화 함수는 다음과 같은 시그모이드(sigmoid)함수를 사용한다.

$$f_j(net_{pj}) = \frac{1}{1+e^{-net_{pj}}} \tag{1}$$

$$net_{pj} = \sum_i W_{ji} O_{pj} + \theta_j \tag{2}$$

$$O_{pj} = f_j(net_{pj}) \tag{3}$$

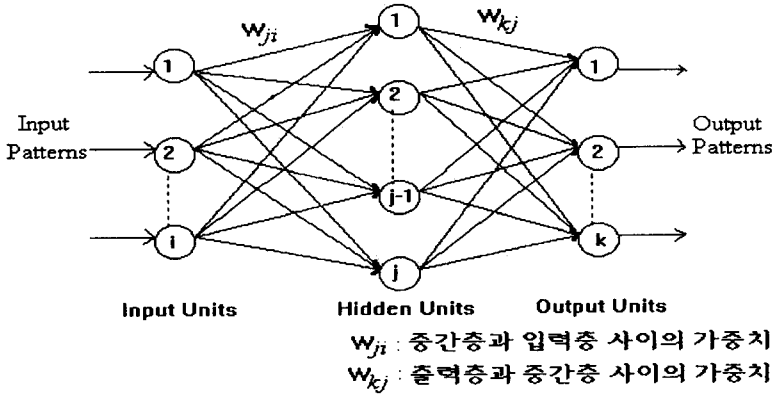


그림 1 다층 신경회로망의 구조
Fig. 1 Structure of multilayer nueral network

여기서 W_{ji}는 이전층의 i번째 뉴런과 다음층의 j번째 뉴런간의 연결 가중치(weight)를 나타낸다. O_{pj}는 p번째 패턴의 j번째 뉴런의 출력이고, θ_j는 임계치(threshold)이다. 출력층에서 계산되는 실제 출력값과 요구되는 출력값 사이의 오류는 다음과 같이 구할 수 있다.

$$E_p = 0.5 * \sum_j (T_{pj} - O_{pj})^2 \quad (4)$$

$$E = \sum E_p \quad (5)$$

식 4는 학습되는 출력층에서 발생하는 p번째 패턴의 오류값을 나타내고 식 5는 각 패턴의 오류를 합산한 전체 오류이다. T_{pj}는 출력층에서 p번째 패턴의 j번째 뉴런의 목표값이다. 계산된 전체 오류가 정의된 오류 예를 들어 10^{-4}이나 10^{-5}보다 작을 때까지 학습을 반복하게 된다.

출력층과 은닉층 사이의 가중치 변화량 δ_{pj}는 다음과 같다.

$$\delta_{pj} = (T_{pj} - O_{pj}) \cdot O_{pj} \cdot (1 - O_{pj}) \quad (6)$$

은닉층과 입력층 사이의 가중치 변화량 δ_{pj}는 다음과 같다.

$$\delta_{pj} = O_{pj} \cdot (1 - O_{pj}) \cdot \sum_k \delta_{pk} \cdot W_{kj} \quad (7)$$

식 6과 식 7을 이용하여 수정되는 가중치의 변화량은 다음 식과 같다.

$$\Delta_p W_{ji}(n+1) = \eta \cdot (\delta_{pi} \cdot O_{pi}) + \alpha \cdot \Delta_p W_{ji}(n) \quad (8)$$

여기에서 η는 학습율(learning rate)을 나타내고, α는 모멘텀율(momentum rate)을 나타낸다.

기본적으로 역전도 알고리즘은 기울기 감소(gradient descent) 기법을 이용하여 식 5의 전체 오류를 최소화하는 학습 기법이다. 이때, 가중치는 식 8에 의하여 변화되며, 아주 작은 양만큼씩 변화하면서 오류평면의 전역최소치로 수렴해 가게 된다. 이 변화량을 제어하는 매개변수의 역할을 학습율이 담당한다. 실제로 학습율은 클수록 빠른 학습이 가능하나, 너무 크게 되면 전역 최소치를 사이에 두고 진동(oscillation)이 일어나는 경우를 관찰할 수 있다. 모멘텀율은 학습속도를 향상시키기 위하여 사용되는데, 이전에 가중치가 변화하였던 양(history of weight change)을 금번 가중치 수정시, 얼마만큼 고려할 것인가를 결정하는 매개변수다. 보편적으로 학습율과 모멘텀율은 0과 1사이의 실수를 사용한다.

전체적인 학습 과정을 살펴보면 다음과 같다.

단계 1 : 초기 가중치를 임의로 설정한다.

단계 2 : 입력 패턴에 대해서 입력층에서 출력층까지 식 1, 2, 3에 의 해 각 뉴런의·출력값을 구한다.

단계 3 : 식 4, 5로 전체 오류를 구하여 정의된 오류보다 작으면 학습을 마치게 된다.

단계 4 : 식 6, 7을 이용하여 층과 층 사이의 가중치 변화량을 계산하고, 식 8을 적용하여 새로운 가중치를 구한다.

단계 5 : 단계 2로 가서 반복 수행한다.

Ⅲ. 가변 기울기 역전도 알고리즘

앞에서 설명한 바와 같이 역전도 알고리즘의 단점 중의 하나는 학습이 종료되는 데 많은 시간이 걸린다는 것이다. 이와 같은 문제점을 해결하기 위하여 시그모이드(sigmoid) 함수의 기울기를 발생하는 전체 오류에 따라 적절하게 변화시킴으로써 학습 수행 시간을 최소화한다.

역전도 알고리즘에서, 학습이 진행됨에 따라 출력층의 각 뉴런 출력값이 0 또는 1에 점점 수렴해 가기 때문에 시그모이드 함수의 기울기를 점점 크게 한다면 출력층에서

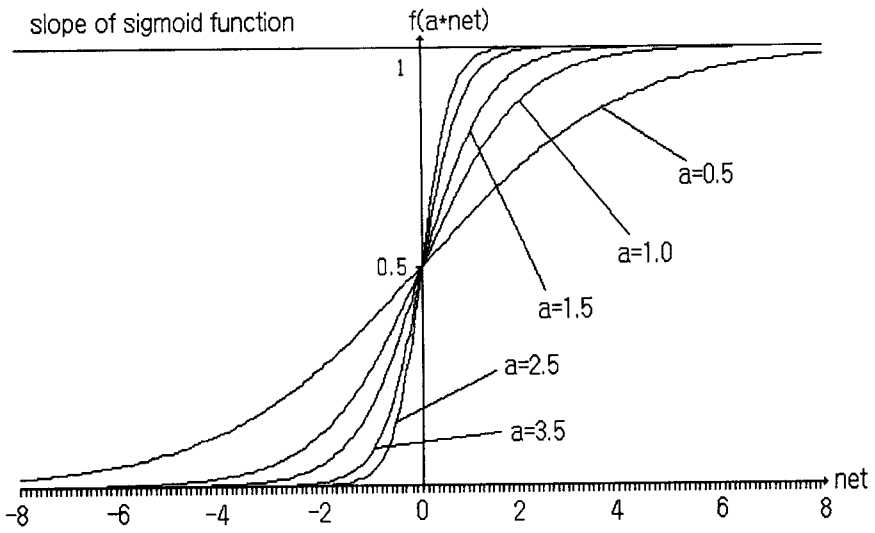


그림 2 시그모이드 함수의 기울기
Fig. 2 Slope of sigmoid function

각 뉴런의 출력이 요구되는 출력(0 또는 1)에 더욱 가깝게 되어 수렴하는 속도가 빨라질 것이다.

$$\lambda = -\log_{10}(E), \quad E \leq 0.1\text{일때} \quad (9)$$

$$\lambda = -\log_{10}(E)+0.5, \quad 0.3 < E < 1.0\text{일때} \quad (10)$$

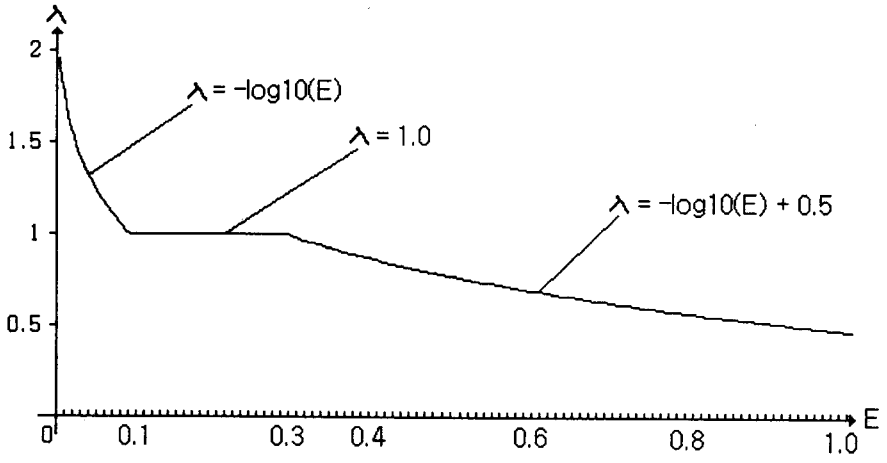


그림 3 전체 오류에 따른 시그모이드 함수의 기울기 변화
Fig. 3 Transformation of slope of sigmoid function for error

그러나, 기울기를 단순히 크게 한다고 해서 학습이 빨리 끝나는 것이 아니다. 왜냐하면 가중치의 계속적인 변화, 패턴들 간의 관계, 각 층의 뉴런 개수 등이 복합적으로 관계하고 있기 때문이다. 따라서, 학습이 빨리 종료되면서 다른 알고리즘보다 효율이 떨어지지 않도록 하기 위하여 다음의 식 9와 10을 이용한다. 이는 학습이 수행되는 과정에서 시그모이드 함수의 기울기를 전체 오류에 따라 변화시킴으로써 학습 시간의 단축을 꾀하였다.

식 10에서 0.5를 더하는 이유는 오류가 0.3에서 1.0 사이에 있을 때 -log10(E)의 범위가 0.507에서 0.0046이 되기 때문에 시그모이드 함수의 출력값 변화 폭이 아주 미세하게 되어 학습 효율을 저하시킨다. 따라서 적당한 기울기를 정하기 위하여 0.5를 더하게 되었다.

그림 3은 전체 오류값과 λ 사이의 관계를 그래프로 표현한 것이다.

식 9, 10에 의거하여 시그모이드 함수에

λ를 추가하면 다음의 식과 같이 된다.

$$f_j(net_{pj}) = \frac{1}{1+e^{-\lambda \cdot net_{pj}}} \quad (11)$$

λ의 추가로 역전도 알고리즘에서 바뀌는 수식은 다음과 같다. 식 6의 수식은 식 6′로 수정되며,

$$\delta_{pj} = (T_{pj} - O_{pj}) \cdot \lambda \cdot O_{pj} \cdot (1 - O_{pj}) \quad (6')$$

식 7은 식 7′로 바뀌게 된다.

$$\delta_{pj} = \lambda \cdot O_{pj} \cdot (1 - o_{pj}) \cdot \sum_k \delta_{pk} \cdot W_{kj} \quad (7')$$

학습이 진행됨에 따라 시그모이드 함수의 기울기 (λ)는 점점 커진다.

결국, f′가 0에 가까워지더라도 λ값이 식 6′와 식 7′처럼 곱해지기 때문에 층과 층 사이의 가중치 변화량은 커지게 된다.

Ⅳ. 점진적 학습

4.1. 대표 패턴의 추출 기법

패턴 인식 문제에 있어, 여러 개의 특징을 갖는 패턴들을 2차원 평면으로 사상(mapping) 하여 자료의 분포를 파악하려는 기법은 이미 오래 전부터 연구되어 온 분야이다. 이는 기본적으로 n차원 공간의 점들을 2차원 공간으로 사상하는 문제이다. 즉 패턴들을 측정 공간(measurement space)으로부터 특성 공간(property space)으로 변환하려는 시도로써, 일반적으로 선형 사상 기법과 비 선형 사상 기법이 있다. 대표적인 선형 기법으로는 Variable by Variable Plotting, Rotation and Projection과 Karhunen-Loeve Transformation (Eigenvector Projection)[4] 등이 있으며, 비선형 기법에는 E. A. Patrick의 dovetail mapping과 column mapping[5], Fukunaga & Olsen 기법[6], NLM (NonLinear Mapping)[7] 등이 있다.

다차원 패턴의 사상에는 선형 기법보다는 비 선형 기법이 효율적인 것으로 나타나 있으며, 본 연구에서는 이중에서 가장 효율적인 Sammon[7]의 알고리즘을 이용하여 프로그램을 구현하였다. 기본적인 개념은 다차원 공간의 점으로 표현되는 패턴들의 거리를 보존하는 사상 기법을 사용한다는 것이다.

초기에는 선형 기법인 Eigenvector Projection의 결과를 이용하여 사상한 후, 각 점간의 거리, d_{ij}를 다음 식 12을 사용하여 계산한다.

$$d_{ij} = \left[\sum_{k=1}^{n} (X_{ik} - X_{jk})^2 \right]^{1/2} \quad (12)$$

여기에서 X_i는 n개의 특징으로 구성된 i번째 패턴을 의미한다. 또한 매핑되는 2차

원 공간상의 두 점간의 거리는 다음 식 13에 의해 계산된다.

$$d_{ij}=[(Y_{i1}-Y_{j1})^2+(Y_{i2}-Y_{j2})^2]^{1/2} \quad (13)$$

위 식의 Y는 Eigenvector Projection 기법 수행·도중에 계산되는 rotation matrix에서 구해진다.

Sammon 기법의 궁극적인 목표는 다음 식 14으로 표현되는 오류 함수 E를 최소로 하는 Y_{i1} 과 Y_{i2}를 반복 기법으로 구하는 것이다.

$$E(\rho)=\sum_{i<j}\frac{(d_{ij}^{*}-d_{ij})^2}{(d_{ij}^{*})^{p}} \quad (14)$$

오류를 최소화하는 과정은 신경회로망의 오류 역전도 알고리즘에서 사용되는 최급강하(steepest descent) 기법을 이용하여, 가능한 한 d_{ij}^{*}에 가까운 d_{ij}를 찾아냄으로써 각 점간의 거리를 보존하려는 시도로 간주된다. 여기에서 d_{ij}^{*}는 주어진 자료로부터 계산되는 상수이므로 사상되는 2차원 공간의 좌표를 나타내는 d_{ij}를 찾아내는 작업이 Sammon기법의 기본적인 골격인 것이다.

표 1 Monk-1 데이터의 2차원 사상 값
Table. 1 Value of 2-dimention mapping for Monk-1's data

패턴 번호	Monk-1 패턴						Sammon 기법의 결과	
	특징 1	특징 2	특징 3	특징 4	특징 5	특징 6	특징 1	특징 2
1	1	2	1	1	2	1	-0.432407	0.684113
2	1	2	1	1	3	1	-0.918071	-0.390909
3	1	2	1	1	4	2	-1.570800	-1.558010

이 기법에서 사용되는 거리의 척도는 통상적인 유클리디안 거리(Euclidean distan-ce) 를 사용하였지만, 다른 척도를 사용하여도 상관없다.

표 1은 실험에 사용한 Monk-1 데이터에 Sammon기법을 적용한 결과이며, 이를 Microsoft Excel을 사용하여 화면에 플롯한 결과는 그림 4에 나타나 있다.

하나의 패턴이 6개의 특징으로 구성되는 Monk-1 자료를 2차원 공간으로 매핑한 후, 각 클래스를 나눌 클러스터의 갯수를 정의하면 각 클러스터의 중심을 대략 추축할 수

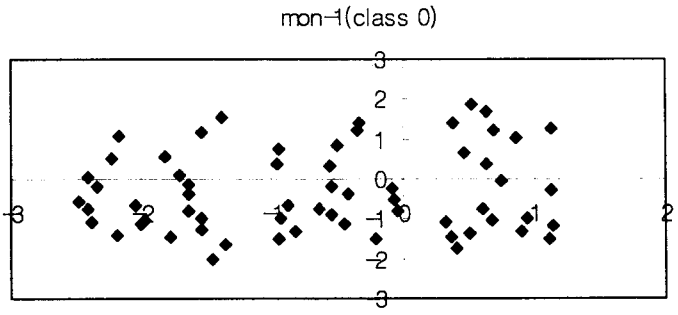


그림 4 6차원 Monk-1 데이터의 2차원 사상 결과

있게 된다. 기존의 K-평균 알고리즘은 초기의 클러스터 중심을 임의로 선정하므로 군집화의 결과가 다양한 반면에, 이 기법에서는 보다 정확한 클러스터의 중심을 제공할 수 있다는 장점을 갖는다. 또한 여러 개의 특징으로 구성되는 원시 자료만 가지고 클러스터의 개수를 정의할 때에는 단순한 작업이 아닌 반면에, 2차원 공간의 사상 결과를 화면으로 보면 대략 클러스터의 개수와 중심점을 짐작할수 있다는 장점도 갖는다.

군집화 작업이 완료되면 각 클러스터에서 임의로, 또는 중심으로부터 멀리 떨어진 패턴들을 대표 패턴 (representative pattern)으로 추출하는 작업이 가능하게 된다. 이때 각 클러스터에서 중심에 가까운 패턴을 추출하는 방법도 가능하며, 또한 클러스터의 가장자리에 위치한 패턴들을 추출하는 방법도 있으나, 본 연구에서 실험한 바에 의하면 중심으로부터 멀리 위치한 패턴들을 추출하는 방법의 성능이 우수한 것으로 판명되었다.

4.2 점진적 학습 알고리즘

전체적인 점진적 학습 과정은 다음과 같다.

단계 1 : 초기 학습 패턴 N개를 신경망에 학습시킨다.

단계 2 : 학습패턴 N개를 클래스별로 구분하여, 각 클래스를 대표하는 패턴들을 구하여 저장한다. 이때 각 클래스별로 앞절에서 설명된 Sammon 기법을 이용하여 대표 패턴을 선정한다. 대표 패턴의 개수는 M개이며, M<N 이다.

* 참고 : 단계 1과 단계2는 병행으로 수행된다.

단계 3 : 후에 추가로 학습시킬 패턴 A개가 입수되면, M+A개를 기존의 신경망에 학습시킨다. 이때 기존의 신경망은 단계 1의 결과인 가중치를 갖는 신경망을 의미한다.

단계 4 : N+A의 전체 패턴을 가지고 단계 2를 반복한다. 이때 대표 패턴의 갯수 M은 늘어나게 된다.

* 참고 : 단계 3과 단계 4는 병행으로 수행된다.

여기에서 대표 패턴을 추출하는 방법으로는 각 클래스의 패턴 개수가 작을 때는 임의로 몇개를 선정해도 크게 성능에 영향을 미치지 않겠지만, 패턴의 개수가 많을 때에는 클러스터링 기법을 사용하여야 한다. 또한 대표 패턴을 선정하는 효율적인 클러스터링 기법도 고려 되어야 하며, 대표 패턴 집합의 원소 갯수 M도 적절히 정해야 한다.

V. 성능 평가

제안된 기법의 성능을 평가하기 위하여 4가지의 실험을 수행하였다. 기계 학습의 벤치마크로 사용되는 Monk's data[8] 와 Iris Data[9]를 이용하여 가변 기울기 역전도 알고리즘이 오류 역전도 알고리즘보다 학습 속도 면에서의 효율성과 대표 패턴의 추출의 타당성을 분석하며 대표 패턴 추출에 의한 점짐적 학습에 적용해 본다.

5.1 Monk's 데이터

기계 학습 분야의 벤치마크로 많이 사용되는 Monk's 데이터는 제반 기계 학습 방법의 평가를 위해 인위적으로 만든 자료로 특징에 의해 2개의 클래스로 분류되는 로봇의 분류 문제이다. Monk's 데이터의 구성은 다음과 같다.

x_1 : head-shape {round, square, octagon}

x_2 : body-shape {round, square, octagon}

x_3 : is-smiling {yes, no}

x_4 : holding {sword, ballon, flag}

x_5 : jacket-color {red, yellow, green, blue}

x_6 : has-tie {yes, no}

각 패턴의 출력 클래스는 0 또는 1이며, 1이면 로봇임을 나탄낸다.

Monk-1

다음 조건을 만족하면 클래스 1, 아니면 클래스 0로 된다.

(head-shape= body-shape) or (jacket-color = red)

Monk-1 데이터는 216개의 클래스 0 패턴과 216개의 클래스 1 패턴으로 총 432개의 패턴으로 구성되어 있다. 그 중에서 클래스 0 패턴 62개와 클래스 1 패턴 62개를 임의로 추출하여 학습 패턴으로 사용하고 전체 432개의 패턴으로 인식 성능을 검사하였다. 여기에서 학습 패턴의 개수인 124는 Monk's 문제의 저자들이 추천한 수치임을 밝힌다.

Monk-2

다음 조건을 만족하면 클래스 1, 아니면 클래스 0가된다.

Exactly two of the six attributes have their first value.

290개의 클래스 0 패턴과 142개의 클래스 1 패턴중에서, 105개의 클래스 0패턴과 64개의 클래스 1 패턴을 임의로 추출하여 학습에 사용하고 전체 432개의 패턴으로 인식 성능을 검사하였다.

Monk-3

다음 조건을 만족하면 클래스 1, 아니면 클래스 0가 된다.

(jacket_color = greens & holding = sword)
or (jacket_color != blue &
body_shape!=octagon)

204개의 클래스 0 패턴과 228개의 클래스 1 패턴중에서, 62개의 클래스 0 패턴과 60개의 클래스 1 패턴을 임의로 추출하여 학습에 사용하고 전체 432개의 패턴으로 인식 성능을 검사하였는데, 학습 패턴에는 잘못 분류된 오류 패턴이 6개가 포함되어 있다.

5.2 Iris data

Iris데이터는 꽃잎에서 추출한 실제 패턴이며 특징은 꽃받침과 꽃잎에 대한 길이와 폭 등 4개의 특징으로 구성되며 클래스는 setosa, versicolor, virginica로서 3개의 클래스를 갖는다. 각각의 클래스는 50개의 패턴으로 구성되며 setosa는 다른 클래스로부터 잘 분리되어 있고 versicolor와 virginica는 3개의 중복된 영역을 갖는다.

5.3 가변 기울기와 역전도 알고리즘의 비교

아래의 표 2는 동일한 실험을 5번씩 반복한 후 평균낸 수치이며 Monk's data는 은닉층의 개수는 15개 초기 학습율은 0.8 모멘텀율은 0.7을 사용했으며 Iris data는 은닉층의 개수를 12개 초기 학습율은 0.3 모멘텀율

은 0.7 을 사용하였다. 그리고 전체 오류가 0.001보다 작아지면 종료하였다.

실험에 사용한 신경망의 구조는 입력 노드가 6개. 은닉층의 노드가 10개이고, 출력 노드가 2개로 구성된다. 학습에 사용된 파라메터들의 값은 시그모이드 함수의 기울기를 0.4로, 학습률을 0.7로, 모멘텀율을 0.6으로 정의하였으며, 초기 가중치의 범위를 [-1.0, 1.0]로 난수발생기를 사용하여 배정하였다.

표 2 가변 기울기와 역전도 알고리즘의 비교
Table. 2 Comparison of variable slope and backpropagation algorithm

	역전도 알고리즘		가변 기울기 역전도 알고리즘	
	학습횟수 (Epoch)	인식성능 (%)	학습횟수 (Epoch)	인식성능 (%)
Monk-1	13,224.6	98.8%	76.8	98.5%
Monk-2	19,378.7	99.0%	123.7	94.2%
Monk-3	10,442.8	90.2%	128.8	89.2%
Iris	1,537.8	90.0%	949.0	91.6%

5.4 대표 패턴 추출

전체 학습 데이터로 학습하는 경우보다는 대표 패턴만으로 학습하는 경우에 인식 성능이 떨어지는 것은 당연하며, 어느 정도나 인식 성능이 저하되는지를 검사하였다. 데이터의 형태는 6개의 특징과 두개의 클래스(01 10)를 가지며, 입력 특징의 값을 아래와 같이 0 과 1사이의 실수값으로 변형하여 학습하였다.

표 3 입력 특징값의 실수 변환
Table. 3 Floation-point transformation of input-feature's value

전(정수)	후(실수)
1	0.3
2	0.7
3	1.0
4	1.3

monk's 데이터의 경우, 각 클래스를 3개의 클러스터로 군집화 한 후 각 클러스터에서 10개의 패턴을 추출하여 총 대표 패턴의 개수는 60개이며, 3가지의 실험을 수행하여 대표 패턴의 추출의 효율성을 검사하였다.

첫째로 실험-1에서는, 허용 오차를 0.0001로 정의하였다. 전체 학습 패턴을 학습시킨 후 테스트 패턴으로 인식 성능을 검사하였다. 둘째로 실험-2에서는, 허용 오차를 0.0001로 정의하여 추출된 대표 패턴을 학습시킨 후 테스트 패턴으로 인식 성능을 검사

하였다.

마지막으로 실험-3에서는 점진적까지 학습시킨 후 가중치를 그대로 사용하여 대표패턴을 호용 오차의 0.0001까지 추가 학습하여 테스트 패턴으로 인식 성능을 검사하였다. 실험의 결과는 다음에 나타나 있다. 이 결과는 동일한 실험을 10번 반복하여 그 평균 수치를 보인 것이다.

① Monk-1 데이타

표 4 Monk-1의 실험-1의 결과 인식율 = 99.3%
Table. 4 Result of experiment-1 for Monk-1 recognition rate=99.3%

time (sec)	epoch	테스트 패턴	
		correct	wrong
247	2639	431	1
258	2733	426	6
251	2651	428	4
238	2517	427	5
237	2705	432	0
235	2662	430	2
239	2712	428	4
229	2607	427	5
223	2551	431	1
228	2603	429	3
238.0	2638.0	428.9	3.1

표 5 Monk-1의 실험-2의 결과 인식율=95.4%
Fig. 5 Result of experiment-2 for Monk-1 recognition rate=95.4%

time (sec)	epoch	학습 패턴		테스트 패턴	
		correct	wrong	correct	wrong
246	5212	120	4	409	23
230	4742	121	3	411	21
240	4933	122	2	414	18
262	5392	122	2	417	15
264	5414	122	2	416	16
237	4889	120	4	409	23
248	5092	121	3	410	22
246	5065	121	3	409	23
261	5358	121	3	413	19
259	5308	121	3	413	19
249.3	5140.5	121.1	2.9	412.1	19.9

표 6 Monk-1의 실험-3의 결과 인식율=98.9%
Table. 6 Result of experiment-3 for Monk-1 recognition rate=98.9%

학습 패턴에 대하여				대표 패턴 학습에 대하여						총 수행 시간
time (sec)	epoch	테스트 패턴		time	epoch	학습 패턴		테스트 패턴		
		correct	wrong			correct	wrong	correct	wrong	
49	531	421	11	173	3678	124	0	427	5	222
46	487	426	6	183	3895	124	0	428	4	229
43	463	423	9	174	3710	124	0	428	4	217
48	522	423	9	178	3777	124	0	427	5	226
49	512	419	13	179	3740	124	0	421	11	228
46	500	422	10	199	4246	124	0	428	4	245
46	497	426	6	182	3868	124	0	429	3	228
44	466	428	4	190	4049	124	0	431	1	234
54	574	426	6	186	3952	124	0	427	5	240
47	508	424	8	174	3686	124	0	427	5	221
47.2	506.0	423.8	8.2	181.8	3860.1	124.0	0.0	427.3	4.7	229.0

② Monk-2 데이터

표 7 Monk-2의 실험-1의 결과
인식율 = 85.6%
Table. 7 Result of experiment-1 for Monk-2
recognition rate=85.6%

time (sec)	epoch	테스트 패턴	
		correct	wrong
1370	11550	358	74
2189	15303	372	60
2052	14780	371	61
1486	12381	360	72
2103	16254	362	70
1268	9353	365	67
2457	20829	374	58
1272	10785	367	65
52099	12870	387	45
1420	12029	382	50
6771.6	13613.4	369.8	62.2

표 8 Monk-2의 실험-2의 결과 인식율=57.4%
Table. 8 Result of experiment-2 for Monk-2
recognition rate=57.4%

time (sec)	epoch	학습 패턴		테스트 패턴	
		correct	wrong	correct	wrong
482	10843	103	66	235	197
627	14135	116	53	261	171
497	11252	106	63	224	208
563	12666	113	56	253	179
646	14570	116	53	260	172
439	9912	102	67	227	205
554	12509	123	46	266	166
669	14808	118	51	266	166
729	16266	105	64	227	205
566	12820	119	50	260	172
577.2	12978.1	112.1	56.9	247.9	184.1

표 9 Monk-2의 실험-3의 결과 인식율=82.1%
Table. 9 Result of experiment-3 for Monk-2
recognition rate=82.1%

학습 패턴에 대하여				대표 패턴 학습에 대하여						총 수행 시간
time (sec)	epoch	테스트 패턴		time	epoch	학습 패턴		테스트 패턴		
		correct	wrong			correct	wrong	correct	wrong	
479	4051	353	79	52	1183	162	7	345	87	531
603	3777	366	66	164	2991	162	7	357	75	767
273	2314	343	89	106	2396	160	9	340	92	379
319	2701	374	58	77	1757	160	9	353	79	396
423	3280	363	69	170	3841	164	5	357	75	593
309	2587	360	72	417	7238	158	11	349	83	726
669	4693	369	63	72	1621	162	7	361	71	741
348	2579	373	59	221	4391	157	12	360	72	569
408	3461	367	65	121	2721	161	8	360	72	529
610	5133	365	67	44	994	165	4	363	69	654
444.1	3457.6	363.3	68.7	144.4	2913.3	161.1	7.9	354.5	77.5	588.5

③ Monk-3 데이터

표 10 Monk-3의 실험-1의 결과
인식율=86.8%
Table. 10 Result of experiment-1 for Monk-3
recognition rate=86.8%

time (sec)	epoch	테스트 패턴	
		correct	wrong
583	6658	381	51
754	8760	377	55
488	5669	370	62
499	5798	380	52
707	8218	371	61
486	5642	372	60
540	6269	376	56
597	6930	378	54
531	6159	377	55
599	6497	368	64
574.4	6660.0	375.0	57.0

표 11 Monk-3의 실험-2의 결과 인식율=78.1%
Table. 11 Result of experiment-2 for Monk-3 recognition rate=78.1%

time (sec)	epoch	학습 패턴		테스트 패턴	
		correct	wrong	correct	wrong
554	12359	105	17	313	119
388	8754	112	10	341	91
380	8587	114	8	338	94
503	11343	111	11	332	100
411	9296	114	8	344	88
428	9165	112	10	333	99
527	11912	112	10	337	95
379	8587	111	11	345	87
439	9912	109	13	344	88
408	9203	108	14	349	83
441.7	9911.8	110.8	11.2	337.6	94.4

표 12 Monk-3의 실험-3의 결과 인식율=84.9%
Table. 12 Result of experiment-3 for Monk-3 recognition rate=84.9%

학습 패턴에 대하여				대표 패턴 학습에 대하여						총 수행 시간
time (sec)	epoch	테스트 패턴		time	epoch	학습 패턴		테스트 패턴		
		correct	wrong			correct	wrong	correct	wrong	
134	1552	384	48	212	4782	120	2	376	56	346
102	1185	377	55	260	5899	120	2	375	57	362
113	1305	377	55	212	4791	119	3	377	55	325
226	2624	364	68	444	10018	119	3	355	77	670
123	1433	371	61	239	5375	118	4	367	65	362
115	1340	369	63	244	5488	120	2	358	74	359
130	1515	370	62	248	5600	119	3	371	61	378
133	1502	370	62	245	5377	119	3	361	71	378
106	1231	369	63	206	4654	120	2	361	71	312
128	1476	373	59	204	4598	119	3	366	66	332
131.0	1516.3	372.4	59.6	251.4	5658.2	119.3	2.7	366.7	65.3	382.4

④ Iris 데이터

실험에 사용된 신경망의 구조는 입력 노드가 4개, 은닉층의 노드가 10개이고, 출력 노드가 3개로 구성된다. 학습에 사용된 파라메터들의 값은 시그모이드 함수의 기울기를 0.4로, 학습율을 0.5로, 모멘텀율을 0.7으로 하였으며 초기 가중치의 범위는 [-3.0,3.0]로 난수발생기를 사용하여 배정하였다. Iris데이터의 경우, 각 클래스를 2개의 클러스터로 군집화한 후, 각 클러스터에서 5개의 패턴을 추출하여 총 대표 패턴의 개수는 30개이며, 3가지의 실험을 수행하여 대표 패턴 추출의 효율성을 검사하였다. 전체 학습 패턴의 개수는 73개이고, 테스트 패턴은 74개이다.

표 13 Iris의 실험-1의 결과 인식율=93.2%
Table. 13 Result of experiment-1 for Iris recognition rate=93.2%

time (sec)	epoch	테스트 패턴	
		correct	wrong
107	2031	69	5
108	2050	69	5
108	2059	69	5
116	2077	69	5
107	2026	69	5
100	1846	69	5
101	1912	69	5
130	2462	69	5
102	1934	69	5
103	1960	69	5
108.2	2033.6	69.0	5.0

표 14 Iris의 실험-2의 결과 인식율=94.5%
Table. 14 Result of experiment-2 for Iris recognition rate=94.5%

time (sec)	epoch	학습 패턴		테스트 패턴	
		correct	wrong	correct	wrong
98	4126	73	0	71	3
92	3813	72	1	69	5
75	3176	72	1	70	4
78	3306	72	1	70	4
82	3438	72	1	70	4
119	4949	72	1	70	4
84	3406	72	1	69	5
71	3021	72	1	70	4
69	2926	72	1	70	4
77	3238	72	1	70	4
84.5	3541.9	72.1	0.9	69.9	4.1

표 15 Iris의 실험-3의 결과 인식율=94.6%
Table. 15 Result of experiment-3 for Iris recognition rate=94.6%

학습 패턴에 대하여				대표 패턴 학습에 대하여						총 수행 시간
time (sec)	epoch	테스트 패턴		time	epoch	학습 패턴		테스트 패턴		
		correct	wrong			correct	wrong	correct	wrong	
36	664	69	5	57	2420	73	0	69	5	93
41	771	70	4	84	3551	73	0	70	4	125
42	801	69	5	43	1808	73	0	70	4	85
39	741	69	5	43	1789	73	0	70	4	82
41	774	69	5	65	2706	73	0	70	4	106
42	781	70	4	87	3668	73	0	71	4	129
37	704	70	4	63	2369	73	0	70	3	100
35	661	69	5	63	2617	73	0	70	4	98
45	849	69	5	44	1823	73	0	70	4	89
42	797	69	5	47	1930	73	0	70	4	89
40.0	754.3	69.3	4.7	59.6	2468.1	73.0	0.0	70.0	4.0	99.6

Iris 데이터의 경우에도, Monk's 데이터 실험과 동일한 실험과 동일한 결과를 얻을 수 있었으며 특히 자료의 분포가 인식율과 학습 시간의 측면에서 결과에 영향을 많이 미치는 것을 주목할 수 있다. Monk-1과 Monk-3 데이터와 Iris데이터의 경우, 대표 패턴을 효율적으로 추출할 수 있었던 반면에, Monk-2 데이터의 경우에는 분포의 복잡성이 그 효율을 저하시킨 요인으로 사료된다. 한가지 해결책은 충분한 실험 분석에 의한 파라메터의 산출이라고 사료된다.

실험의 결과를 인식율과 학습에 소요되는 시간 측면에서 종합하면 다음과 같다.

표 16 대표 패턴 추출의 실험결과(인식율)
Table. 16 Experiment result of representative-pattern's extraction(recognition rate)

데이터	인식율(%)		
	실험 1	실험 2	실험 3
Monk-1	99.3	95.4	98.9
Monk-2	85.6	57.4	82.1
Monk-3	86.6	78.1	84.9
Iris	93.2	94.5	94.6

표 17 대표 패턴 추출의 실험결과(학습시간(초))
Table. 17 Experiment result of representative-pattern's extraction(learning time(/sec))

데이타	학습 시간(초)		
	실험 1	실험 2	실험 3
Monk-1	238.0	249.3	229.0
Monk-2	6771.6	577.2	588.5
Monk-3	574.4	441.7	382.4
Iris	108.2	84.5	99.6

전체 학습 패턴을 실험-1의 결과가 가장

좋은 것은 당연하지만, 이 결과를 보면, 대표 패턴을 추출한 효율성을 입증할 수 있다. 대표 패턴만을 학습한 실험-2의 경우, 인식율이 많이 저하된 사실이 주목되지만, 학습 패턴과 대표 패턴을 동시에 사용한 실험-3의 경우, 짧은 시간에 실험-1에 필적 할 만한 인식율을 얻었다는 사실이 대표 패턴 추출의 타당성을 확인해 주는 결과이다.

5.5 Monk's 데이터의 추가 학습

점진적 학습 알고리즘을 검증하려면, 초기의 학습 패턴외에도 추가 학습 패턴이 있어야 하는데 monk's 데이터를 이용하여 간단한 실험을 수행하였다.

초기에는 전체 학습 패턴으로 학습을 수행한 후, 인식 성능 검사 단계에서 틀린 답을 출력하게 된 테스트 패턴을 추가로 학습시키는 실험을 수행하였다. 이때 추가 학습은 미리 추출한 대표 패턴과 추가 학습 패턴만으로 학습을 행하게 된다. 이 실험의 결과는 상당히 고무적이었으나, Monk-1데이터의 분포가 이러한 결과에 영향을 주었다고 사료된다. 다음은 실험의 결과이다.

위의 결과는 동일한 실험을 10번 수행한 평균 수치이며, 인식율이 초기의 98.0%에서 추가 학습후에는 99.6%로 향상되었다. 그림 4에 나타난대로, Monk-1 데이터는 패턴들이 상당히 전역에 걸쳐 분산되어 있는 자료이므로 대표 패턴의 추출이 용이한 작업이다. 이러한 특성이 좋은 결과에 공헌하였다고 사료되며, 다른 데이터에 추가 학습 기법을 사용해 본 결과는 고무적이지 못하였다.

표 18 Monk-1의 실험결과(추가학습)
Table. 18 Experiment result of Mink-1(Incremental Learning)

124개 패턴 학습			추가 학습				
Epoch	test패턴에 대하여		학습 패턴갯수	Epoch	124개 패턴에서 없어진 패턴의 갯수	test패턴에 대하여	
	Correct	Wrong				Correct	Wrong
104	420	12	52	634	0	432	0
89	426	6	46	701	1	429	3
83	426	6	46	558	1	429	3
98	421	11	51	419	0	432	0
98	418	14	54	480	0	432	0
87	421	11	51	521	0	430	2
99	425	7	47	726	0	428	4
89	427	5	45	520	0	429	3
91	426	6	46	558	0	432	0
72	423	9	49	389	0	431	1
910/10=91.0	4233/10=423.3	87/10=8.7	487/10=48.7	5506/10=550.6	2/10=0.2	4304/10=430.4	16/10=1.6

VI. 결 론

학습해야 할 데이터가 주기적으로 발생하는 경우에 기하 급수적으로 늘어나는 학습 시간을 줄이기 위해 대표 패턴 추출의 타당성과 대표 패턴을 이용한 추가 학습이 학습 속도면이나 인식 성능면에서 결코 뒤떨어지지 않음을 확인했다. 그러나 점진적 학습을 수행시킨 결과에서 보듯이 Monk-1 데이터만이 만족할 만한 결과가 나왔다. 이 결

과는 데이터의 분포가 많은 영향을 가져온 것으로 판단되며 데이터의 분포가 전역에 걸쳐 고르게 분포될 수 있도록 전 처리하는 과정에 대한 연구가 필요하며 대표 패턴 추출에 있어 패턴의 분포나 대표 패턴의 개수 클러스터의 개수 등이 성능에 영향을 미칠 수 있는 중요한 변수이기 때문에 이런 파라미터들의 선정이 중요한 관건이 될 수 있으며 이런 변수들은 많은 실험에 의해 선정될 수 있을 것이다.

참 고 문 헌

[1] D. E. Rumelhart, J. L. McCelland, Parallel Distributed Processing, Vol. 1. pp. 318-330, The MIT Press, Cambrige, MA, 1986.

[2] K. Fukunaga, W. L. G. Koontz, IEEE Trans. Comput, C-19, p. 311, 1970.

[3] E. A. Patrick, D. R. Anderson, F. K. Bechtel. IEEE Trans. Comput. C-17, p. 949, 1968.

[4] K. Fukunaga, D. R. Olsen, Pattern Recong., p. 917, 1971.

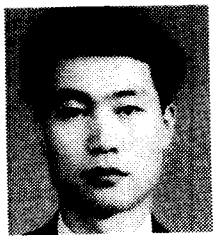
[5] J. W. Sammon, Jr., A. H. Proctor, D. F. Roberts, Pattern Recog., pp. 3, 37, 1971.

[6] S. B.. Thrun. et. al., "The Monk's Problems: A Performance Comparison of Different Learning Algorithm", CMU-CS-91-197, December, 1991.

[7] M. Nadler, E. P. Smith, Pattern Recognition Engineering, Wiley Interscience, 1993.

[8] 방승양, 뉴로 컴퓨터, 상조사, 1992년 10월

□ 筆者紹介



심범식
1985년 명지대학교 전자계산학과 학사
1987년 명지대학교 전자계산학과 석사
1994년 명지대학교 컴퓨터공학과 박사과정 수료
1991년 ~ 현재 마산전문대학 사무자동화과 조교수

윤충화
1979년 서울대학교 수학과 학사
1984년 미국 텍사스대 전자계산학과 석사
1989년 미국 루이지아나 주립대 전자계산학과 박사
1990년 ~ 현재 명지대학교 컴퓨터 공학과 부교수