데이터 마이닝을 위한 경쟁학습모델과 BP알고리즘을 결합한 하이브리드형 신경망

강문식* · 이상용**

A Neural Network Combining a Competition Learning Model and BP ALgorithm for Data Mining

Moon-Shik Kang* · Sang-Yong Lee**

Abstract

Recently, neural network methods have been studied to find out more valuable information in data bases. But the supervised learning methods of neural networks have an overfitting problem, which leads to errors of target patterns. And the unsupervised learning methods can distort important information in the process of regularizing data. Thus they can't efficiently classify data.

To solve the problems, this paper introduces a hybrid neural networks HACAB(Hybrid Algorithm combining a Competition learning model And BP Algorithm) combining a competition learning model and BP algorithm. HACAB is designed for cases which there is no target patterns. HACAB makes target patterns by adopting a competition learning model and classifies input patterns using the target patterns by BP algorithm.

HACAB is evaluated with random input patterns and Iris data. In cases of no target patterns, HACAB can classify data more effectively than BP algorithm does.

* 공주대학교 전자계산학과 이학석사

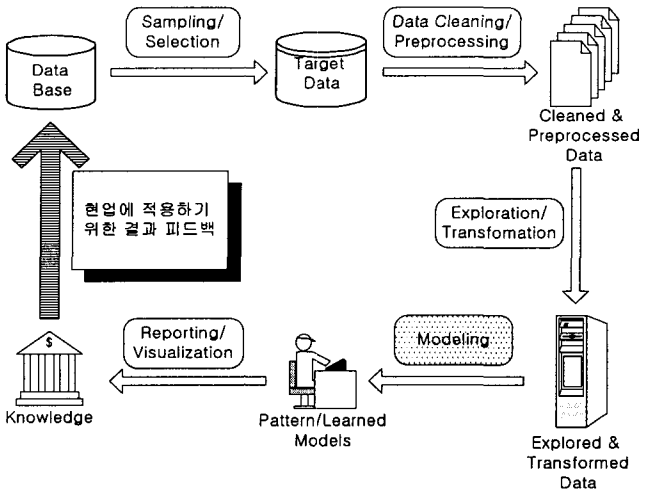
** 공주대학교 정보통신공학부 컴퓨터전공 부교수

1. 서론

전통적인 전문가시스템의 한계와 대용량 데이터의 급증으로 인하여 데이터 마이닝이 주목받고 있다. 데이터 마이닝은 데이터베이스로부터 데이터 간의 연관성을 분석하여, 그 결과들로부터 유용한 정보를 발견하는 하는 방법론으로써 현재 이에 대한 연구들이 활발하게 진행되고 있다[9-11].

데이터마이닝을 위해서는 샘플 추출(Sampling / Selection), 데이터 정제 및 전처리(Data Cleansing / Preprocessing), 탐색 및 변형(Exploration / Transformation), 모형화(Modeling), 보고 및 시각화(Reporting / Visualization)의 과정을 거치게 된다(그림 1).

본 논문에서는 데이터마이닝 과정 중 가장 중요한 모형화 단계를 대상으로, 입력 패턴에 대해 목적 패턴이 존재하지 않는 경우에 데이터를 효율적으로 분류할 수 있는 모델을 제안하였다. 데이터마이닝을 위한 모형화 단계의 학습 기법으로는 기계 학습과 통계적 기법 등이 있는데, 전통적인 통계적 기법을 통한 데이터 분석의 한계로 인하여 최근 기계 학습 방법이 활발하게 연구되고 있다. 본 연구에서는 기계 학습 방법 중, 패턴 분류에 뛰어난 성능을 가진 신경망 모델을 사용하였다. 본 연구에서 제안한 신경망 모델은 감독학습 방법으로 가장 일반적으로 사용되는 BP알고리즘에 목적 패턴을 생성하기 위하여 경쟁 학습 모델인 인스타 규칙을 결합한 하이브리드형 신경망으로 구성되어 있다.



(그림 1) 데이터마이닝 수행 과정

2. 관련 연구

2.1 데이터 마이닝

21세기 기업경영에서 데이터베이스 마케팅(Database Marketing), 고객관계관리(CRM: Customer Relationship Management), 위험관리(Risk Management)등의 중요성이 크게 부각되기 시작하면서 다양한 분야에서 데이터 마이닝이 사용되고 있다[6].

그러나 거대한 양의 데이터를 분석하기란 쉽지 않기 때문에, 많은 기업들은 단순한 통계 정보만을 얻는데 그쳐서 귀중한 데이터를 제대로 활용하지 못하고 있다. 이러한 분석 작업을 지원하는 정보기술이 데이터 마이닝으로[3], 데이터 마이닝 시스템이란 다양한 분야의 이론 및 알고리즘들을 통합하여 데이터로부터 유용한 지식을 추출해내는 총체적인 시스템을 의미한다[8][9].

데이터 마이닝 기법은 크게 통계적 방법, 기계 학습 등으로 나눌 수 있는데, 통계학적인 방법은 많이 실용되고 있는 상태이고, 최근에는 기계 학습 방법이 많이 연구되고 있다[13].

기계 학습은 주로 의사 결정 트리, 사례 기반 학습, 연관 규칙, 유전자 알고리즘, 신경망 등이 있으며, 특히, 신경망은 복잡한 데이터 사이의 관계나 패턴 도출에 유용하게 사용되고 있다<표 1>[2][4][5][7][12].
신경망은 보통 감독 학습 방법(Supervised Learning Method)과 비감독 학습 방법(Unsupervised Learning Method)으로 나누어지는데, 전자는 목적 패턴이 존재하고, 후자는 목적 패턴이 존재하지 않는 신경망을 말한다[1]. 본 연구에서는 데이터마이닝의 모형화 단계에서 목적 패턴이 존재하지 않는 경우를 대상으로, 비감독 학습 방법인 경쟁학습 모델과 감독 학습 방법인 BP알고리즘을 결합한 하이브리드형 신경망 모델을 제안하였다.

<표 1> 주요한 기계 학습 방법의 장단점

	장 점	단 점
의사 결정 트리	· 분석자가 과정을 쉽게 이해하고 설명 가능	· 데이터의 변화에 따라 새로운 의사 결정 트리 구축이 필요
사례 기반 학습	· 의사결정지원, 예측, 진단, 계획 등의 다양한 영역에 사용 가능	· 지식 획득의 병목 현상 초래 · 사례 적응 단계의 구현이 곤란
연관규칙	· 장바구니 분석 문제들에 적용	· 부분구간의 규칙 추출이 곤란 · 응답 속도가 느림
유전자 알고리즘	· 최적해 부근으로의 수렴은 빠른 속도로 이행	· 최적해 수렴에 시간이 걸림 · 학습이 어려움
신경망	· 예측 문제에 유용 · 복잡한 데이터 사이의 관계나 패턴 도출에 유용	· 결과에 대한 설명 능력 부족

2.2 경쟁 학습 모델

본 논문에서는 감독식 학습 방법인 BP알고리즘의 목적패턴을 생성하기 위하여 비감독학습 방법인 경쟁학습 모델을 사용하였다.

경쟁 학습 모델의 가장 큰 특징은 경쟁에서 이긴 승자 신경세포의 연결 가중치만이 주어진 입력 패턴에 대해 조절될 권리를 가지게 되는 승자전취 메커니즘(Winner Takes All)을 사용하고 있다는 것이다. 승자 신경세포로 선정되면 활성 값이 1로 되고, 나머지 다른 신경세포들의 활성값은 0이 된다. 이때 승자 신경세포는 한 개이거나 여러 개일 수도 있다.

본 연구에서는 경쟁 학습을 이용하여 목적 패턴만을 생성하면 되기 때문에, 복잡한 계산 시간과 학습 시간을 고려하여 경쟁 학습 모델의 기본이 되는 인스타 규칙을 이용하였다. 인스타 규칙은 면(Group)개념을 사용하고 있는데, 면이란 같은 층 내의 신경세포들을 무리지어 놓은 것을 말한다. 같은 면내에 속한 신경세포들끼리 경쟁을 하는데, 한 면에서는 하나의 승자 신경세포만이 생성될 수 있다.

(그림 2)는 인스타 규칙의 모델로서, 다른 모든 신경망의 학습 규칙과 마찬가지로 연결 가중치를 조절하는 것으로, 어떤 신경세포가 특정 연결을 자극하면 그것의 연결 가중치를 그 자극과 같아지도록 조절하게 된다.

이러한 사실을 식으로 표현하면 다음과 같다.

$w(new)_{ij} = w(old)_{ij} + \alpha(a_i - w(old)_{ij})$

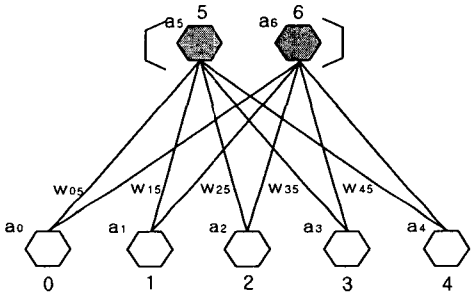
$w(new)_{ij}$: 신경 세포 i, j 사이의 조절된 후 연결 가중치

$w(old)_{ij}$: 신경세포 i, j 사이의 조절되기 전 연결 가중치

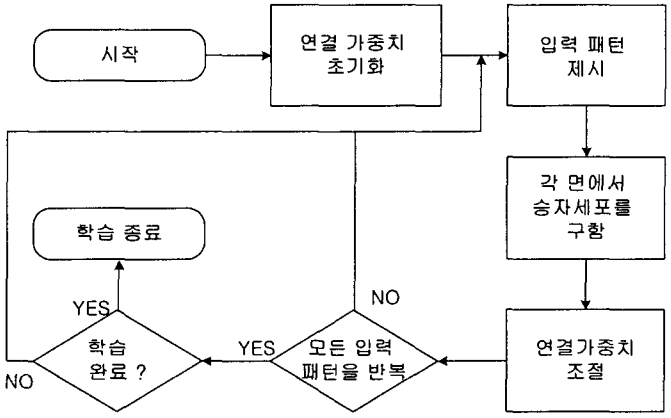
α : 학습률($0 < \alpha <= 1$)

a_i : 신경세포 i의 활성값

위 식에서 알 수 있듯이 목적 패턴은 사용되지 않으며, 신경세포 j의 연결 가중치 w_{ij}는 단지 그것에 달린 신경세포 a_i의 활성값과 현재 연결 가중치의 차이에 비례하여 조절될 뿐이다.



(그림 2) 인스타 규칙 모델



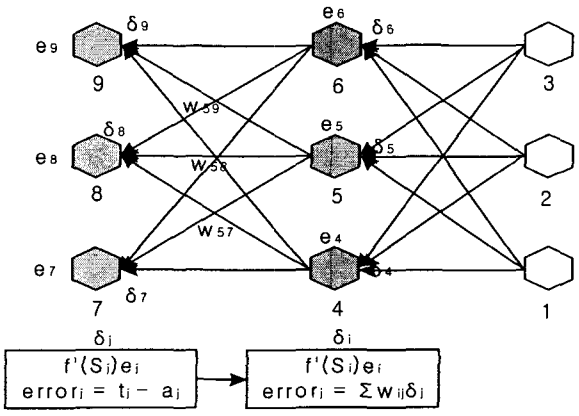
(그림 3) 경쟁 학습 모델의 학습 흐름도

인스타 규칙을 사용하는 경쟁 학습 모델에서 신경망을 학습시키는 과정은 (그림 3)과 같다. 그림에서 "학습 완료?"는 학습 종료 조건을 확인하는 부분으로, 여러 번의 반복 학습을 거쳐서 어느 시점에서 에러값이 더 이상 줄지 않을 때 학습을 종료하게 된다. 본 논문에서는 학습 종료 후 출력되는 패턴은 다음 절에서 설명하는 감독식 학습방법인 BP알고리즘의 목적 패턴으로 사용하게 된다.

인스타 규칙을 사용하는 경쟁 학습 모델은 어느 신경세포의 연결 가중치가 다른 신경세포들의 값에 비해 현저하게 클 경우, 모든 입력 패턴에 대해 반응해 버리는 경우가 발생한다. 이러한 경우를 방지하기 위해서 신경세포들의 연결 가중치를 초기화한 후, 각각의 신경세포가 가진 연결 가중치들의 합이 일정한 값을 넘지 못하도록 하는 방법을 사용하기도 한다. 이것을 연결 가중치의 정규화(Weight Normalization)과정이라고 하는데, 이러한 과정은 서로 다른 입력 패턴을 변질시켜서 같은 패턴으로 만들 수도 있다[3]. 그러나 본 논문에서 사용한 경쟁 학습 모델은 최종적인 패턴 분류가 목적이 아니라, BP알고리즘에 적용시킬 목적 패턴만을 생성하면 되기 때문에 정규화 과정이 필요없게 된다. 따라서 본 논문에서 제안한 HACAB은 정규화의 위험을 피하면서 경쟁학습 모델을 활용하여 목적패턴을 생성하게 된다.

2.3 BP알고리즘

다층 퍼셉트론(Multi Layer Perceptron)에 일반화된 델타 규칙(Generalizion Delta Rule)을 학습 규칙으로 사용한 감독 학습 방법의 알고리즘을 BP알고리즘(Backpropagation Algorithm)이라 한다. 이 알고리즘은 "만일 어떤 신경세포의 활성이 다른 신경세포의 잘못된 출력에 공헌을 하였다면, 두 신경세포 간의 연결 가중치를 그것에 비례하여 조절해 주어야 하며, 이러한 과정은 그 아래에 있는 신경세포들까지 계속된다"는 특징을 가지고 있다[1][2]. 이와 같이 출력 층에서 발생한 에러를 아래층으로 역전파시키므로 오류 역전파 알고리즘이라 한다[3].



(그림 4) BP 알고리즘 모델

(그림 4)는 일반적인 BP알고리즘의 모형으로 복잡함을 피하기 위해 은닉층이 하나만 있는 경우를 고려하였다. 목적 패턴에서 출력 신경세포의 활성값을 뺀 값이 바로 해당 출력 신경세포의 에러가 된다.

(그림 4)에서 e_7, e_8, e_9 로 나타낸 것이 각 출력 신경세포들의 에러이다. 에러를 여러 번 가공하여 각각의 출력층 신경세포에 대해 델타(δ)를 구하게 된다. 그리고 f'(S)는 활성함수의 미분값을 말하며 여기에서는 시그모이드 함수를 썼다. (그림 4)에서 δ_7, δ_8, δ_9로 나타낸 것이 해당 출력층 신경세포들의 델타가 되고, 그 수식은 다음과 같다.

$w(new)_{ij} = w(old)_{ij} + \alpha\delta_j a_i + \beta\Delta w_{ij}(old)$

$bias(new)_{ij} = bias(old)_{ij} + \alpha\delta_j \cdot 1 + \beta\Delta bias_{ij}(old)$

$\delta_j = a_j(1 - a_j)e_j$

$e_j = \begin{cases} t_j - a_j & \leftarrow \text{출력층 신경세포의 경우} \\ \sum_k w_{jk}\delta_k & \leftarrow \text{은닉층 신경세포의 경우} \end{cases}$

$w(new)_{ij}$: 신경세포 i, j 사이의 조절된 후 연결 가중치

$w(old)_{ij}$: 신경세포 i, j 사이의 조절되기 전 연결 가중치

α : 학습률($0 < \alpha \leq 1$)

β : 모멘텀 상수

bias : 바이어스

δ_j : 신경세포 j의 델타

a_i : 신경세포 i의 활성값

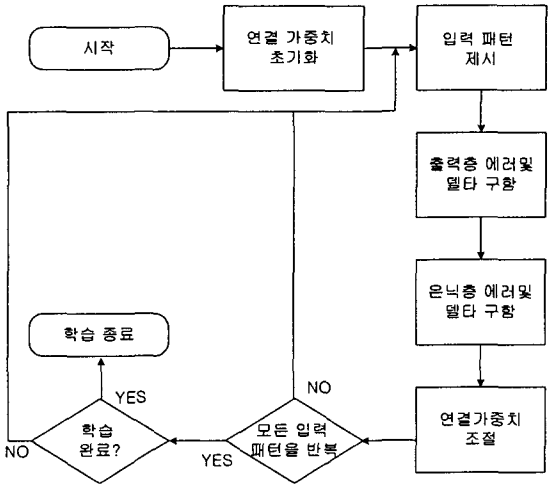
a_j : 신경세포 j의 활성값

e_j : 신경세포 j의 에러

t_j : 신경세포 j가 출력 층인 경우 해당 목적 패턴의 성분값

w_{jk} : 신경세포 j가 은닉 층인 경우 위 층 신경세포 k에 달린 연결 가중치

δ_k : 신경세포 j가 은닉 층인 경우 위 층 신경세포 k의 델타



(그림 5) BP알고리즘의 학습 흐름도

여기에서 주의해야 할 것은 δ_j 는 신경세포 j의 에러로부터 구해지는데 신경세포 j가 출력층이냐 은닉층이냐에 따라 에러를 구하는 방법이 달라진다는 것이다. BP알고리즘에서 시그모이드 함수를 활성함수로 쓰면, 에러는 보통 0에서 1사이의 값을 가져야 학습되었다고 말할 수 있다. 수식 중 $e_j = t_j - a_j$ 을 보면 목적 패턴 t가 존재하는데, 이 목적 패턴은 입력 패턴을 분류할 수 있게 인위적으로 주어지는 것이다.

BP알고리즘을 사용하여 신경망 학습하는 과정은 (그림 5)와 같다. 그림에서 '학습 완료?'는 학습의 종료 조건을 확인하는 부분으로, 여러 번의 반복 학습을 거쳐서 에러가 어느 시점에서 0과 1사이의 값이 나오게 되면, 학습이 종료하게 된다.

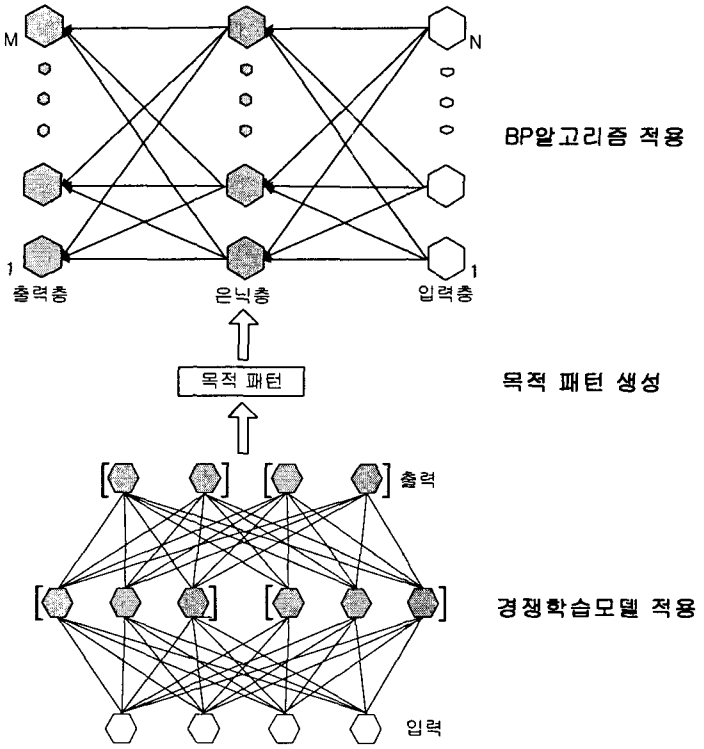
BP알고리즘은 훈련데이터 집합에 대하여 학습 후에는 훌륭한 성능을 보이지만, 새로운 데이터에 대해서는 성능이 저하되는 오버피팅(Overfitting) 현상이 발생할 수 있다. 그러나 본 논문에서는 경쟁 학습 모델에 의하여 목적 패턴이 생성되기 때문에, 인위적으로 생성된 목적 패턴으로 인한 오버피팅을 방지할 수 있다. 그리고 BP알고리즘은 학습 소요 시간이 비교적 오래 걸리지만, 일단 학습이 끝나면 응용 단계에서는 결과가 매우 빠르게 출력된다. 이러한 특성으로 BP알고리즘은 데이터 마이닝에 유용할 수 있다.

3. HACAB

본 논문에서는 데이터 마이닝을 위하여 경쟁 학습 모델과 BP알고리즘을 결합한 하이브리드형 신경망 모델인 HACAB(Hybrid Algorithm Combining a Competition Leaning Model and BP Algorithm)을 제안하였다.

HACAB은 입력 패턴에 대해 목적 패턴이 존재하지 않는 경우를 대상으로, 경쟁학습 모델을 이용해서 목적 패턴만을 생성하고 이를 BP알고리즘의 목적 패턴으로 사용하여 데이터를 분류하였다.

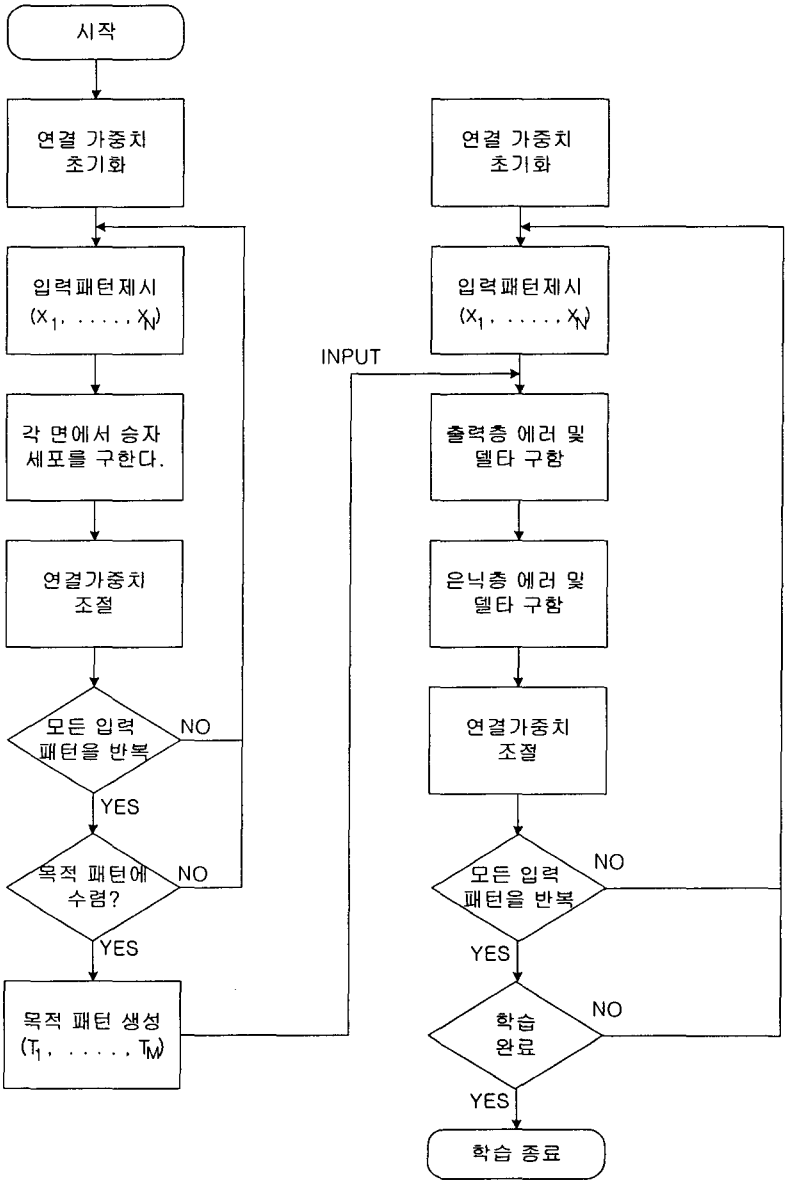
HACAB은 경쟁 학습 모델의 인스타 규칙과 BP알고리즘을 결합하여 만든 알고리즘으로써 네트워크 모형은 (그림 6)과 같다.



(그림 6) HACAB의 모형

HACAB은 경쟁 학습 모델인 인스타 규칙을 적용하여 주어진 입력 패턴에 대하여 목적 패턴을 생성할 때까지 학습을 진행시킨다. 인스타 규칙을 통하여 생성된 목적 패턴을 이용하여 BP알고리즘은 효율적으로 데이터를 분류하게 된다.

HACAB의 학습 흐름도는 (그림 7)과 같다. (그림 7)에서 보는 바와 같이 HACAB은 먼저 경쟁 학습 모델의 네트워 연결 가중치를 초기화하고, 입력 패턴(X_1, X_2, …, X_N)을 제시하여 승자 세포를 구한다. 그 다음 경쟁 학습 모델의 연결 가중치를 조절하고, 모든 입력 패턴의 반복 학습 후에 에러의 수치가 더 이상 변하지 않는다면 목적 패턴에 수렴한 것이라고 볼 수 있고 경쟁 학습 모델에서의 학습이 완료되게 된다. 이 때 목적 패턴(T_1, T_2, …, T_N)을 생성하며, 생성된 목적 패턴은 BP알고리즘의 목적 패턴으로 입력된다. BP알고리즘에서는 경쟁학습 모델에서 목적 패턴이 생성되기 전에 연결 가중치를 초기화한다. 그리고나서 입력 패턴(X_1, X_2, …, X_N)을 제시한 후에 출력값을 얻어 낸 후 생성된 목적 패턴과 비교하여 에러값을 구하게 된다. 이 때 에러값이 0~1 사이의 값이 나오지 않는다면, 만족하는 값(0~1)이 나올 때까지 연결가중치를 다시 조절하게 된다.



(그림 7) HACAB의 흐름도

4. 실험 및 분석

본 논문에서 제안한 HACAB에 대하여 임의의 입력 패턴과 공인된 아이리스 데이터(Iris Data)에 대하여 실험한 후 그 결과를 분석해 보았다.

4.1 임의의 입력 패턴

임의의 입력 패턴은 20개의 레코드로 구성되고 각 레코드는 두 개의 인자를 가지고 있다<표 2>. 그리고 좌표 평면에 나타내기 위해서 두 개의 인자는 -1에서 1사이의 값 중에서 임의로 추출하였고, 각 레코드를 4가지의 패턴인 (1 0 0 0), (0 1 0 0), (0 0 1 0), (0 0 0 1)으로 분류하였다.

<표 2> 임의의 입력 패턴

레코드 번 호	입력패턴		레코드 번호	입력패턴	
0	-0.17	0.00	10	0.31	-0.09
1	-0.35	0.18	11	-0.21	-0.1
2	-0.7	-0.8	12	-0.5	0.62
3	-0.65	-0.55	13	-0.49	0.52
4	-0.9	0.19	14	0.61	0
5	0.6	-0.35	15	-1	0.98
6	0.75	0.7	16	0.07	0.72
7	0.23	0.36	17	-0.72	-0.32
8	0.67	0.54	18	0.29	-0.34
9	0.23	0.5	19	-0.61	0.41

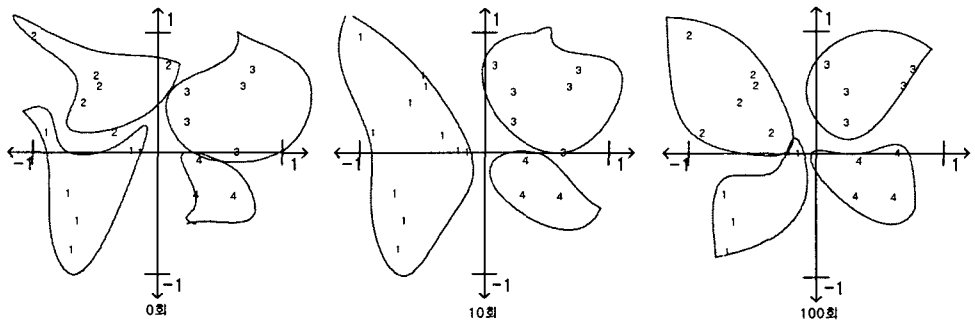
<표 2>의 입력 패턴을 경쟁 학습 모델의 인스타 규칙으로 학습을 시키게 되면 각 입력패턴에 대하여 목적 패턴은 <표 3>과 같이 생성된다. 여기에서 최적의 목적 패턴을 생성도 중요하지만 학습 반복 횟수도 고려되어야 한다. 여러 번의 반복적인 학습 결과 두드러진 특징을 나타내는 것은 0, 10, 100번의 학습 횟수였다.

본 논문에서 제안한 경쟁학습 모델에서는 정규화 과정이 필요없다. <표 2>를 보면 20개의 레코드로 구성된 입력 패턴들이 있고, 각 레코드의 입력 패턴별로 학습을 시도하였다. 따라서 어떤 레코드 중 한 입력 값으로 인하여 발생하는 연결가중치의 편중은 목적패턴 생성에 영향을 미치지 않는다. 왜냐하면 본 실험에서는 (1 0 0 0), (0 1 0 0), (0 0 1 0), (0 0 0 1)의 형태의 목적 패턴을 생성하며, 목적 패턴의 생성 결과가 (0.99 0.01 0.01 0.01), 또는 (0.89 0.11 0.11 0.11)의 형태라 할지라도 모두 동일한 목적 패턴인 (1 0 0 0)으로 수렴시킨다. 그러므로 연결가중치의 편중이 발생한다면 예를 들어 (0.89 0.11 0.11 0.11)이 (0.99 0.01 0.01 0.01)의 정도로 값의 편중이 초래될 뿐이다. 즉 목적패턴의 요소인 1이 0으로, 또는 0이 1로 바뀌는 경우는 일어날 수 없기 때문에 정규화의 과정이 필요없게 된다.

<표 3>를 이용해서 목적 패턴 분포도를 좌표평면상에 그리게 되면 (그림 8)과 같이 된다. 이 때 4가지의 패턴 분류를 한 눈에 확인하기 위하여 각각의 목적 패턴은 좌표평면 상에 나타내었다. 예를 들어 목적 패턴 (1 0 0 0)은 1의 위치가 첫 번째임으로 1로, (0 1 0 0)은 1의 위치가 두 번째임으로 2로, (0 0 1 0)은 3으로, (0 0 0 1)은 4로 표현하였다. 만약, 입력 패턴(0.5, 0.5)에 대한 목적 패턴(0 0 1 0)의 점을 표현하려면, 좌표평면상에 입력 패턴을 (x, y)놓고 점을 찍어주면 되는데 목적 패턴의 3번째 인자 1이므로, 좌표 (0.5, 0.5)를 좌표 평면 상에 3이라는 숫자로 나타내면 된다.

<표 3>학습횟수에 따른 목적 패턴

번호	입력패턴		학습횟수 0번 목적패턴	학습횟수 10번 목적패턴	학습횟수 100번 목적패턴
0	-0.17	0.00	1 0 0 0	1 0 0 0	1 0 0 0
1	-0.35	0.18	0 1 0 0	1 0 0 0	0 1 0 0
2	-0.7	-0.8	1 0 0 0	1 0 0 0	1 0 0 0
3	-0.65	-0.55	1 0 0 0	1 0 0 0	1 0 0 0
4	-0.9	0.19	1 0 0 0	1 0 0 0	0 1 0 0
5	0.6	-0.35	0 0 0 1	0 0 0 1	0 0 0 1
6	0.75	0.7	0 0 1 0	0 0 1 0	0 0 1 0
7	0.23	0.36	0 0 1 0	0 0 1 0	0 0 1 0
8	0.67	0.54	0 0 1 0	0 0 1 0	0 0 1 0
9	0.23	0.5	0 0 1 0	0 0 1 0	0 0 1 0
10	0.31	-0.09	0 0 0 1	0 0 0 1	0 0 0 1
11	-0.21	-0.1	1 0 0 0	1 0 0 0	1 0 0 0
12	-0.5	0.62	0 1 0 0	1 0 0 0	0 1 0 0
13	-0.49	0.52	0 1 0 0	1 0 0 0	0 1 0 0
14	0.61	0	0 0 1 0	0 0 1 0	0 0 0 1
15	-1	0.98	0 1 0 0	1 0 0 0	0 1 0 0
16	0.07	0.72	0 1 0 0	0 0 1 0	0 0 1 0
17	-0.72	-0.32	1 0 0 0	1 0 0 0	1 0 0 0
18	0.29	-0.34	0 0 0 1	0 0 0 1	0 0 0 1
19	-0.61	0.41	0 1 0 0	1 0 0 0	0 1 0 0



(그림 8) 학습 횟수에 따른 목적 패턴 분포도

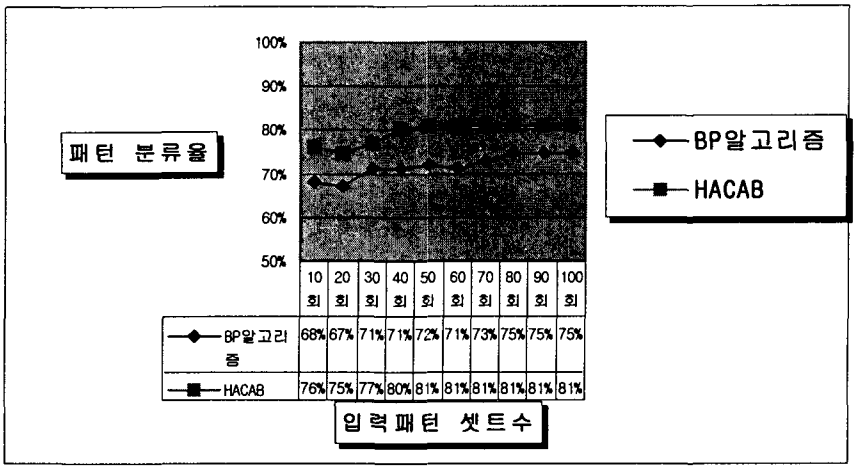
(그림 8)을 보면, 10회의 반복 학습을 한 두 번째 목적 패턴 분포도가 오히려 한 번도 학습을 하지 않은 것보다 잘 분류되지 않은 것을 볼 수 있다. 그 이유는 에러가 더 이상 줄지 않는 시점인 학습이 완료된 후에 반복 학습을 멈추어야 되는데, 학습

도중에 측정을 하였기 때문이다. 본 실험에서 10회의 경우 에러가 3.265, 100회에 2.513으로 100회 이상 학습을 시켜도 거의 에러가 거의 줄지 않았다. 또한 100회의 목적 패턴 분류율이 0회나 10회보다 뛰어남을 알 수 있었다. 따라서 본 실험에서는 경쟁 학습 모델 부분의 학습 횟수를 100회 반복한 후에 목적 패턴을 생성하였다. 생성된 목적 패턴으로 임의의 입력 패턴을 HACAB에 의하여 반복 학습을 시켰을 때, 20개의 입력 중 15개의 에러값이 0에서 1사이의 값을 가졌고, 출력도 목적 패턴에 가깝게 수렴하여 패턴 분류율이 75%로 나타났다.

여기에서 (그림 8)의 세 번째 분포도를 보자. 점들의 묶음을 각각 1사분면, 2사분면, 3사분면, 4사분면으로 나눈다면, 1사분면을 (0 0 1 0)으로, 2사분면을 (0 1 0 0)로, 3사분면을 (1 0 0 0)으로, 4사분면을 (0 0 0 1)으로 놓고 <표 3>의 학습횟수 100번일때의 목적패턴과 비교를 해 본다면, 두 점을 제외한 나머지는 일치함을 할 수 있다. 그 두 점은 x축에 걸린 2개의 점인 "1"과 "4"이다. 현재는 알고리즘에 의해서 현재는 "1"과 "4"가 각각 1의 영역과 4의 영역에 위치하지만, 만약 분포도의 x축에 걸린 두점 "1"과 "4"를 각각 "2"와 "3"의 영역으로 놓는다면, 이것은 인위적으로 목적 패턴을 생성하게 되는 것임으로 오버피팅을 초래한다. 그 결과 실제로 실험에는 패턴분류율이 68%로써 75%에 못미치는 것을 알 수 있었다.

실험의 일반성을 위해서 앞에서 설명한 임의의 입력 패턴을 HACAB과 BP알고리즘에 적용하여 총 100회의 실험을 하였다. 그 결과 (그림 9)와 같은 패턴 분류율을 기록하였다. 단, 여기에서 BP알고리즘의 목적 패턴은 앞에서 언급된 두 개의 점을 "2"와 "3"의 영역으로 놓고 목적 패턴을 인위적으로 만들었다. 결과적으로 인위적인 목적 패턴은 학습에 의해 생성된 목적 패턴과 2개가 다르게 된다.

(그림 9)의 패턴 분류율을 살펴보면, HACAB이 각 실험 횟수에 대하여 BP알고리즘 보다 전체적으로 패턴 분류율이 높은 것을 알 수 있다. 이는 HACAB의 경우에는 경쟁 학습 모델을 사용하여 목적 패턴을 생성함으로써 오버피팅을 줄일 수 있었기 때문이다.



(그림 9) HACAB과 BP알고리즘의 패턴 분류율

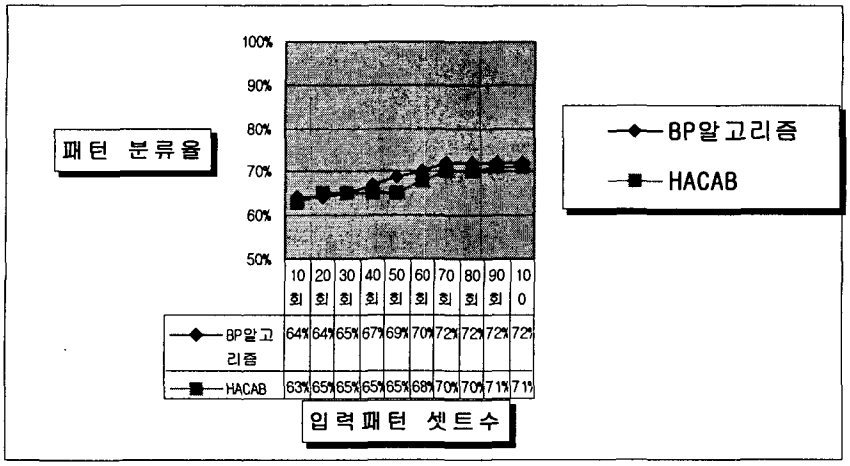
<표 4> 아이리스 데이터(Iris Data)

0.5	0.33	0.14	0.02	1	0.59	0.32	0.48	0.18	2	0.63	0.28	0.51	0.15	3
0.49	0.36	0.14	0.01	1	0.64	0.32	0.45	0.15	2	0.65	0.3	0.52	0.2	3
0.51	0.38	0.19	0.04	1	0.55	0.24	0.38	0.11	2	0.68	0.32	0.59	0.23	3
0.51	0.35	0.14	0.02	1	0.5	0.23	0.33	0.1	2	0.77	0.38	0.67	0.22	3
0.46	0.32	0.14	0.02	1	0.57	0.29	0.42	0.13	2	0.49	0.25	0.45	0.17	3
0.5	0.36	0.14	0.02	1	0.49	0.24	0.33	0.1	2	0.67	0.33	0.57	0.21	3
0.49	0.31	0.15	0.02	1	0.66	0.29	0.46	0.13	2	0.77	0.28	0.67	0.2	3
0.44	0.29	0.14	0.02	1	0.56	0.3	0.41	0.13	2	0.61	0.3	0.49	0.18	3
0.47	0.32	0.13	0.02	1	0.51	0.25	0.3	0.11	2	0.61	0.26	0.56	0.14	3
0.49	0.31	0.15	0.01	1	0.61	0.29	0.47	0.14	2	0.71	0.3	0.59	0.21	3
0.54	0.39	0.13	0.04	1	0.55	0.23	0.4	0.13	2	0.74	0.28	0.61	0.19	3
0.45	0.23	0.13	0.03	1	0.56	0.25	0.39	0.11	2	0.56	0.28	0.49	0.2	3
0.51	0.37	0.15	0.04	1	0.63	0.33	0.47	0.16	2	0.68	0.3	0.55	0.21	3
0.46	0.34	0.14	0.03	1	0.61	0.28	0.4	0.13	2	0.63	0.33	0.6	0.25	3
0.46	0.36	0.1	0.02	1	0.54	0.3	0.45	0.15	2	0.64	0.28	0.56	0.22	3
0.51	0.33	0.17	0.05	1	0.67	0.31	0.44	0.14	2	0.63	0.25	0.5	0.19	3
0.55	0.35	0.13	0.02	1	0.56	0.3	0.45	0.15	2	0.63	0.27	0.49	0.18	3
0.5	0.35	0.16	0.06	1	0.6	0.29	0.45	0.15	2	0.72	0.32	0.6	0.18	3
0.48	0.34	0.19	0.02	1	0.56	0.27	0.42	0.13	2	0.64	0.28	0.56	0.21	3
0.46	0.31	0.15	0.02	1	0.52	0.27	0.39	0.14	2	0.77	0.3	0.61	0.23	3
0.51	0.35	0.14	0.03	1	0.5	0.2	0.35	0.1	2	0.72	0.3	0.58	0.16	3
0.52	0.35	0.15	0.02	1	0.59	0.3	0.42	0.15	2	0.64	0.31	0.55	0.18	3
0.53	0.37	0.15	0.02	1	0.6	0.22	0.4	0.1	2	0.77	0.26	0.69	0.23	3
0.48	0.31	0.16	0.02	1	0.67	0.31	0.47	0.15	2	0.57	0.25	0.5	0.2	3
0.47	0.32	0.16	0.02	1	0.63	0.25	0.49	0.15	2	0.69	0.31	0.51	0.23	3
0.48	0.3	0.14	0.03	1	0.57	0.28	0.41	0.13	2	0.65	0.3	0.55	0.18	3
0.5	0.3	0.16	0.02	1	0.56	0.29	0.36	0.13	2	0.67	0.33	0.57	0.25	3
0.43	0.3	0.11	0.01	1	0.55	0.25	0.4	0.13	2	0.67	0.3	0.52	0.23	3
0.54	0.34	0.15	0.04	1	0.66	0.3	0.44	0.14	2	0.64	0.32	0.53	0.23	3
0.51	0.34	0.15	0.02	1	0.65	0.28	0.46	0.15	2	0.79	0.38	0.64	0.2	3
0.48	0.34	0.16	0.02	1	0.61	0.3	0.46	0.14	2	0.62	0.28	0.48	0.18	3
0.57	0.38	0.17	0.03	1	0.57	0.28	0.45	0.13	2	0.63	0.34	0.56	0.24	3
0.52	0.34	0.14	0.02	1	0.58	0.26	0.4	0.12	2	0.6	0.3	0.48	0.18	3
0.44	0.32	0.13	0.02	1	0.58	0.27	0.41	0.1	2	0.6	0.22	0.5	0.15	3
0.44	0.3	0.13	0.02	1	0.57	0.26	0.35	0.1	2	0.69	0.32	0.57	0.23	3
0.51	0.38	0.16	0.02	1	0.57	0.3	0.42	0.12	2	0.73	0.29	0.63	0.18	3
0.5	0.32	0.12	0.02	1	0.6	0.34	0.45	0.16	2	0.65	0.3	0.58	0.22	3
0.58	0.4	0.12	0.02	1	0.55	0.24	0.37	0.1	2	0.72	0.36	0.61	0.25	3
0.49	0.3	0.14	0.02	1	0.63	0.23	0.44	0.13	2	0.58	0.28	0.51	0.24	3
0.5	0.34	0.16	0.04	1	0.61	0.28	0.47	0.12	2	0.67	0.31	0.56	0.24	3
0.57	0.44	0.15	0.04	1	0.69	0.31	0.49	0.15	2	0.58	0.27	0.51	0.19	3
0.52	0.41	0.15	0.01	1	0.68	0.28	0.48	0.14	2	0.62	0.34	0.54	0.23	3
0.55	0.42	0.14	0.02	1	0.62	0.22	0.45	0.15	2	0.76	0.3	0.66	0.21	3
0.54	0.39	0.17	0.04	1	0.6	0.27	0.51	0.16	2	0.59	0.3	0.51	0.18	3
0.5	0.34	0.15	0.02	1	0.7	0.32	0.47	0.14	2	0.58	0.27	0.51	0.19	3
0.5	0.35	0.13	0.03	1	0.55	0.26	0.44	0.12	2	0.63	0.29	0.56	0.18	3
0.54	0.37	0.15	0.02	1	0.58	0.27	0.39	0.12	2	0.67	0.25	0.58	0.18	3
0.48	0.3	0.14	0.01	1	0.62	0.29	0.43	0.13	2	0.69	0.31	0.54	0.21	3
0.51	0.38	0.15	0.03	1	0.64	0.29	0.43	0.13	2	0.65	0.32	0.51	0.2	3
0.54	0.34	0.17	0.02	1	0.67	0.3	0.5	0.17	2	0.64	0.27	0.53	0.19	3

4.2 아이리스 데이터

앞 절에서는 임의의 입력 패턴에 대해서 실험하였고, 본 절에서는 공인된 데이터인 아이리스 데이터를 이용하여 실험하려고 한다. 아이리스는 꽃의 이름으로서, 아이리스 데이터는 꽃의 모양을 속성값으로 갖는 데이터 집합이다<표 4>. 이 데이터는 150개의 레코드를 가지고 있고, 각 레코드는 5개의 인수를 가지고 있다. 인수의 속성은 각각 꽃받침의 길이, 꽃받침의 폭, 꽃잎 길이, 꽃잎 폭을 그리고 클래스를 의미한다.

여기에서 클래스 '1'은 'Iris-Setosa', 클래스 '2'는 'Iris-Versicolor', 클래스 '3'은 'Iris-Virginica'라고 명명한다. 즉, 아이리스 데이터의 레코드는 4개의 숫자와 1개의 클래스로 구성되어져 있다. 이 데이터 셀의 숫자 4개를 이용하여 각각의 클래스로 구분되는지를 알아볼 수 있다. 그러나 아이리스 데이터에서는 데이터 자체에서 목적 패턴을 제시한다. 따라서 이 실험에서는 목적 패턴이 존재하지 않는 것으로 간주하여 실험에



(그림 10) HACAB과 BP알고리즘의 패턴 분류율

임한다. 결과를 예측해 본다면, HACAB과 BP알고리즘과의 비교하여 BP알고리즘이 완벽한 목적패턴을 가지고 학습하므로 HACAB보다 좋은 성능을 보일 수 있다. 본 실험은 목적 패턴이 없는 걸로 가정하고, 아이리스 데이터의 클래스가 3종류이므로 패턴 분류를 3가지로 하였다.

<표 4>의 아이리스 데이터에 대해서도 앞 절의 임의의 입력 패턴과 동일한 방식으로 실험하여, (그림 10)과 같은 실험결과를 얻었다. 예상했던 대로 앞 절의 임의의 입력 패턴과는 다르게, BP알고리즘이 HACAB 보다 패턴 분류율이 높은 것을 알 수 있다. 그 이유는 BP알고리즘의 목적 패턴은 아이리스 데이터의 특성상 완벽한 목적 패턴으로 학습시킨 반면에, HACAB의 목적 패턴은 알고리즘에 의해서 생성되어졌기 때문에 불완전한 목적 패턴이다.

따라서 HACAB보다 BP알고리즘에 의한 패턴 분류율이 높은 것은 당연한 것이다. 그러나 (그림 10)을 보면 90회가 넘어가면서 패턴 분류율의 차가 근접하여 HACAB도 좋은 성능을 보임을 알 수 있다.

5. 결론 및 향후 연구 방향

데이터 마이닝은 대용량의 데이터 베이스에서 의미있는 지식을 발견하기 때문에, 빠른 시간 내에 정확하고 유용한 정보를 발견하는 것이 중요하다. 이런 특성으로 인하여 오늘날 신경망 기법을 가장 일반적인 데이터 마이닝 기법의 하나로 서 사용되고 있다.

본 논문은 신경망 기법의 경쟁 학습 모델과 BP알고리즘을 결합하여 HACAB을 제안하였다. HACAB은 비감독 학습 방법의 경쟁 학습 모델에서 정규화 과정의 위험없이 목적 패턴을 도출해냈고, 감독 학습 방법의 BP알고리즘에서 목적 패턴을 인위적으로 생성하지 않아 감독 학습의 오버피팅을 예방할 수 있었다. 여기에서 HACAB은 다른 신경망의 처리 기법에 비해 두 단계를 처리하는 하이브리드형이기 때문에 처리속도(Computing Time)에 오버헤드가 발생할 수 있지만, 신경망의 큰 문제점인 정규화의 위험과 오버피팅을 방지할 수 있어서 다른 신경망 처리기법에 비해 좋은 결과를 얻을 수 있었다.

본 알고리즘은 총 100회의 임의의 데이터 집합에 대해 평균 79.4%의 패턴 분류율을 보였다. 이는 완벽한 패턴 분류를 하진 못했지만, 전체적으로 BP알고리즘의 평균 71.8%의 패턴 분류율보다 나은 분류를 보여주었는데, 이는 오버피팅을 줄일 수 있었기 때문이다. 또한 아이리스 데이터의 경우에도 만족할 만한 패턴 분류를 보여주었으며, HACAB이 목적 패턴이 존재하지 않을 경우에 효과적으로 활용할 수 있음을 알 수 있었다.

향후 연구방향으로는 국제적으로 각종 UCI데이터를 이용하여 제안된 HACAB의 일반성을 검토하고, 대용량의 데이터에 대하여 학습시간을 고려한 실험이 진행되어야 할 것이다.

참 고 문 헌

[1] 박민용, 최항식, "뉴로컴퓨터", 대영사, p301, 1991.

[2] 백준걸 외 3명, "실시간 기계 상태 데이터베이스에서 데이터 마이닝을 위한 적응형 의사결정 트리 알고리듬", Journal of the Korean Institute of Indusstrial Engineers. Vol.26, No.2, p171-182, 2000

[3] 이상원, "학습하는 기계 신경망", Ohm사, p.412, 1995

[4] 이재문, "대화형 환경에서 효율적인 연관 규칙 알고리즘", 정보처리학회논문지 D 제8-D권 제4호, p339-346, 2001.

[5] 이재필, 조경달, 김기태, "사례기반 추론을 위한 적응 지식의 자동 학습", 한국정보처리학회지 제 6권 제1호, p96-106, 1999.

[6] 장남식, 홍성완, 장재호, "데이터 마이닝", 대청, p202, 1999

[7] 장수현, 윤병주, "유전자알고리즘을 이용한 탐색공간분할 학습방법에 의한 규칙 생성", 한국저보처리학회 논문지 제5권 제11호, p2897-2907, 1998.

[8] 황석해, 문태수, 이준한, "데이터마이닝 기법을 이용한 효과적인 연구관리에 관한 연구", 추계공동학술대회 논문집, p241-252, 1999

[9] Jiawei Han, Micheline Kamber, "Data Mining:Concepts and Techniques," MORGAN KAUFMANN, p550, 2001.

[10] Tian Zhang, Raghu Ramakrishnan, and Miron, "Birch : an efficient data clustering method for very large database," the ACM SIGMOD Conference on Management of Data, Montreal, Canada, 1996.

[11] Tian Zhang, Raghu Ramakrishnan, and Riron, "Data Mining and Knowledge Discovery," p141-182, 1997.

[12] Tom M.Mitchell, MC Graw Hill, "MACHINE LEARNING," p414, 1997.

[13] http://im.inha.ac.kr/~jjeong/research.html

■ 저자소개

강문식

현재 (주)휴먼테크의 연구원으로 재직 중이다. 공주대학교 전자계산학과에서 학사(2000년), 석사(2002년)학위를 취득하였다. 주요 관심 분야는 에이전트 시스템, 신경망 알고리즘, 인공 지능 등이다.

이상용

현재 공주대학교 정보통신공학부 교수로 재직 중이다. 중앙대학교 전자계산학과에서 학사(1984년), 일본 동경공업대학대학원 시스템과학 전공에서 석사(1988년), 중앙대학교 전자계산학과에서 박사(1993년) 학위를 취득하였다. 일본 NEC 중앙연구소에서 연구원(1988년-1989년), 미국 University of Central Florida에서 visiting scholar(1996년-1997년)로 근무하였다. 주요 관심 분야는 에이전트 시스템, 기계 학습, 바이오인포메틱스 등이다.