論文2001-38CI-6-2

영역이론정련을 위한 지식기반신경망의 확장

(Extensions of Knowledge-Based Artificial Neural Networks for the Theory Refinements)

沈 東 熙 *

(Dong-Hee Shim)

요 약

분석적학습과 귀납적학습을 결합한 지식기반신경망은 다른 기계학습모델보다 우수한 성능을 나타내고 있다. 그러나 지식기반신경망에서는 신경망이 형성된 후 그 구조를 동적으로 변경할 수 없어서 영역이론정련화 기능을 제공하지 못한다. 이러한 단점을 갖고 있는 지식기반신경망을 보완하기 위하여 TopGen 알고리즘이 제안되었지만 부분적인 문제점을 안고 있다. 본 논문에서는 TopGen의 문제점을 해소하면서 지식기반신경망을 확장하여 영역이론정련기능을 부여하는 방안 2가지를 제시하고 이를 평가하였다.

Abstract

KBANN (knowledge-based artificial neural network) combining the analytical learning and the inductive learning has been shown to be more effective than other machine learning models. However KBANN doesn't have the theory refinement ability because the topology of network can't be altered dynamically. Although TopGen was proposed to extend the ability of KABNN in this respect, it also had some defects. The algorithms which could solve this TopGen's defects, enabling the refinement of theory, by extending KBANN, are designed.

Ⅰ. 서 론

영역이론(Domain Theory)이란 어떤 문제영역에 적용될 수 있는 지식을 의미하는데 이 영역이론은 완전성(Completeness), 일관성(Consistency), 정확성(Correctness)을 갖추고 있어야 한다[1]. 그러나 실세계에서 영역이론을 이와 같은 3가지 측면에서 모두 갖춘 경우는 드물기 때문에 새로운 기계학습방법에서는 불완전한 영역이론을 이용하여 예제를 해결하면서 추가적인 지식을 획득하여 영역이론을 갖추어 나간다. 불완전한 영역이론을 이와같이 이용하는 것은 분석적 학습에 해당하며, 한편 예제를 해결하면서 추가적인 지식을 획득하는 것은 귀납적 학습에 해당하기 때문에 이 새로운 방법을 통합적 학습(Hybrid Learning)이라 한다[1,2]. 그리고 이와 같이 영역이론을 점진적으로 확대하는 것을 영역이론정련화(Theory Refinements)라고 한다[3,4,5,6]. 한편 지식기반신경망(Knowledge-based Artificial Neural Network)은 어떤 문제영역에 대한 이론이 명제논리를 이용한 혼절(Horn-clause) 형태의 규칙집합으로 표현되어 있으면 이를 신경망으로 변환한다[7,8]. 이 규칙들을 신경망으로 변환한 후 예제에 의거하여 역전파 알고리즘[9]을 이용하여 신경망을 학습시킨다. 그

* 平生會員, 全州大學校 情報技術 컴퓨터工學部
(Jeonju University, School of Information Technology and Computer Engineering)

接受日字:2000年11月13日, 수정완료일:2001年5月28日

런데 문제영역에 대한 규칙을 이용한 것은 분석적 학습에 해당하며 예제를 이용한 것은 귀납적학습에 해당하기 때문에 지식기반신경망은 기계학습에서 결합적 방법으로 간주된다. 그리고 문제영역에 대한 규칙은 바로 기호적 요소에 해당하며, 신경망이 수치적 요소에 해당하기 때문에 지식기반신경망을 기호적 방법과 수치적 방법의 결합이라고도 한다. 그러나 지식기반신경망에서는 규칙들을 신경망으로 변환한 후, 신경망 구조를 변경시킬 수 없다. 만약 지식기반신경망에서 다루는 최초의 규칙집합이 완벽하지 않은 경우 규칙집합이 갖고 있는 오류의 수정을 위하여 신경망 구조는 변경될 수 있어야 한다. 이 신경망 구조의 변경이 바로 영역이론정련화에 해당하는 것이다. 본 논문에서는 지식기반신경망이 신경망구조 변경방법을 통하여 이론정련화 능력을 갖도록 하는 알고리즘을 두 가지를 제시하고자 한다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 지식기반신경망을 살펴보고 여기에 영역이론능력을 부여하여 확장한 TopGen[3,6]에 대하여 특징을 살펴보았다. Ⅲ장에서는 기존의 영역이론정련방법[4]을 간단하게 소개하였다. Ⅳ장에서는 본 연구에서 제안하는 알고리즘 THRE-KBANN (THeory REfinement for KBANN)을 기술하였으며, THRE-KBANN을 변형한 알고리즘 TR-KBANN (Theory-Refinement KBANN)을 제시하였다. 그리고 Ⅴ장에서는 이 두개의 알고리즘을 다른 알고리즘과 비교하여 성능평가를 하였으며 Ⅵ장에는 결론을 기술하였다.

Ⅱ. 지식기반신경망과 TopGen

1. 지식기반신경망 알고리즘과 그 한계

지식기반신경망은 규칙기반추론과 신경망을 결합한 접근방법으로서 3개의 알고리즘에 의하여 형성되는데, (그림1)에 나타낸 바와 같은 처리절차를 거친다[7,8]. 먼저 사용자는 문제영역에 대한 초기영역이론과 훈련예제를 제공해야 한다. 여기서 초기영역이론이란 명제논리를 이용하여 표현된 규칙들로서 문제영역에 대하여 완벽하지 않을 수도 있다. 규칙-신경망 변환알고리즘이 이 초기영역이론을 신경망으로 변환하며, 신경망의 노드들의 위상과 연결가중치를 결정한다.

다음에는 역전파 알고리즘[9]을 이용하여 훈련예제로서 신경망을 훈련시킨다. 이 훈련과정에서 신경망의 위상은 변하지 않고 연결가중치를 변경시키게 된다. 마지막으로 훈련된 지식기반신경망에서 규칙을 도출할 수 있다.

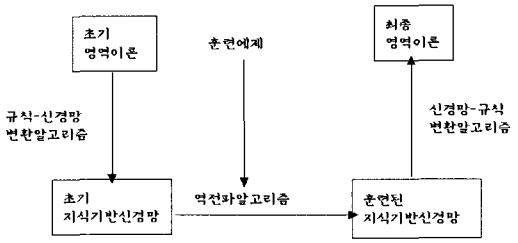


그림 1. 지식기반신경망에서의 처리절차
Fig. 1. Process in Knowledge-based Artificial Neural Network.

그러나 지식기반신경망에서는 규칙-신경망변환알고리즘에 의하여 지식기반신경망이 일단 수립되면 가중치나 바이어스의 변경만 가능하고 노드의 추가나 삭제는 이루어 질 수 없는 단점을 갖고 있다. 즉 영역이론정련화의 능력이 거의 없다고 볼 수 있다.

2. TopGen

지식기반신경망의 단점인 영역이론정련화기능 부족을 보완할 수 있도록 지식기반신경망을 확장시키는 연구도 이루어졌다[3,6]. 첫째 TopGen 알고리즘[3,6]에서는 훈련된 지식기반신경망을 이용하여 노드의 오류율에 근거하여 노드의 추가위치를 N개 선정한다. 다음에는 빔탐색을 이용하여 신경망을 훈련시킨 후 가장 오류율이 적은 신경망을 도출한다. 그러나 이 알고리즘에서는 오류의 종류 즉 부(False)를 진(True)으로 분류, 진을 부로 분류 등에 관계없이 처리하였고, 추가노드를 모두 입력노드에 연결시켰으며, 빔탐색을 함으로써 지나치게 방대한 탐색이 소요된다는 단점을 안고 있다. TopGen에 의하여 도출된 신경망은 훈련된 지식기반신경망을 다시 정련화시킨 결과이기 때문에 원래의 KBANN보다 좋은 성능을 보이고 있다.

본 논문에서는 이와 같은 TopGen 알고리즘보다 효율적이 될 수 있도록 KBANN을 확장하여 추가노드의 삽입 방법을 제안하였으며, 탐색시 언덕오르기를 채택하여 빔탐색으로 인한 복잡성부담을 완화시켰다.

Ⅲ. 영역이론정련방법

1. 영역이론의 오류

예제는 진제와 부제로 분류될 수 있는 데 진제란 양의 예제(진을 진으로 분류) 및 음의 예제(부를 부로 분류)를 가르키는 것으로 신경망에 의하여 옳바르게 분류된 예제를 말한다[3,4,6]. 부제란 진제가 아닌 것을 의미하는 데 양의 예제가 음으로, 음의 예제가 양으로 분류된 경우이다. 이러한 부제는 다시 양부제(false positive)와 음부제(false negative)로 나뉠 수 있는 데, 양부제는 진이 아닌 것이 진으로 분류된 경우이며, 음부제는 진이 부로 분류된 경우를 의미한다.

한편 명제논리 혼절(Propositional Horn-clause)을 이용하여 표현된 영역이론에서의 오류는 <표1>에 나타낸 바와 같이 6종류가 있다[4].

표 1. 영역이론에서의 오류 종류

Table 1. Type of errors in domain theory.

구분	오류종류
과잉구체화(Overly Specific)	추가조건(Additional Antecedent)
	누락규칙 (Missing Rule)
	도치조건 (Inverted Antecedent)
과잉일반화(Overly General)	누락조건 (Missing Antecedent)
	추가규칙 (Additional Rule)
	도치조건 (Inverted Antecedent)

먼저 과잉구체화란 범주에 해당하는 예제를 부로 분류하는 경우로서 음부제에 해당한다. 이는 규칙의 조건이 추가됨으로 인하여 결론에 대한 and 조건이 강화되었을 때 발생할 수 있고, 또 정당한 규칙이 누락되어 or 조건이 부족하게 되었을 때 발생할 수 있고 또는 조건의 not이 잘못 표기되었을 때 발생할 수 있다. 과잉일반화란 과잉구체화의 반대로서 범주에 해당하지 않는 예제를 진으로 분류하는 경우로서 양부제에 해당한다. 이는 규칙의 조건이 누락됨으로 인하여 결론에 대한 and 조건이 완화되었을 때 발생할 수 있고, 또 잘못된 규칙이 추가되어 or 조건이 완화되었을 때 발생할 수 있고 또는 조건의 not이 잘못 표기되었을 때 발생할 수 있다.

2. 오류의 처리방법

과잉구체화나 과잉일반화를 수정하는 방법은 <표2>에 나타낸 바와 같다.

표 2. 오류의 수정방법

Table 2. Methods of error correction.

현재 노드	음부제(과잉구체화) 해결시	양부제(과잉일반화) 해결시
OR 노드 A = B v C	A = B v C v N	A = B v C, N2 = A ^ N1
AND 노드 A = B ^ C	A = B ^ C, N2 = A v N1	A = B ^ C ^ N

현재 노드가 OR노드인 경우 음부제를 해결하기 위해서는 새로운 노드 N을 만들어 A에 OR로 연결하며, 양부제 해결시에는 새로운 노드 N1과 N2를 만들어 A와 N1을 N2에 AND로 연결시킨다. AND노드에서 음부제를 해결하기 위해서는 새로운 노드 N1과 N2를 만들어 A와 N1을 N2에 OR로 연결시키며, 양부제 해결시에는 새로운 노드 N을 만들어 A에 AND로 연결시킨다. 음부제의 해결은 과잉구체화를 해소시키는 것이며, 양부제의 해결은 과잉일반화를 해소시키는 것이다. 이 방법은 과잉구체화와 과잉일반화를 그 원인별로까지는 다루지 못하지만 전체적인 해결방법이 되는 것이다.

Ⅳ. KBANN의 확장 알고리즘

1. THRE-KBANN

THRE-KBANN(THeory REfinement for KBANN) 알고리즘은 KBANN에 적용할 수 있는 영역이론정련화를 위한 알고리즘으로서 아래에 나타낸 바와 같다. 이는 3장에서 설명한 6가지 종류의 영역이론 오류에 대처할 수 있는 것이다.

① 훈련사례를 시험집합, 조정집합1, 조정집합2로 임의로 분류한다.

② 시험집합을 이용하여 훈련된 KBANN을 생성한다.

③ 조정집합1을 이용하되 다음의 절차에 의하여 신경망을 생성한다.

ⓐ 각 노드의 음부제와 양부제의 값을 0으로 초기화한다.

ⓑ 조정집합1에서 각 부제에 대하여 각 노드에서의 양부제, 음부제 여부를 판단하여 해당값을 증가시킨다.

ⓒ 양부제와 음부제의 합이 가장 큰 노드를 선정한다. 같은 경우는 양부제나 음부제의 비율이 편중된 노

드, 입력계층에 가까운 노드 순으로 선정한다.

ⓓ 노드 추가방법에 의거하여 노드를 추가하여 신경망을 생성한다.

④ 새로운 신경망을 조정집합2를 이용하여 훈련시키고 오류율이 이전의 신경망보다 높으면 단계ⓒ로 되돌아간다.

⑤ 새로운 신경망의 오류율이 중지조건을 만족하면 이를 출력하고 만족하지 않으면 단계③으로 간다.

(1) 노드 추가 위치

Ⅲ장에서 설명한 양부제 및 음부제의 개념은 출력노드의 입장에서 정의된 것이다. 그런데 위 알고리즘에서 사용된 양부제 및 음부제는 신경망을 구성하는 각 노드 측면에서 정의되어야 한다. 즉 부제에 대하여 어떤 노드가 양인 경우 음으로 바꾸어 진제가 되면 양부제(false positive)라 하고, 노드가 음인 경우 양으로 바꾸어 진제가 되면 음부제(false negative)라고 볼 수 있다. 지식기반신경망에서 활성값(activation value)은 1에 가깝거나 0에 가깝다. 음인 경우는 활성값이 0에 가깝고 양인 경우는 활성값이 1에 가까운 것이다. 이것은 Towell이 실험적으로 입증하여 주장한 "지식기반신경망에서 각 노드는 완전활성화(fully active)되거나 완전불활성화(fully inactive)된다"[7] 성질에 근거한다. 노드의 추가 위치를 결정하기 위하여 부제에 대하여 각 노드는 양부제수, 음부제수에 대한 자료를 유지한다. 그리하여 노드의 추가위치는 양부제수와 음부제수의 합이 큰 노드를 선택하고, 같은 경우는 양부제나 음부제의 비율이 편중된 노드를 선정한다.

(2) 노드의 추가방법

새로운 노드의 추가위치가 결정되면 노드는 <표2>에 나타낸 바와 같이 추가가 되는 데 이때의 Bias, 링크가중치 등은 <표3>과 같이 한다.

1) 새로운 노드로의 연결노드와 링크 가중치

새로운 노드는 하위계층의 노드들로부터 링크를 갖도록 한다. 이때 하위계층의 노드들은 다음과 같이 결정한다. 먼저 OR노드에서 음부제 해결시와 AND 노드에서 양부제 해결시에는 B,C의 다음 하위계층에 연결하되 만일 B,C가 입력계층이면 입력계층과 연결한다. 한편 AND노드에서 음부제 해결시와 OR 노드에서 양부제 해결시에는 B, C와 같은 계층의 다른 노드들과 연결하되 B, C와 같은 계층의 다른 노드가 없으면 다음 하위계층의 노드들과 연결한다.

이때 링크 가중치는 노드의 추가위치를 결정하기 위하여 수집한 음부제, 양부제의 자료에 의거하여 음부제와 양부제의 비율에 의거하여 결정한다. 그리하여 양부제나 음부제의 비율이 유사하면 +1 다르면 -1의 가중치를 갖도록 한다. 즉 노드가 추가되는 위치가 양부제의 비율이 높았다면 하위계층의 노드중 양부제의 비율이 높은 노드로부터의 링크 가중치는 W로 하며, 하위계층의 노드중 양부제의 비율이 낮은 노드로부터의 링크 가중치는 -W로 하며, 양부제의 비율이 음부제와 대등한 노드로부터의 링크 가중치는 0으로 한다. 이와 반대로 노드가 추가되는 위치가 음부제의 비율이 높았다면 하위계층의 노드중 음부제의 비율이 높은 노드로부터의 링크 가중치는 W로 하며, 하위계층의 노드중 음부제의 비율이 낮은 노드로부터의 링크 가중치는 -W로 하며, 음부제의 비율이 양부제와 대등한 노드로부터의 링크 가중치는 0으로 한다. 한편 노드가 추가되는 위치가 양부제의 비율과 음부제의 비율이 유사한 경우는 하위계층의 노드중 양부제의 비율이 높은 노드로부터의 링크 가중치는 -W로 하며, 하위계층의 노드중 양부제의 비율이 낮은 노드로부터의 링크 가중치는 -W로 하며, 양부제의 비율이 음부제와 대등한 노드로부터의 링크 가중치는 W로 한다.

하위계층의 노드중 양부제나 음부제가 발생하지 않

표 3. 노드 추가시 링크가중치와 Bias

Table 3. Weight and bias in node addition W : 지식기반신경망에서 링크의 기본가중치.

현재 노드	음부제 해결시	양부제 해결시
OR 노드 A = B ∨ C A의 Bias=-W/2	A = B ∨ C ∨ N A의 Bias=-W/2 N-A 링크가중치=W	A = B ∨ C, N2 = A ∧ N1 A의 Bias=-W/2 A-N2 링크가중치=W N1-N2 링크가중치=W N2의 Bias=-3W/2
	-B,C의 다음 하위계층에 연결 -B,C가 입력계층이면 입력계층과 연결	-B,C와 같은 계층의 다른 노드들과 연결 -다른 노드가 없으면 다음 하위 계층에 연결
AND 노드 A = B ∧ C A의 Bias=-3W/2	A = B ∧ C , N2 = A ∨ N1 A의 Bias=-3W/2 A-N2 링크가중치=W N1-N2 링크가중치=W N2의 Bias=-W/2	A = B ∧ C ∧ N A의 Bias=-5W/2 N-A 링크가중치=W
	-B,C와 같은 계층의 다른 노드들과 연결 -다른 노드가 없으면 다음 하위 계층에 연결	-B,C의 다음 하위계층에 연결 -B,C가 입력계층이면 입력계층과 연결

는 경우는 링크가중치를 0으로 한다.

2) 새로운 노드의 Bias

OR노드에서 음부제 해결시와 AND 노드에서 양부제 해결시 노드 N의 Bias 및 AND노드에서 음부제 해결시와 OR 노드에서 양부제 해결시 노드 N1의 Bias 모두 -(2N-1)W/2 (N는 하위계층 노드로부터의 링크수)로 한다. 또한 OR노드에서 음부제 해결시 A의 Bias는 -W/2로, 양부제 해결시 N2의 Bias는 -3W/2로 , AND 노드에서 음부제 해결시 노드 N2의 Bias는 -W/2로, 양부제 해결시 A의 Bias는 -5W/2로 한다.

3) 새로운 노드로부터 상위계층 노드로의 링크 가중치

새로운 노드로부터 상위계층 노드로의 링크 가중치는 본래의 노드가 갖고 있던 가중치를 따르도록 한다.

2. TR-KBANN

THRE-KBANN은 추가되는 노드를 다음 하위계층의 노드에 연결시킴으로써 하위계층노드들이 보유하고 있는 현재까지의 학습내용을 유지시킨다는 장점이 있지만 오류발생노드들간의 종속적 효과로 인한 단점을 보유하고 있다. 그래서 다른 대안은 새로운 노드를 하위계층의 노드와 연결하지 않고 입력노드와 연결하는 방안 TR-KBANN(Theory Refinement algorithm for KBANN)을 고려할 수 있다. 이 TR-KBANN도 역시앞에서 설명한 THRE-KBANN과 같은 절차를 따른다. 다만 새로운 노드는 THRE-KBANN과 같이 하위계층의 노드와 연결하지 않고 입력노드와 연결을 하되 가중치를 다음과 같이 설정한다.

새로운 노드는 입력노드들중 관련이 깊은 노드들로부터 0 이 아닌 가중치를 갖도록 한다. 이러한 노드를 결정하기 위하여 다음과 같은 휴리스틱 방법을 이용한다. 여기서 사용하는 기호는 다음과 같다. 먼저 입력노드는 입력특성이 있을 수 있고(값이 1) 없을 수 있는데(값이 0) 이를 각각 P(present)와 A(absence)로 나타낸다. 추가노드에 대해서는 추가노드가 진 또는 부(활성값이 1에 가까운지 0에 가까운지에 따라 결정됨)일 수 있는 데 이를 각각 T(true)와 F(false)로 나타낸다. 또한 추가노드의 값이 옳을 수 있고 틀릴 수 있는 데 이를 각각 C(correct)와 I(incorrect)로 나타낸다. 이 C와 I에 대한 결정은 출력노드부터 입력노드 방향으로 결정되어 온다. 조건 -> 결론과 같은 규칙에서 만약 결론이 맞으면 조건도 맞은 것으로 간주하며 결론이 틀린 경우 조건을 바꾸어 결론이 맞게 되는 경우 조건이 틀린 것으로 간주한다. 그리고 P(C|T)는 T인 가정아래 C일 조건부확률을 표기하기로 하고 다음과 같이 처리한다.

앞절에서 설정된 추가노드에 대하여(표1에서 노드 N 또는 노드 N1에 해당)

① 각 입력노드별로 8개의 자료를 유지한다.

-입력특성의 존재여부에 따른 P 또는 A로의 분류, 추가노드의 활성화여부에 따른 T 또는 F로의 분류, 추가노드 값이 옳았을 경우는 C(Correct), 틀렸을 경우는 I(Incorrect)로 분류,

위의 3가지 경우를 조합하여 8가지 자료를 유지한다.

② 조정집합2의 각 예제에 대하여 다음을 처리한다.

ⓐ 예제에 대하여 추가노드의 활성화 여부 T,F를 판정하며, 옳고 그름 C,I를 판단한다.

ⓑ 각 입력노드별로 다음을 처리한다.

㉠ 입력노드의 입력특성여부 P, A를 판단한다.

- P,A 여부, T,F 여부, C,I여부에 따라 8가지 경우중 하나에 해당한다.

㉡ 입력노드의 8가지 자료중 해당자료에 경우수 1을 누적시킨다.

③ 각 입력노드별로 위에서 산출한 경우의 수에 의거하여 다음을 계산한다.

ⓐ on= P(C|T) * P(P|C & T)

P(C|T)는 추가노드가 활성화되었다는 가정 아래 이 활성화가 올바를 확률인 데 이런 경우는 확률이 1까지 되어 가중치가 1로 될 수 있다. 그러나 이 확률은 입력특성여부에 관계가 없기 때문에 입력노드별로 모두 동일하다. 따라서 입력노드별로 다른 값을 갖도록 조정하는 것이 필요하다. 그런데 입력특성의 존재가 추가노드에 영향을 미치기 때문에 이 확률을 계산하여야 한다. 그래서 활성화되어 있고 올바를 가정하에 입력특성이 있을 조건부확률을 곱하도록 한다. 확률값을 이용한 것이므로 이 값은 0에서 1의 값을 갖게 된다.

ⓑ off = - P(C|F) * P(P|C & F)

off의 계산은 on의 경우와는 활성/비활성 측면에서 대칭이 되어야 한다. 또한 가중치가 반대값이 되도록 부호가 바뀌어야 한다. 따라서 on의 수식에서 T를 F로 바꾸고 부호를 변경시키면 된다. 이의 해석은 다음과 같다.

P(C|F)는 비활성화되었다는 가정아래 올바를 확률인 데 이런 경우는 확률이 1까지 되어 가중치가 -1까

지 될 수 있다. 그러나 이 확률 역시 입력특성여부에 관계없이 입력노드별로 모두 동일하다. 따라서 입력노드별로 다른 값을 갖도록 조정하는 것이 필요하다. on에서와 마찬가지로 입력특성의 존재가 추가노드에 영향을 미치기 때문에 이 확률을 계산하여야 한다. 그래서 활성화되어 있으면서 틀렸을 가정하에 입력특성이 있을 조건부확률을 곱하도록 한다. 이 값 역시 확률간의 계산이므로 0에서 1의 값을 갖게 된다. 한편 이 값은 on을 계산한 수식에서 활성화 여부를 바꾼 값이 되며 부호를 바꾼 값이 된다.

ⓒ 입력노드에서 추가노드로의 링크가중치

앞에서 계산된 on 과 off 중 절대값이 큰 수를 링크가중치로 한다.

V. THRE-KBANN의 성능평가

1. 평가대상 알고리즘

본 연구에서 제시한 THRE-KBANN, TR-KBANN의 성능을 영역이론정련기능을 보유한 TopGen[3,6]과 비교하고, 또한 간단한 확장신경망[6]과 비교한다. 간단한 확장신경망이란 노드들을 하나씩 추가하고 이 노드들을 모든 입력노드 및 출력노드와 연결한 것으로 이 방안 역시 기본적인 영역이론정련능력을 보유하는 것으로 간주할 수 있다.

2. 평가대상 문제영역

유전학에서 나타나는 두가지 문제 즉 프로모터(promoter) 인식문제와 접목점(splice-junction) 결정문제를 평가에 이용하였다. 분자생물학에서 DNA는 뉴클레오티드라고 불리우는 {A, G, T, C} 문자집합에서 선택된 문자들의 선형배열이다. 이 DNA는 RNA로, RNA는 다시 프로테인(protein)으로 복제가 된다. 이 프로테인은 세포의 활성적 요소에 해당한다. 그런데 DNA에 있는 명령에 근거하여 프로테인이 생성되기 때문에 DNA에서의 오류는 잘못된 프로테인을 유발하며 선천적 질병으로 나타나게 된다. 따라서 인간의 DNA배열을 아는 것은 선천적 질병을 처리하기 위한 첫 단계가 된다. DNA배열에서는 위치를 명시하는데 특별한 표기법을 이용한다. 즉 생물학적으로 의미있는 고정된 점(기준점)을 기준으로 위치를 숫자화한다. 음수는 기준점에 앞서는 위치를 나타내고 양수는 기준점 뒤의 위치를 나타낸다. 예를 들어 @3 'AGTC" 는 기준점에서 오른쪽으로 3번째의 뉴클레오티드는 A이며 다음에는 GTC임을 의미한다.

(1) 프로모터 인식문제

DNA의 배열중 프로테인으로 복제되는 부분을 유전인자(genes)라고 한다. DNA배열의 이해를 위해서는 먼저 이 유전인자의 시작부분과 끝을 인식해야 한다. 유전인자의 끝부분은 3개의 뉴클레오티드에 의하여 표시가 되기 때문에 쉽게 인식이 되는 데 시작부분의 인식은 쉽지가 않다. 프로모터란 유전인자앞에 선행하는 짧은 DNA배열을 의미한다. 따라서 프로모터를 인식하면 유전인자의 시작부분을 알게 된다. 프로모터 DNA배열은 뉴클레오티드 57개로 구성되는데 이것이 프로모터인지 아닌지를 판단할 수 있는 일부의 규칙[5]들이 알려져 있는 데 이 규칙들이 완벽하지는 않다. 프로모터 인식문제에 대한 예제는 프로모터에 해당하는 234개의 진제와 4921개의 부제로 구성하였다.

(2) 접목점 결정(splice-junction determination) 문제

접목점(splice junction)은 프로테인 생성과정중 DNA 배열에서 불필요한 DNA가 제거되는 지점들이다. 이 DNA 배열에는 접목후에도 유지되는 부분에 해당하는 엑손(exons), 접목후에 없어지는 부분에 해당하는 인트론(introns)이 있다. 그래서 E/I 부분이라 불리우는 엑손/인트론 경계와 I/E 부분이라 불리우는 인트론/엑손 경계를 인식해야 하는 데 이러한 경계인식문제를 접목점 결정문제[5]라 한다. 이 문제에 대한 예제는 1210개의 진제와 1890개의 부제로 구성하였다.

3. 평가방법

이 평가를 위하여 Solaris 7에서 Common LISP으로 시뮬레이션을 하였다. 프로모터 인식문제와 접목점 결정문제에 대하여 규칙을 1개부터 5개까지 삭제한 경우에 대하여 THRE-KBANN과TR-KBANN을 수행하였다. 각 경우에 대하여 조정집합2의 오류율이 0 %가 되거나 오류율에 더 이상의 개선이 없을 때까지 THRE-KBANN과 TR-KBANN을 수행시켰다. 수행결과 최종 형성된 각 신경망의 오류율을 시험집합을 이용하여 측정하였으며 이러한 시험을 5회씩 수행하였다.

4. 평가결과

프로모터 인식문제와 접목점 결정문제에 THRE-KBANN과 TR-KBANN을 적용한 결과 인식오류율의 평균은 <표4>에, 추가된 노드수의 평균치는 <표5>에 나타낸 바와 같다. <표4>에서는 오류율을 비교하였는

데 본 논문에서 제안한 TR-KBANN의 프로모터 인식 문제와 접목점 결정문제에 대한 오류율은 각각 1.97%, 4.05%로서 TopGen의 오류율 2.06%, 4.17%보다 낮아 오류율 측면에서 성능이 좋음을 알 수 있으며 이는 THRE-KBANN과 비교할 때도 비슷한 수준을 나타냈다. <표5>에서는 추가된 노드수를 비교하였는 데 TR-KBANN과 THRE-KBANN 모두 추가된 노드수가 더 적게 나타났다. 인식오류율 측면에서는 TR-KBANN이 THRE-KBANN 과 비슷한 수준이지만 추가노드수 측면에서는 더 적다는 것은 학습효율이 더 좋음을 간접적으로 나타내는 것이다.

VI. 결론 및 향후과제

KBANN은 명제논리로 표현된 문제영역에 대하여 기존의 학습방법보다 더 효율적인 방법으로 입증이 된 바 있다. 그러나 실세계의 대부분의 초기지식은 근사적으로 옳바르기 때문에 영역이론정련화가 사실 필요하다. 그런데 KBANN은 영역이론정련화 능력을 보유하고 있지 않다. KBANN의 이러한 단점을 보완하기 위하여 TopGen에서는 오류의 원인이 되는 노드를 찾아 노드를 추가하되 이 노드를 모든 입력노드들과 연결시켰다. 또한 노드를 추가할 때 빔탐색을 이용하므로써 시간복잡성의 부담을 안고 있다. 본 논문에서 제안한 THRE-KBANN에서는 TopGen알고리즘을 2가지 측면에서 개선하였다. 즉 빔탐색을 역추적을 허용한 언덕오르기 탐색으로 바꾸었고, 추가노드를 모든 입력노드와 연결하는 대신 하위계층의 노드와 연결하였다. 그 결과 TopGen보다 좋은 효율성을 높였다. 또 다른 대안으로 제안한 TR-KBANN은 THRE-KBANN과 처리골격은 같지만 영역이론 정련화를 위하여 추가되는 노드에 대하여 관계가 깊은 입력노드만을 연결시켰다. 그리하여 추가된 노드를 관계가 깊은 일부의 입력노드에만 링크를 시켰으며 링크의 가중치를 노드간의 성격에 따라 차별성을 부여함으로써 그 동안의 학습효과가 반영되도록 하였다. 유전학에서 발생하는 두 문제영역에 이 알고리즘을 적용하여 본 결과 이러한 개선으로 인하여 두가지 확장방안 모두 KBANN과 TopGen보다는 인식율뿐만 아니라 추가노드수 측면에서도 효율적인 성능을 보였다.

향후 연구로는 본 연구에서 제안한 THRE-KBANN이나 TR-KBANN을 서술논리에 대해서도 적용할 수 있도록 개선할 필요가 있겠다. 그리고 이 두 가지 알고리즘을 결합하여 이용할 수 있는 방안이 강구될 필요도 있다 하겠다. 그리고 영역이론의 상위계층의 규칙이 누락되었을 경우에도 THRE-KBANN, TR-KBANN이 잘 적용될 수 있도록 보완되어야 할 것이다. 이를 위해서는 복수전략학습(multistrategy learning)과 같은 방법을 도입해야 할 것이다.

표 4. 인식오류율 결과

Table 4. Error rate comparison.

알고리즘	오류율	
	프로모터 인식문제	접목점 결정문제
TR-KBANN	1.97	4.05
THRE-KBANN	1.98	4.05
TopGen	2.06	4.17
간단한 확장신경망	2.12	4.53
KBANN	2.31	4.58

표 5. 추가 노드 수

Table 5. Total number of nodes added.

알고리즘	추가된 노드수	
	프로모터 인식문제	접목점 결정문제
TR-KBANN	4.0	3.6
THRE-KBANN	4.2	3.8
TopGen	4.4	4.0
간단한 확장신경망	5.0	5.2

참 고 문 헌

[1] S. B. Thrun and T. M. Mitchell, "Integrating Inductive Neural Network Learning and Explanation-Based Learning", In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, pp. 930~936, 1993.

[2] J. W. Shavlik and G. G. Towell, "An Approach to Combining Explanation-based and Neural Learning Algorithms", Connection Science, Vol.1, No. 3, pp. 233~255, 1989.

[3] D. W. Opitz, 'An Anytime Approach to Connectionist Theory Refinement : Refining the Topologies of Knowledge-Based Neural Networks', PhD thesis, University of Wisconsin-Madison, 1995.

[4] D. Ourston and R. Mooney, "Theory Refinement Combining Analytical and Empirical Methods", Artificial Intelligence, Vol.66, pp. 273~309, 1994.

[5] G. G. Towell, & J. W. Shavlik and M. Noordewier, "Refinement of Approximate domain theories by Knowledge-based Neural Networks", In Proc. of the 8th National Conference on Artificial Intelligence, pp. 861~866, Boston, MA, 1990.

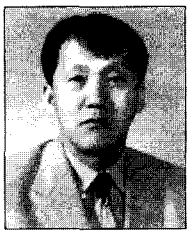
[6] D.W. Opitz, and J. W. Shavlik , "Heuristically Expanding Knowledge-Based Neural Networks", In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence,1993, pp. 1360~1365, 1993.

[7] G.G. Towell, "Symbolic Knowledge and Neural Networks : Insertion, Refinement, and Extraction", PhD thesis, University of Wisconsin-Madison, 1991.

[8] G. G. Towell, & J. W. Shavlik, "Using Symbolic Learning to Improve Knowledge-Based Neural Networks", Proceedings of AAAI, pp.177~182, 1992.

[9] D.E. Rumelhart, G.E.Hinton, and J. R. Williams, "Learning Internal Representations by Error Propagation", Vol. 1, pp. 318~363, MIT press, 1986.

저 자 소 개



沈 東 熙(平生會員)

1980년 2월 : 서울대학교 산업공학과 졸업. 1982년 2월 : 서울대학교 대학원 졸업(공학석사). 1994년 2월 : 고려대학교 대학원 전산과학과 졸업(이학박사). 1990~현재 : 전주대학교 정보기술컴퓨터공학부 교수. <주관심분야> 기계학습, 네트워크보안, 게임공학