

## 뉴로-퍼지 추론을 적용한 포석 바둑

이병두  
성결대 정보통신공학부  
blee026@korea.com

Applying Neuro-fuzzy Reasoning to Go Opening Games

Byung-Doo Lee  
Dept. of Information and Communication Engineering,  
Sungkyul University

### 요 약

본 논문은 포석 바둑을 위해, 패턴 지식을 근간으로 바둑 용어 지식을 수행할 수 있는 뉴로-퍼지 추론에 대한 실험 결과를 설명하였다. 즉, 포석 시 최선의 착점을 결정하기 위한 뉴로-퍼지 추론 시스템의 구현을 논하였다. 또한 추론 시스템의 성능을 시험하기 위하여 시차 학습 (TD( $\lambda$ ) learning) 시스템과의 대결을 벌였다. 대결 결과에 의하면 단순한 뉴로-퍼지 추론 시스템조차 시차 학습 모델과 충분히 대결할 만하며, 뉴로-퍼지 추론 시스템이 실제 바둑 게임에도 적용될 수 있는 잠재력을 보였다.

### ABSTRACT

This paper describes the result of applying neuro-fuzzy reasoning, which conducts Go term knowledge based on pattern knowledge, to the opening game of Go. We discuss the implementation of neuro-fuzzy reasoning for deciding the best next move to proceed through the opening game. We also let neuro-fuzzy reasoning play against TD( $\lambda$ ) learning to test the performance. The experimental result reveals that even the simple neuro-fuzzy reasoning model can compete against TD( $\lambda$ ) learning and it shows great potential to be applied to the real game of Go.

**Keyword** : Go, Neuro-fuzzy reasoning, TD( $\lambda$ ) learning, Go term knowledge, Pattern knowledge

## 1. 서론

바둑은 새로운 인공지능 기법을 시험하기 적합한 흥미로운 연구 영역이다[1,2]. 간단한 바둑 규칙에도 불구하고, 바둑은 꽤나 복잡한 전략·전술을 요하며 방대한 탐색 공간으로 인해 컴퓨터 바둑은 컴퓨터 체스보다 구현하기가 용이하지가 않다.

보드 게임의 문제 해결을 위한 전통적인 인공지능 기법은 유리한 게임 전개를 위해 가능한 일련의 모든 수순을 다 탐색하여 최적의 수순을 찾아낸다. 탐색 시간을 절감하기 위하여 휴리스틱(heuristic) 탐색 기법이 흔히 사용된다. 바둑은 꽤나 복잡성을 갖고, 상대편 정보를 지득 가능하며, 확률성이 아니며, 전술적 게임이 아닌 전략적 게임이기에, 이러한 휴리스틱 탐색기법은 컴퓨터 바둑에서는 적용될 수 없다. 즉, 컴퓨터 바둑은 컴퓨터 체스에서 적용되었던 휴리스틱 탐색 기법이 아닌 바둑 지식을 근간으로 한 접근법으로 풀어나가야 한다.

지난 수 세기에 걸쳐 프로 기사들에 의해 수많은 바둑 지식들이 축적되어 왔으며, 이들 대부분이 귀감적인 기보 형태로 내재되어 왔다. 바둑 기사들이 인지하고 있는 패턴들은 반상 위에 있는 단순한 돌들과 빈 공간의 형상이 아닌 그 이상의 의미를 갖고 있다. 즉, 바둑 기사들은 직감적으로 포위된 돌들의 사활을 인식해내며, 게임 중에 바둑돌 간의 영향력, 안정성 및 연결성 등을 고려한 사전 지식(priori knowledge)을 갖고 효율적인 차기 착점을 찾아낸다. 바둑에 있어 이러한 시각적인 인식력은 바둑 기사들에게는 쉽지만, 컴퓨터에서 이를 모델화하기란 꽤나 어렵다.

본 논문에서 저자는 바둑 지식을 내재적 및 외재적 바둑 지식으로 양분하였다. 내재적 바둑 지식은 패턴 지식으로 간주하였으며, 귀감적인 프로 기보에 내재되어 있는 패턴 지식을 추출해내기 위하여 패턴 인식이 가능한 기법을 본 실험에 적용하였다. 한편, 외재적 바둑 지식은 바둑속담, 격언, 용어 등을 포함하는 모든 바둑의 제언으로 간주하

였으며, 이러한 바둑 제언을 구현할 수 있는 규칙을 기반으로 한 퍼지 추론 시스템을 구현하였다.

## 2. 퍼지 추론 시스템

[그림 1]에 보여준 바와 같이 퍼지 추론 시스템은 (1) 통제 학습 신경망, (2) 후보 착점 정량화 및 (3) 뉴로-퍼지 제어기 등의 세 가지로 구성되어 있으며, 시스템 내부에서의 작업 단계는 다음과 같다.

[1 단계] : 후보 착점 생성

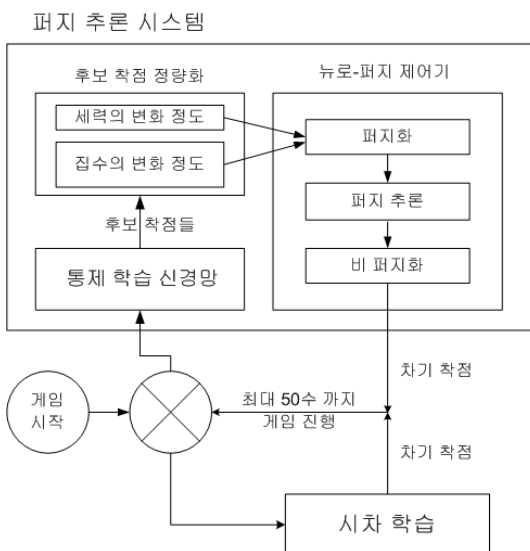
패턴 인식이 가능한 통제 학습 신경망을 통해 10개의 후보 착점들을 생성.

[2 단계] : 후보 착점 평가

각 후보 착점들은 (1) 세력 및 (2) 잠재적 집수의 변화율을 산정하여 뉴로-퍼지 제어기에 입력.

[3 단계] : 최선의 착점 선택

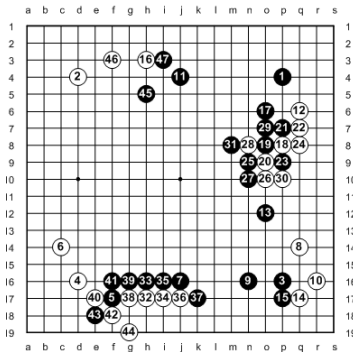
길들여진 뉴로-퍼지 제어기가 퍼지 추론 과정을 거쳐 10개의 후보 착점 중에서 최선의 착점을 선택.



[그림 1] 시차 학습과 대결을 위한 퍼지 추론 시스템

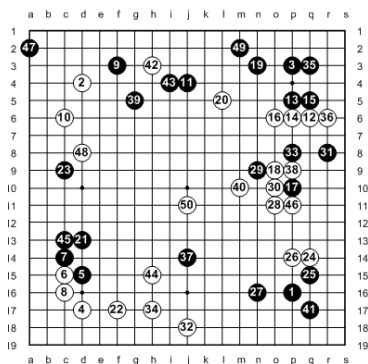
## 2.1 통제 학습 신경망

포석 시 대부분의 기사들은 반상에 놓여있는 돌들의 모양과 그들이 기억하고 있는 귀감적인 기보들을 비교하여 최선의 후보 착점을 선택한 후, 각 후보 착점 시 바둑판에 미치는 세력, 집들의 안정성 등과 같은 바둑 지식을 근간으로 각 착점의 효과/효율성을 검토한 후 최선의 착점을 결정한다. 저자는 컴퓨팅 시간을 고려하여 임의의 개수인 후보 착점 10개를 생성해내기 위해 패턴 인식이 가능한 신경망을 구현하였으며, 이 신경망은 [그림 2]와 같은 귀감적인 1,704개의 포석 기보에 의해 훈련되었다.



[그림 2] 신경망을 훈련시킨 포석

훈련된 신경망은 [그림 3]과 같은 포석을 생성해 내는 우수성을 보였으나, 간혹 흑47과 같은 엉뚱한 수를 생성해내곤 한다.



[그림 3] 신경망이 생성해 낸 포석

이는 신경망이 수순의 중요성에 대한 인식 결여 및 전략·전술을 구사하기 위한 바둑 지식의 부족에 기인되며, 이러한 결함은 바둑 지식을 탑재한 퍼지 추론 시스템에 의해 어느 정도 보상될 수 있다.

## 2.2 후보 착점들 평가

포석에 있어서의 착점 선택은 집을 당장 확보하는 단기적인 이익보다는, 효율적인 잠재 세력을 담보할 수 있는 착점을 함으로서 향후 점진적으로 집을 확장해 나갈 수 있는 장기적인 이익에 있다[3].

세력은 형세 판단을 할 수 있는 경험적 방법으로 자주 이용되며, 돌 간의 두터움에 관련된 바둑 이론 중의 하나이다. 바둑에서 한 돌로 부터의 세력은 다른 점에 미치는 영향으로 간주되며, 세력의 정도는 거리가 멀어질수록 감소한다. [4]는 박사학위 논문에서 세력 함수를 이용하여 최초의 컴퓨터 바둑을 구현하였으며, [5]는 바둑판 내에 있는 돌들의 세력을 계산해내기 위하여 [식 1]과 같은 간단한 세력 함수를 제안하였다.

$$F(d) = 2^{6-d} \quad \text{[식 1]}$$

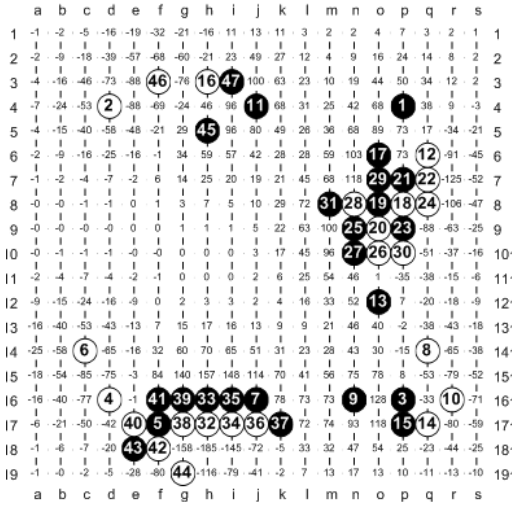
여기서  $d$ 는 세력을 내는 점과 세력을 받는 두 점간의 최단 거리, 즉 두 돌 간의 거리가 된다.

한편 저자는 3칸 이내로 떨어진 바둑돌 간의 연결 정도를 중시하여, [식 2]와 같은 거리에 따른 세력 함수를 제안하였으며, 제안된 세력 함수를 이용한 [그림 2]의 흑47 착수 후 세력도는 [그림 4]와 같다.

$$I(d) = C \cdot \exp\left(\frac{-d^2}{4}\right) \quad \text{[식 2]}$$

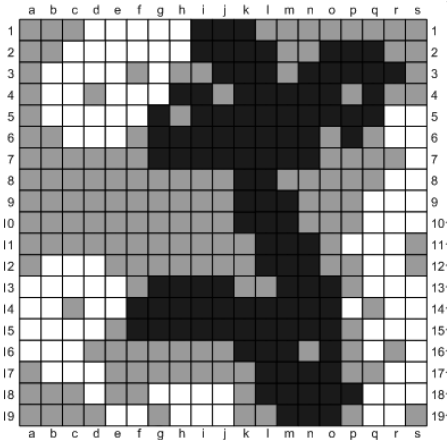
여기서  $d$ 는 두 돌 간의 최단 거리가 되며, 상수  $C$ 는  $d=0$ 일 때의 값으로 흑에게는 +64, 백에게는 -64가 부여되었다.

[그림 4]에서 양수 값은 흑의 세력 정도를 표시한 것이며, 음수 값은 백의 세력 정도를 표시한 것이다.



[그림 4] 흑 47 착수 후 세력도

또한 이 세력도를 활용하여 흑백 간의 잠재적 집의 형태를 나타내면 [그림 5]와 같다.

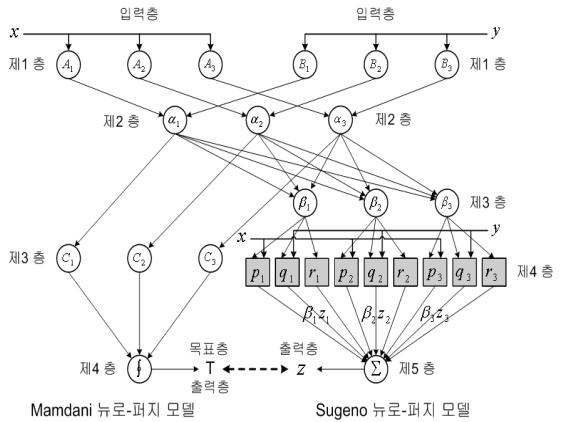


[그림 5] 잠재적 흑백 간의 영역도

[그림 5]에서 검은색 부분은 흑의 잠재적 집의 영역이며, 흰색 부분은 백의 잠재적 집의 영역이며, 회색은 흑백간의 중립 지역이 된다. 다시 말해, 세력 함수를 이용하여 각 후보 착점 후 발생하는 세력의 변화 값 및 잠재적 집의 변화 값을 구할 수 있으며, 이 값들은 뉴로-퍼지 제어기의 입력 자료로 활용된다.

### 2.3 뉴로-퍼지 제어기

일반적으로 뉴로-퍼지 제어기를 구현하기 위하여 Tsukamoto, Mamdani 및 Sugeno 모델들이 사용된다[6]. 이들 중 Mamdani 모델이 퍼지를 잘 표현할 수 있기에 가장 많이 사용되고 있으며[7], Sugeno 모델은 Mamdani 모델에 비해 여러모로 강점이 있는데도 불구하고 구현상의 어려움의 이유로 사용이 제한되고 있다[8,9]. 본 실험에서는 Mamdani와 Sugeno 모델의 혼합형인 [그림 6]과 같은 ANFIS(Adaptive Neuro Fuzzy Inference System) 모델의 변형인 초유의 ‘유사 ANFIS(SANFIS: Similar ANFIS)’를 제안하였다.



[그림 6] 유사 ANFIS(SANFIS)

SANFIS를 제안하게 된 근본 동기는 (1) Mamdani 모델에서 퍼지화된 입력 값을 비 퍼지화시키는데 너무 많은 계산 시간이 드는 단점을 획기적으로 개선 할 필요성, (2) 퍼지를 가장 잘 표현하는 Mamdani 모델을 대신 할 유사 모델인 Sugeno 모델을 구현해야 하는 필요성에 있었다.

SANFIS는 10개의 후보 착점들의  $x$ (세력의 변화율),  $y$ (잠재적 집의 변화율)라는 두개의 입력 변수를 정량적으로 평가한 후, 한 개의 출력 변수인  $z$ (후보 착점의 선택 정도)값을 추론해 내는 제어기가 되며, 적용된 퍼지 규칙은 다음과 같다.

제 1규칙:

IF  $x$ 가  $A_1$   
 AND  $y$ 가  $B_1$   
 THEN  $z$ 는  $z_1$

제 1규칙(설명):

만약 세력이 유리  
 하고 집수가 많아  
 지면 착수로 좋다.  
 $(z_1 = p_1x + q_1y + r_1)$

제 2규칙:

IF  $x$ 가  $A_2$   
 AND  $y$ 가  $B_2$   
 THEN  $z$ 는  $z_2$

제 2규칙(설명):

만약 세력이 보통  
 이고 집수가 보통  
 이면 착수로 보통이다.  
 $(z_2 = p_2x + q_2y + r_2)$

제 3규칙:

IF  $x$ 가  $A_3$   
 AND  $y$ 가  $B_3$   
 THEN  $z$ 는  $z_3$

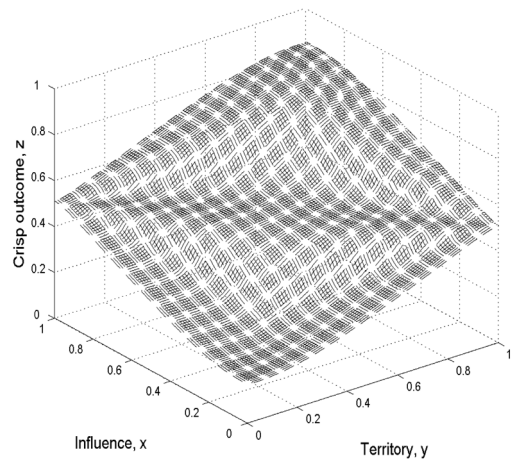
제 3규칙(설명):

만약 세력이 불리  
 하고 집수가 적어  
 지면 착수로 나쁘다.  
 $(z_3 = p_3x + q_3y + r_3)$

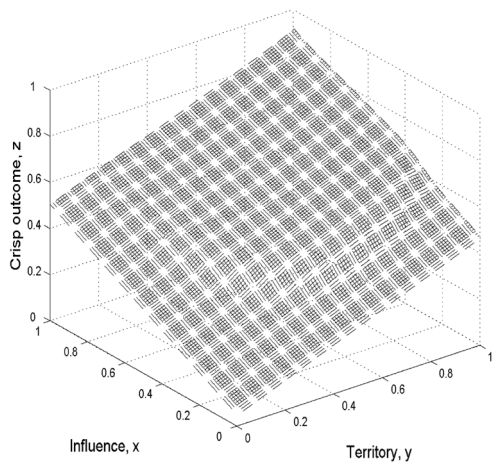
[그림 7]은 Mamdani 뉴로-퍼지에 의해 생성된 제어 함수 표면도이며, 이 표면도에 있는 값들을 목 표치로 하여 Sugeno 뉴로-퍼지가 100번 훈련 후

생성한 제어 함수 표면도는 [그림 8]과 같다. [그림 8]에 있는 제어 함수 표면도를 일차 함수로 표현하 면 [식 3]과 같으며, 이로서  $x$ ,  $y$ 의 값을 알게 되면 즉각적으로  $z_1$ ,  $z_2$ ,  $z_3$ 의 값을 산출해낼 수 있다.

$$\begin{aligned} z_1 &= +0.45x + 0.42y + 0.04 & \text{[식 3]} \\ z_2 &= +0.35x + 0.41y + 0.13 \\ z_3 &= -0.24x + 0.93y + 0.55 \end{aligned}$$



[그림 7] Mamdani 뉴로-퍼지 제어 함수 표면도

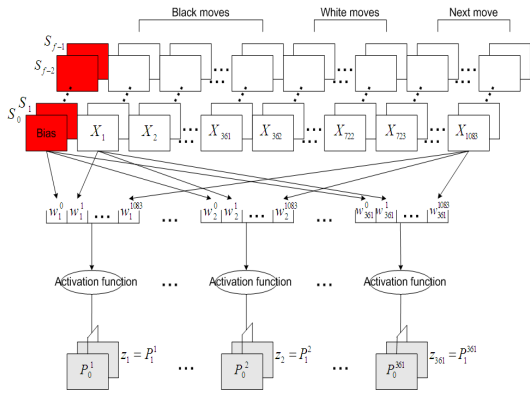


[그림 8] Sugeno 뉴로-퍼지 제어 함수 표면도

## 2.4 시차 학습

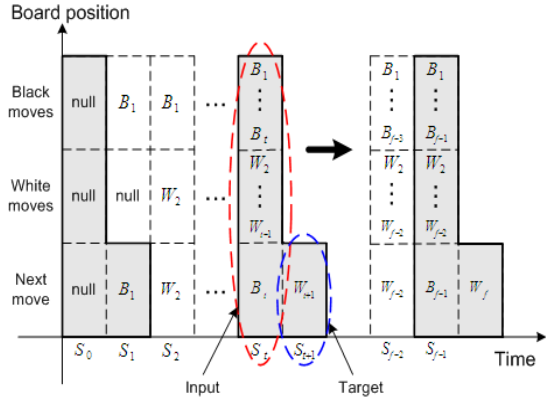
[10]에 의해 제안된 시차 학습(Temporal difference learning)은 가장 잘 알려진 강화 훈련(reinforcement learning)의 일종이며, 부여된 보상 값(scalar rewards)을 갖고 우리가 원하는 근사 함수(function approximator)를 구하는 통제 학습(supervised learning)이다[11]. 즉, 우리가 원하는 경로로 접근하는 경우는 보상을 해주어 이 경로를 선호하게 만들고, 반면에 우리가 원하는 경로로 접근하지 않는 경우는 벌을 주어 이 경로를 회피하게 하는 훈련 방식이다. 이와 같은 훈련을 계속 반복·강화함으로써 우리가 원하는 어느 정도 수준의 근사 함수를 얻어낼 수 있다.

포석 바둑에 적용할 시차 학습인 TD( $\lambda$ ) 학습 시스템을 위해 단일 계층망으로 된 [그림 9]와 같은 1083-361 신경망을 구현하였다.



[그림 9] 신경망 구조도

훈련용 데이터로 1,704개의 모범 포석이 사용되었고, 입력 데이터 및 목표치에 대한 구조도는 [그림 10]과 같다. 시간  $t$ (홀수인 경우)에서 [그림 10]에 있는 입력 데이터  $S_t$ 는 흑백의 착점들인  $\langle B_1, W_2, B_3, W_4, \dots, W_{t-1}, B_t \rangle$ 로 구성되었으며 이는 [그림 9]의 입력층에 입력이 된다. 반면에  $S_{t+1}$ 에 있는 다음 착점인  $W_{t+1}$ 는 출력층에 있는 목표치인  $z_i (i = 1, 2, \dots, 361)$ 로 입력이 된다.



[그림 10] 입력 데이터 및 목표치의 구조도

결국 신경망은 시간  $t$ 에서 [그림 10]에 있는 입력 데이터  $S_t$ 를 갖고 바둑판 내에 있는 모든 점들에 대한 예상치( $P_0^i, i = 1, 2, \dots, 361$ )를 산출해 내며, 이는 출력층에 있는 목표치  $z_i$ 에 대한 신경망의 추정치 정도를 나타내 준다.

## 3. 실험 결과와 논의 사항

일반적으로 기사들은 효과적/효율적인 포석 진행을 위해,

- 그들이 기억하고 있는 형상으로 부터 몇 개의 후보 착점을 정한 후,
- 지득하고 있는 사전 지식을 통해, 면밀하게 각 후보 착점들에 대한 일련의 모든 수순을 탐색한 후, 최선의 착점을 결정한다.

대부분 프로 기사들은 보통 4개 미만의 후보 착점들을 정한 후, 쌍방이 둘 수 있는 최선의 수들 중에서 절대 선수, 후수, 집들의 안정과 연결, 세력 등을 고려하여 차기 착점을 결정한다. 아직까지 바둑계에서는 포석 시 최선의 착점을 정할 수 있는 최선의 지침에 대한 공감대가 형성되어 있지 않으나, [12,13]은 세력과 집의 안정성이 그러한 목표를 달성키 위해 고려되어야 할 하나의 사안임을 제언하고 있다. 본 실험에서는 집들의 안정 정도 및 연

결 정도와 같은 바둑 지식을 컴퓨터로 구현해 내는데 상당한 어려움이 있는 관계로, 단지 세력에 대한 바둑 지식만 다루었다.

### 3.1 세력 함수

세력이란 바둑 이론에 있어서 두터움, 얽음이라는 바둑 개념과 밀접한 관계가 있으며, 포석 시 전술 수행을 위해 중요한 척도로 사용될 뿐만 아니라 형세 판단에도 자주 이용된다. 본 실험에서는 단순히 '돌 간의 거리에 반비례'하는 기존의 세력 함수인 [식 1]과는 달리 '3점 이내에 있는 두 돌 간의 두터움을 더욱 강조'한 새로운 세력 함수인 [식 2]를 제안하였다. 제안된 세력 함수를 활용하여 착수에 따른 세력도를 생성하였으며, 이 세력도를 이용하여 세력의 변화 및 잠재적 집수들을 계산해낼 수 있었다. 이러한 세력의 변화 및 잠재적 집수는 시차 학습 시스템인 TD( $\lambda$ )와의 대결을 위해, 최선의 차기 착점을 구하기 위한 뉴로-퍼지 제어기의 입력 자료로 활용되었다.

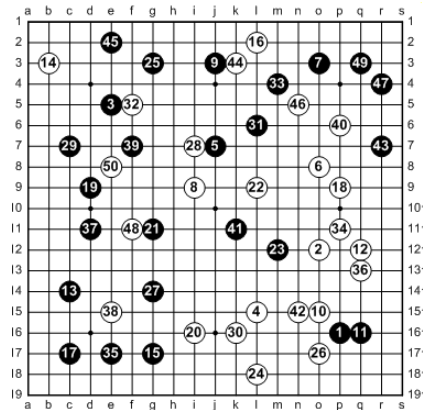
### 3.2 뉴로-퍼지 제어기

효과적/효율적인 포석 진행을 위한 많은 바둑 제언들이 있으나, 이를 컴퓨터에 적용시키는 것은 여간 쉽지가 않다. 본 실험에서는 '세력'과 '잠재적 영역'이라는 바둑 개념을 갖는 아주 단순한 세 가지 규칙을 뉴로-퍼지 제어기에 적용하였다. 일반적으로 바둑 개념을 구현할 수 있는 뉴로-퍼지 제어기로는 Tsukamoto, Mamdani 및 Sugeno 모델 등이 있다. Mamdani 모델이 바둑 개념 구현에 있어 이상적이거나, 문제는 퍼지화된 입력 값을 입력할 때마다 무게 중심법(center-of-gravity method)과 같은 방법을 통해 일일이 적분을 하여 비퍼지화시키는데 너무 많은 계산 시간이 든다는데 있다. 한편 Sugeno 모델은 비퍼지화를 위해 입력 값에 대해 일일이 적분을 할 필요없이 곧 바로 선형 처리가 가능하기 때문에 Mamdani 모델에 비해 계산 시간을 상당히 줄일 수 있으나, 문제는 Sugeno 모

델 내의 선형 처리를 위한 계수 값을 임의로 정할 수 없음에 있다. 본 실험에서는 Sugeno 모델내의 계수 값을 설정하기 위하여, Mamdani 모델로 부터 나온 비퍼지화된 결과 값을 목표치로 설정한 후 Sugeno 모델을 길들였다. 이 길들여진 Sugeno 모델은 향후 입력되는 값에 대해 비퍼지화를 선형 처리함으로서 계산 시간을 상당히 절감하였다. 즉, 제안된 SANFIS모델은 계산 시간을 상당 부분 절감하는 장점이 있는 반면에, 그 구현에 있어 Mamdani모델 보다 쉽지가 않는 단점도 갖고 있었다[14,15,16].

### 3.3 성능 평가상의 어려움

길들여진 퍼지 추론 시스템의 성능을 평가하기 위하여, 시차 학습 시스템인 TD( $\lambda$ )와의 대결을 벌였다. 대결을 위해 최종 착수는 50수로 제한하였으며, 퍼지와 TD( $\lambda$ ) 간에 흑백을 교대로 바꾸어 여덟 번의 대결을 벌였으며, TD( $\lambda$ )내의  $\lambda$ 값은 0.0, 0.3, 0.7 및 1.0의 4가지로 제한하였다. [그림 11]과 [그림 12]는 퍼지와 TD( $\lambda$ ) 간에 벌어진 두 기보를 보여주고 있다.

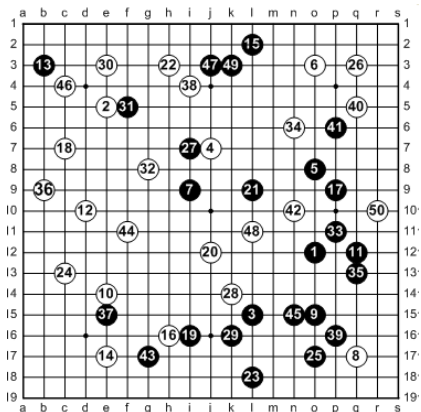


[그림 11] 흑 절대 우세: 흑(퍼지), 백(TD(0.7))

시행된 여덟 게임의 최종 형세 판단을 위해, 추정되는 승부 결과를 프로 9단에게 의뢰하여 흑 절

대 우세(흑+), 흑 우세(흑), 백 절대 우세(백+), 백 우세(백), 프로조차 단정하기 어려움(부정) 등으로 구분하였으며, 그 결과는 [표 1]과 같다.

시스템의 급수는 15급 정도가 됨을 추정할 수 있다[18].



[그림 12] 백 절대 우세: 흑(TD(0.7)), 백(퍼지)

[표 1] 승부 결과

	게임 1	게임 2	게임 3	게임 4
흑	퍼지	TD(0.0)	퍼지	TD(0.3)
백	TD(0.0)	퍼지	TD(0.3)	퍼지
결과	백+	백	백+	부정

	게임 5	게임 6	게임 7	게임 8
흑	퍼지	TD(0.7)	퍼지	TD(1.0)
백	TD(0.7)	퍼지	TD(1.0)	퍼지
결과	흑+	백+	백	백+

[표 1]에서 보는 바와 같이 어느 시스템이 절대 우위에 있다고 단정하기는 어려우나, 퍼지 추론 시스템이 TD( $\lambda$ ) 학습 시스템보다 약간 우위에 있음을 보였다.

한편 본 실험에 적용된 퍼지 추론 시스템과 TD( $\lambda$ ) 학습 시스템들의 급수를 정확히 판정하기는 어려우나, 1995년 FOST컵에서 7급을 부여 받은 Go++와의 대결 결과로 추정해 보면[17], 퍼지 추론시스템은 10~15급 사이가 되고, TD( $\lambda$ ) 학습

## 4. 결 론

본 논문에서 ‘세력’ 및 ‘잠재적 영역’이라는 바둑 개념을 구현하기 위하여, 규칙을 기반으로 한 퍼지 추론 시스템을 적용하였다. 실험 결과에 따르면, 간단한 퍼지 추론 시스템조차 시차 학습 시스템인 TD( $\lambda$ ) 학습 시스템을 능가하였으나, 좀 더 나은 퍼지 추론 시스템을 구현하기 위해서는 더 많은 양의 바둑 지식을 필요로 함을 발견하였다. 한편, 본 실험의 주 공헌으로는 계산 능력이 탁월한 Sugeno 형태의 뉴로 퍼지 모델을 적용하기 위해 Mamdani 모델과 Sugeno 모델을 합성한 새로운 뉴로 퍼지 모델인 SANFIS를 제시한 것이며, 아울러 바둑계에서 초유로 퍼지 추론 시스템과 신경망을 이용한 시차 학습 시스템 간의 대결을 벌인 것에 그 의의가 있다.

## 참고문헌

- [1] J. Burmeister and J. Wiles, “The Challenge of Go as a Domain: A Comparison Between Go and Chess”, Proc. of the Third Australian and New Zealand Conference on Intelligent Information Systems (1995) 181-186.
- [2] X. Cai and D. Wunsch, “A Parallel Computer Go Player, Using HDP Method”, Proc. of IEEE International Joint Conference on Neural Networks 4 (2001) 2373-2375.
- [3] Y. Moon, Discovery of Go: Understanding of Modern Go, Booki Press, Korea, 1998.
- [4] A. Zobrist, Feature Extraction and Representation for Pattern Recognition and the Game of Go, PhD Thesis, University of Wisconsin, USA, 1970.
- [5] K. Chen, “Heuristic Search in Go Game”, Proc. of Joint Conference on Information



- Sciences '98 2 (1998) 274-278.
- [6] J. R. Jang, "Neuro-Fuzzy Modeling and Control", Proc. of the IEEE 83 (1995) 378-406.
  - [7] M. Negnevitsky, Artificial Intelligence: A Guide to Intelligent Systems, Addison Wesley, England, 2005.
  - [8] J. Binfet and B. M. Wilamowski, "Microprocessor Implementation of Fuzzy Systems and Neural Networks", IEEE (2001) 234-239.
  - [9] D. Nauck and R. Kruse, "A Neuro-Fuzzy Approach to Obtain Interpretable Fuzzy Systems for Function Approximation", Proc. IEEE International Conference on Fuzzy Systems (USA, 1998) 1106-1111.
  - [10] R. S. Sutton, "Learning to Predict by the Methods of Temporal Differences", Machine Learning 3 (1988) 9-44.
  - [11] J. Baxter and A. Tridgell and L. Weaver, "Learning to Play Chess Using Temporal Differences", Machine Learning 40 (2000) 243-263.
  - [12] J. Burmeister and J. Wiles, "AI Techniques Used in Computer Go", Proc. of the Fourth Conference of the Australasian Cognitive Science Society (Australia, 1997).
  - [13] J. Burmeister, Studies in Human and Computer Go: Assessing the Game of Go as a Research Domain for Cognitive Science, PhD thesis, University of Queensland, Australia, 1995.
  - [14] Y. Jin and J. Jiang, "Techniques in Neural Network Based Fuzzy System Identification and Their Application in Control of Complex Systems", Fuzzy Theory Systems: Techniques and Applications 1 (1999) 112-128.
  - [15] J. Binfel and B. M. Wilamowski, "Microprocessor Implementation of Fuzzy Systems and Neural Networks", IEEE (2001) 234-239.
  - [16] D. Nauck and R. Kruse, "A Neural-Fuzzy Approach to Obtain Interpretable Fuzzy Systems for Function Approximation", Proc. IEEE International Conference on Fuzzy Systems (USA, 1998) 1106-1111.
  - [17] M. Reiss, Go++, Available from <http://www.goplusplus.com/go4ppfaq.htm> [accessed on 27 September 2009].
  - [18] B. Lee, Multi-strategic Learning, Reasoning

and Searching in the Game of Go, PhD Thesis, University of Auckland, New Zealand, 2005.



이 병 두(Byung-Doo Lee)

1982 한양대 원자력공학 학사  
1991 서강대 정보처리학 석사  
2005 Auckland University 컴퓨터공학 박사  
2007~현재 성결대 정보통신학부 강사

관심분야 : 컴퓨터공학, 인공지능, 컴퓨터바둑

---