論文2002-39TE-1-12

역전파 신경회로망과 Q학습을 이용한 장기보드게임 개발

(The Development of Janggi Board Game Using Backpropagation Neural Network and Q Learning Algorithm)

黃 庠 文 *, 朴 仁 圭 **, 白 德 洙 ***, 晋 達 福 ****

(Sang-Moon Hwang, In-Kue Park, Deok-Soo Baek, Dal-Bok Chin)

요 약

본 논문은 2인용 보드게임의 정보에 대한 전략을 학습할 수 있는 방법을 역전파 신경회로망과 Q학습알고리즘을 이용하여 제안하였다. 학습의 과정은 단순히 상대프로세스와의 대국에 의하여 이루어진다. 시스템의 구성은 탐색을 담당하는 부분과 기물의 수를 발생하는 부분으로 구성되어 있다. 수의 발생부분은 보드의 상태에 따라서 갱신되고, 탐색커널은 αβ 탐색을 기본으로 역전파 신경회로망과 Q학습을 결합하여 게임에 대해 양호한 평가함수를 학습하였다. 학습의 과정에서 일련의 기물의 이동에 있어서 인접한 평가치들의 차이만을 줄이는 Temporal Difference학습과는 달리, 기물의 이동에 따른 평가치에 대해 갱신된 평가치들을 이용하여 평가함수를 학습함으로써 최적의 전략을 유도할 수 있는 Q학습알고리즘을 사용하였다. 일반적으로 많은 학습을 통하여 평가함수의 정확도가 보장되면 승률이 학습의 양에 비례함을 알 수 있었다.

Abstract

This paper proposed the strategy learning method by means of the fusion of Back-Propagation neural network and Q learning algorithm for two-person, deterministic janggi board game. The learning process is accomplished simply through the playing each other. The system consists of two parts of move generator and search kernel. The one consists of move generator generating the moves on the board, the other consists of back-propagation and Q learning plus αβ search algorithm in an attempt to learn the evaluation function. while temporal difference learns the discrepancy between the adjacent rewards, Q learning acquires the optimal policies even when there is no prior knowledge of effects of its moves on the environment through the learning of the evaluation function for the augmented rewards. Depended on the evaluation function through lots of games through the learning procedure it proved that the percentage won is linearly proportional to the portion of learning in general.

* 正會員, 全州工業大學 電子情報科
(Jeonju technical college Dept. of Info-Electronics)
** 正會員, 中部大學校 情報工學部 電子計算學科
(Joongbu University Dept. of computer science)
*** 正會員, 益山大學 電子情報科
(Iksan national college Dept. of Info-Electronics)
*** 正會員, 圓光大學校 電氣電子工學部 電子工學科
(Wonkwang University Dept. of Electronic Eng.)
接受日字:2002年2月21日, 수정완료일:2002年3月19日

Ⅰ. 서 론

기계와 게임과 추론간의 관계는 많은 사람들의 관심의 대상이 되어왔다. 1948년에 로버트 위너(Robert Wiener)는 기계와 인간의 잠재력사이에 근본적인 차이점을 나타내기 위한 일환으로 체스의 경기수준을 향상시키는데 몰두하였다. 또한 1950년에 알란 튜링(Alan Turing)은 기계가 생각을 할 수 있는지에 대해 가상게임을 제안하였고, 클라우드 샤논(Claude Shannon)은 체스의 경기수준을 생각을 할 수 있는 기계의 척도로 많은 대안을 제안하였다[1]. 1951년에 처음으로 디지털 컴퓨터가 사용되기 시작한 이후에 기계와 게임과 추론에 관한 이론과 제안들이 이론적인 범주를 벗어나 실용을 띄게 되었다. 결국 게임을 할 수 있는 컴퓨터프로그램이 인공지능 분야의 주요 관심분야가 되었다. 초기의 체스프로그램의 수준은 초보자의 수준이었다.

그러나 오늘날 많은 프로그램이 상당한 수준에 이르러 있고, 현재 컴퓨터 체스의 챔피언인 DEEP THOUGH는 토너먼트선수의 점수가 평균1500이고 세계챔피언은 2900인데 반해 2600에 도달해 있다. 1951년에 파울 리차드(Paul Richard)와 마빈 윈버그(Marvin Weinberg)는 기계가 학습을 할 수 있는 아이디어를 제안하였고, 이 이론에 따르면 인공지능의 기법을 이용하여 단순히 게임을 하여 기계가 지능을 확보할 수 있음이 밝혀졌다.

대개의 보드게임은 주어진 일정한 시간에 많은 판단을 요구한다. 이러한 판단에 대한 결과를 예견하여 게임을 하기에는 주어진 시간이 부족할 뿐만 아니라, 일단 결정이 내려지면 번복할 수가 없다. 여기에 적군의 전략이 불확실하기 때문에 결과 또한 불확실한 것이 특징이다. 이러한 게임의 특징은 게임의 학습기능을 고려하면 많은 판단이 요구되는 상황에 잘 부합할 것이다. 이와 같은 판단이 필요한 분야로는 자연어의 처리, 영상처리, 수학적 이론의 증명과 정보 처리등이 있다. 보드게임은 컴퓨터로 구현하기가 간단하고 승패를 가리는데 분명한 판단기준이 있고 큰 데이터베이스를 필요로 하지 않기 때문에 안성맞춤이다. 많은 판단이 요구되는 문제의 가장 큰 특징은 판단이 조합적 이라는 사실이다[2-3]. 특히 결론을 도출하기 위하여 수많은 조합의 수를 모두 탐색하는 것도 가능 하지만, 그 조합이 지수 함수적으로 늘어나기 때문에 실제로는 실현 불가능하다. 예를 들어 샤논이 추정한 수치를 보면 체스의 경우의 수를 보면 10^{120}경우의 게임이 가능하다고 알려져 있다. 모든 경우의 수를 탐색하는 것이 가능하지 않기 때문에 가장 적합한 해를 얻기 위한 다른 방법이 필요하다. 체스와 같은 2인용 보드게임은 이러한 조합적인 문제에 대한 해결 방법을 연구하기 위한 좋은 실험적인 분야를 제공하고 있다.

이와 같은 탐색의 폭발성을 극복하고 조합적인 탐색의 효율성을 부여하기 위한 일환으로 본 논문에서는 단순히 게임을 함으로써 2인용 보드게임에 대한 학습기능을 가지는 장기보드게임을 구현하는 것을 목표로 하고 있다.

Ⅱ. 게임트리 탐색

1. minimax알고리즘

게임트리의 루트는 게임의 현재의 상태를 나타낸다. 트리의 각 노드는 일단의 자식노드를 가지고 있다. 하나의 노드가 가지는 각 자식노드는 그 노드에서 한 수를 둔 이후의 새로운 상태를 나타낸다. 이러한 과정은 게임트리에서 자식노드가 하나도 없는 단말 노드에 도달할 때까지 이어져서 각 단말 노드에 이득 값(payoff)을 발생한다. 일반적인 게임에서 이러한 값은 양 선수에게 있어서 최종위치에 대한 이동도(utility)를 나타낸다. 일반적으로 게임에서 이기는 경우는 양의 이동도를 가지고 패하는 경우는 음의 이동도를 가진다[4-5].

그림1의 OX게임에서 두 명의 선수가 서로 교대로 게임을 할 경우에 X는 루트에서 적군 O는 루트의 바로 아래에서 시작한다. 하나의 위치는 트리에서 일단의 레벨(ply)을 통하여 나타낼 수 있다. 다른 보드게임에서와 같이 OX도 승(1), 패(-1), 비김(0)의 세 가지의 경우가 있다. 각 선수에게 최대(Max)와 최소(Min)의 이름을 부여하면 최대는 점수를 최대로 하는 수를 두게 되고 반대로 최소는 점수를 최소화 해주는 수를 두게 된다. 이러한 방식으로 트리의 모든 노드는 이득 값인 최대최소값을 할당받는다. 최대의 가장 우수한 수는 트리의 루트와 같은 최대최소값을 가지는 수이다. 따라서 이와 같은 값을 가지는 값을 통하여 트리를 따라 내려갈 경우에 이 경로가 각 선수에게는 최상의 경로를 나타내며 이를 우수변량(Principal Variation)이라고 한다.

2. αβ 알고리즘

브러드노(Brudno)에 의해서 처음으로 소개된 αβ알고리즘은 최대최소알고리즘을 개량한 것이다. 트리의 노드마다 두 개의 한계 α와 β가 사용된다. 이 값은 깊이 탐색에 따라서 인가된다. 각 노드에서 α값은 가장 작은 값을 나타내며 트리의 상위노드들의 최대최소값에 영향을 줄 수 있다. 반면에 β값은 최대최소값에 영향을 줄 수 있는 가장 큰 값을 나타낸다. α는 자기 자신의 노드를 포함하여 연결되어 있는 최대노드들의 평가된 가지들중에서 가장 큰 최대최소값을 나타낸다. 최대노드아래의 각 부 트리가 탐색되어 짐에 따라서 α는 점차적으로 증가한다. 따라서 탐색경로를 따라 트리를 탐색해 가는데 있어 α는 단조적으로 증가한다. 이와 유사하게 β도 자기 자신의 노드를 포함하여 그 노드에 연결되어 있는 최소노드에서 평가된 가지 중에서 가장 작은 최대최소값을 나타낸다. α가 β보다 크거나 같은 위치에 이르면 트리의 루트에 가까운 우수한 경로가 있음을 알 수 있다. α≥β인 노드아래는 더 이상 탐색할 필요가 없고 바로 부모 노드로 복귀할 수 있다. 결과적으로 최대최소값에 영향을 주지 않는 노드들을 삭제하게 된다. αβ알고리즘은 루트가 탐색창(-∞, +∞)으로 탐색이 되어진다면 올바른 최대최소값 을 반환할 수 있다.

negamax알고리즘을 이용한 αβ알고리즘이 그림 2에 나타나 있다. 알파(alpha)와 베타(beta)를 다음 레벨로 패스다운함에 따라서 한계가 항상 alpha에 유지될 수 있도록 두 개의 매개변수를 반전하고 교환한다. 이 알고리즘에서 트리의 짝수ply(최대노드)에서는 알파가 상승하고, 홀수ply(최소노드)에서는 베타가 감소하여 αβ알고리즘의 조건에 부합한다.

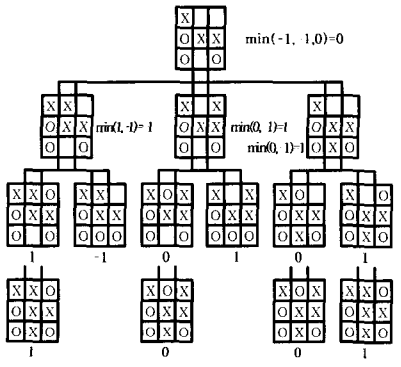

그림 1. Minimax값을 가지는 OX게임트리
Fig. 1. Naughts and crosses game tree with minimax values.

```
int AlphaBeta(position p, int alpha, int beta) {

        int numOfSuccessors;
        int gamma;

        int i;
        int sc;


        if(EndOfSearch(p)) { return(Evaluate(p));}
        gamma = alpha;
        numOfSuccessors=GenerateSuccesssors(p);
        for(i=1; i <= numOfSuccessors; i++) {
            sc=-AlphaBeta(p.succ[i],-beta,-gamma);
            gamma=max(gamma,sc);
            if(gamma >= beta) {return(gamma);}
        }
        return(gamma);

}
```

그림 2. αβ알고리즘의 negamax구현
Fig. 2. The negamax formulation of αβ algorithm.

그림3은 얕은 차단을 보여준다. 깊이 탐색에 의하여 트리를 좌에서 우로 진행해 가면 먼저 루트의 좌측가지를 탐색하게 되고 첫째의 수는 최대화측 에게는 4의 최대최소값을 발생한다. 따라서 우측의 가지를 탐색하게 되면 α는 4로 고정이 되어 있는 상태에서 노드G는 1의 최대최소값을 가지게 된다. 그리고 β는 노드C에서 1로 감소된다. 최소측은 노드C의 1을 가지고 있지만 최대는 노드B로 이동함으로써 4의 값을 가지게 된다. 따라서 노드C의 다른 자식노드를 탐색할 필요가 없게 된다. 최대측은 노드C로 가는 것보다 노드B로 이동할 것이다. 따라서 노드H와 I는 탐색에서 제외될 것이다.

αβ알고리즘은 깊은 차단을 발생할 수 있다. 깊은 차단은 깊이 d에서의 정보에 따라서 깊이 d보다 더 깊은 노드들을 탐색에서 제외시킬 수 있다. 예를 들어 그림4에서 노드B아래의 부 트리를 탐색함으로써 얻어진 탐색창의 값에 따라서 노드J를 탐색에서 제외할 수 있다.

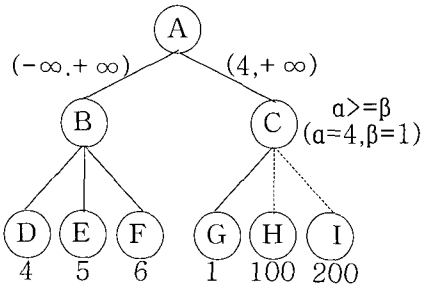

그림 3. αβ 의 얕은 차단
Fig. 3. αβ shallow cutoff.

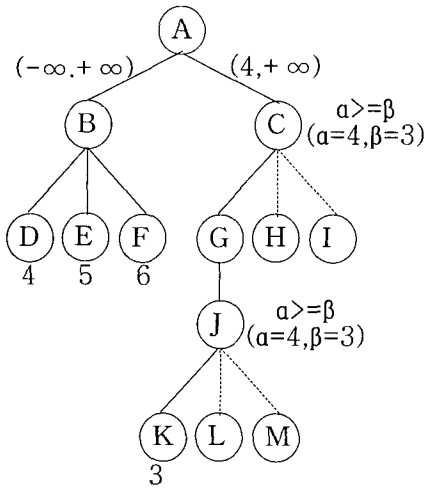

그림 4. αβ 의 깊은 차단
Fig. 4. αβ deep cutoff.

Ⅲ. 인공신경회로망

1. 역전파 알고리즘

역전파의 구조는 지도학습의 일종으로써 출력단의 오차를 역방향으로 전파하여 다음의 전방향의 계산을 위하여 오차를 줄여 나가는 방식으로 그림5와 같다. 또한 다층의 구조이고 전방향이며 가중치의 연결은 다양한 형태를 취할 수 있다. 따라서 기물의 이동에 따른

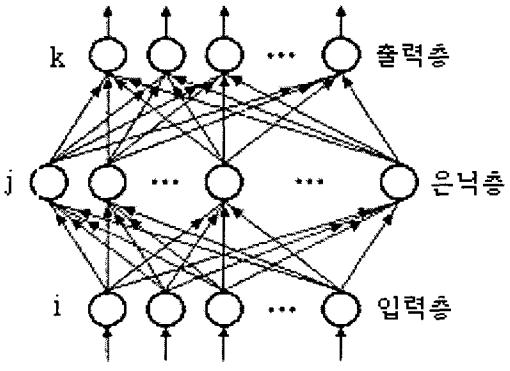

그림 5. 역전파 알고리즘
Fig. 5. Backpropagation Neural Network.

평가함수의 평가치에 대하여 역전파의 학습이 이루어진다. 역전파의 기본적인 알고리즘은 기존의 알고리즘과 동일하다.

2. Q 학습알고리즘

Q학습알고리즘은 강화학습의 하나로 최적의 목적을 달성하기 위하여 임의의 환경에서 가장 우수한 행동을 도출할 수 있는 학습알고리즘이다. 이에 대한 응용으로는 각기의 서로 다른 상태의 천이를 반복하여 이루어지는 로봇의 주행학습, 보드게임 등이 있다.

본 논문에서의 보드게임과 같은 경우에 하나의 상태에서 여러 가지의 다른 상태로의 천이에 대한 결정을 하기 위해 평가함수를 사용하고 있다. 이러한 경우에 Q학습알고리즘을 이용하여 가장 우수한 경로를 지정할 수 있다. 보드게임의 경우에도 임의의 상태(State)에 대한 기물의 움직임(action)들의 연속(r(si, ai), i=0,1,2...)으로 나타낼 수 있다. 따라서 이러한 종류의 학습정보에 의해 상태와 행동에 대해 정의되는 수치적인 평가함수를 학습하여 이 평가함수를 바탕으로 최적의 경로를 설정할 수 가 있다. 그림에서와 같이 각각의 상태에서 다른 상태로의 평가치가 있을 때 상태1에서 상태2로 장기의 기물이 이동하여 갱신되는 평가치는 다음의 식 (1)에 의하여 얻어진다.

$$Q(s,a) = r(s,a) + \gamma \max_{a'} Q(\delta(s,a),a') \qquad (1)$$

$r(s,a)$ 는 현재상태의 이동에 의해서 얻어지는 보상값이고 γ MAXQ(δ(s,a),a')는 다음의 새로운 상태($\delta(s,a)$)에서 다른 여러 가지 이동(a')들 중에서 최대화를 보장하는 값들의 합으로써 상태차이에 대한 평가치가 갱신되어 진다. 다음은 Q학습의 전체적인 알고리즘이다.

단계1. 각각의 상태와 이동의 평가치 초기화.
단계2. 현재의 상태(s)를 고려.
단계3. 하나의 이동을 결정하여 이동.
단계4. 상태이동에 따른 보상값.
단계5. 새로운 상태(s')의 고려.
단계6. 식1에 따른 평가치의 갱신
단계7. 상태의 천이(s→s')
단계8. 단계3으로 반복수행

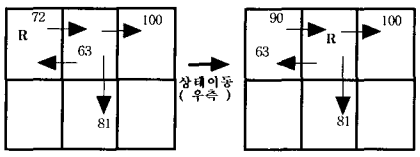

그림 6. Q학습의 동작
Fig. 6. The operation of Q learning.

결국 이전의 상태에서 새로운 상태로 이동할 때마다 Q학습에 의하여 평가치가 새로운 상태에서 이전의 상태로 역전파되고, 동시에 다른 상태로의 이동에 의하여 얻어지는 보상값에 의하여 평가치가 증가한다. 이는 기물들이 가지는 값에만 의존하여 학습이 이루어지지 않고, 갱신된 평가치 들에 대하여 학습이 이루어진다. 단순히 연속적인 최종 평가치 들의 차이에 대하여 학습이 이루어지는 Temporal Difference학습에 비하여 양호한 결과를 보장한다.

$$\Pi^*(s) = \underset{a}{\text{argmax}}\ Q(s,a) \qquad (2)$$

위의 식에서 도출된 Q의 평가치 들에 대하여 최적의 결정은 Q의 값을 최대화하는 이동으로써 위의 식 (2)에 의하여 정하여진다.[8]

3. 장기의 구조

본 논문의 프로그램은 크게 두 개의 부분으로 그림7에서와 같이 구성되어 있다. 첫째부분은 탐색과 학습으로 구성되어 있고 다른 부분은 적법한 수를 발생시키는 부분이다.

게임의 규칙은 수 발생기(move generator)에 의하여 발생된다. 따라서 수 발생기, 수 두기와 게임의 종료에 의하여 게임을 진행하는데, 수 발생기는 임의의 주어진 위치에 대하여 움직일 수 있는 모든 수를 발생시키며, 수두기는 일단 수가 두어지면 보드의 위치를 갱신한다. 마지막으로 게임종료는 게임의 승패와 비김을 나타낸다.

본 논문에서는 두 개의 평가함수를 사용하는데 하나는 아군에 대한 평가함수이고 다른 하나는 적군의 것이다. 아군이 움직일 수 있는 수들을 이용하여 아군의

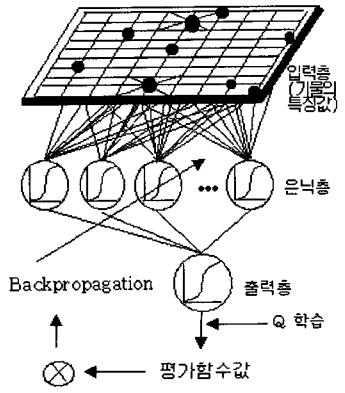

그림 7. 장기의 구조
Fig. 7. The architecture of janggi game.

표 1. 특징 값의 구성
Table 1. The formation of feature vectors.

항목	내 용
위치 특징	-현재의 보드에 대한 위치에서 계산. -보드상의 각 위치에 기물 1, 아니면 0. -한 기물이 여러 개면 두 개의 특징벡터는 1이고 세 번째는 두 개를 제외한 기물의 개수.
특징	-움직인 수와 먹힌 기물의 특징 값은 1.
규칙 특징	- 위험에 노출되어 있는 기물과 위치는 1. - 먹을 수 있는 기물과 위치는 1.

```
void startAutogame( )

{
 #define READ_PIPE     0
 #define WRITE_PIPE    1
 static int pid;
 static int fdRead[2],fdWrite[2];
 int status;
 status=pipe(fdread);

 if(status<0) perror("Error opening read pipe to
                      janggi.₩n");
 status=pipe(fdwrite);

if(status<0) perror("Error opening write pipe to
                     janggi.₩n");
 signal ( SIGCHLD, DeadChild);
 pid=fork( );

 if(pid==0){
   close(fdRead[READ_PIPE]);
   close(1);
   dup(fdRead[WRITE_PIPE]);
   close(fdRead[WRITE_PIPE]);
   close(fdWrite[WRITE_PIPE]);
   close(0);
   dup(fdWrite[READ_PIPE]);
   close(fdWrite[READ_PIPE]);
   execl("/root/gcc/opponent", "opponent",
         "-1",(char*)0);
   perror(" Error after starting janggi process.₩n");
   exit(1);
 }
else if (pid== -1)
   perror(" There has been an error forking the janggi
          process.₩n");

   close(fdRead[WRITE_PIPE]);
   close(fdWrite[READ_PIPE]);
}
```

그림 8. 두 프로세스간 통신
Fig. 8. The interface of two processes.

평가함수를 학습하고 적군의 평가함수도 적군의 수에 의하여 학습된다.

하나의 평가함수를 사용하면 양질의 수에 대하여 선수의 분별이 혼선의 우려가 있기 때문에 두 개의 평가함수를 사용하였다. 각각의 평가함수는 하나의 은닉층을 가지는 역전파 신경회로망의 출력이다.

게임의 유효한 특징 값들을 발견하기 위하여 평가함

수 학습알고리즘을 이용하는 데에는 실제로 게임의 학습에 있어서 아주 많은 학습패턴이 필요하다. 따라서 표1과 같이 특징 값들의 항목을 추출하여 학습을 수행하였다.

게임을 하는 동안에 수행되는 각각의 수들은 신경회로망을 구성하는 평가함수에 대한 각각의 입력패턴을 형성한다. Q 학습에 의하여 갱신된 평가치들은 각각의 패턴에 대한 기대치(target value)로 작용하여 역전파 신경회로망의 학습이 이루어진다. 따라서 역전파 알고리즘은 각각의 패턴에 대하여 네트워크의 가중치를 적응시킴으로써 게임에 대한 지능을 가지게 된다.

두 프로세스간의 학습의 과정은 그림 8과 같이 fork 함수와 pipe를 이용하여 양방향 통신이 가능하도록 하여 구성하였고, 이에 대한 아군과 적군의 구성도는 그림 9와 같다.

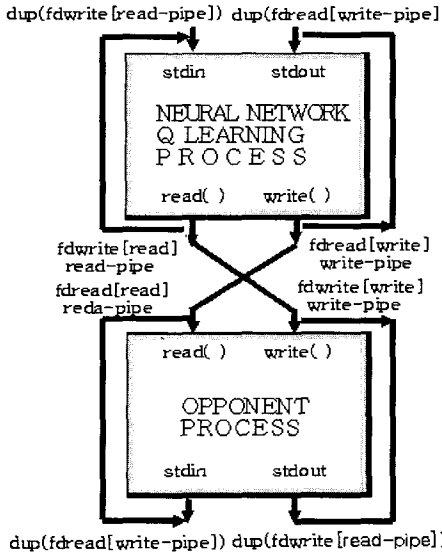

그림 9. 학습을 위한 인터페이스 쉘
Fig. 9. Interface shell for learning.

Ⅳ. 결과고찰

본 논문에서 제안한 장기와의 학습을 위하여 신경회로망이 없는 일반적인 장기프로그램(적군 프로세스)을 ANSI C로 작성하여 1330MHz의 IBM PC상에서 평균적으로 1000번의 게임을 진행하는 데에 20일정도의 시간이 걸려 학습을 진행하였다. 적군 프로세스의 ply는 4로 가정하였다. 이는 통신상의 프로그램의 중급에 해당한다. 물론 ply를 높이면 급수를 높일 수는 있다. 신경회로망을 가지는 프로그램(아군 프로세스)과의 학습은 Linux상에서 fork()함수와 pipe를 이용한 인터페이스 쉘(auto.c)을 이용하여 그림9와 같이 두 프로세스간에 서로 수를 주고받으며 그림과 같이 게임을 진행하였다. 학습과정에서 적군이 이기기까지 아군이 둔수는 평균적으로 20수정도 두었다. 평균적으로 제안한 아군은 최대노드를 3000노드에서 1500~1800노드를 탐색하였다. 14개의 기물과 10*9의 보드에 대하여 1594(보드기물위치*기물유형+기물유형*특징벡터*3+기물유형*2+보드기물위치*2+2+1)개의 입력을 구성하여 진행하였다. 신경회로망의 네트워크는 1594×104×1이며 신경회로망의 초기 가중치는 -0.05~0.05안의 난수를 이용하였다. 학습률은 일반적으로 적용하는 0.1로 적용하였다. 그림 10은 두 프로세스가 서로 메시지를 주고받기 위한 보드이다. 보드에서 각각의 기물의 값은 졸(J)이 1, 마(M)가 5, 상(S)이 3, 포(P)가 7, 차(C)가 13, 사(B)가 3 그리고 왕(K)은 154이다. 메시지의 구성은 A1의 위치에서 B1의 위치로 이동하기 위한 명령은 move a1-b1이다. 학습의 실험은 500번의 게임에서부터 2500번까지 500번 단위로 학습을 진행하였다. 이러한 과정에서 게임의 횟수가 증가함에 따라 즉, 학습의 횟수가 늘어남에 따라 신경회로망의 프로세스가 가지는 지능이 증가함을 알 수 있었다. 이는 학습의 패턴이 늘어남에 따라서 신경회로망의 가중치의 적응이 증가했다는 것을 알 수 있었다.

2500번의 학습을 수행한 아군의 컴퓨터와 적군(4 ply 중급)의 또 다른 컴퓨터에 대하여 사람이 마우스를 이용하여 상호의 수를 두어 50번의 게임을 수행한 결과, 41번을 이기고 9번을 졌다. 이는 보다 많은 학습을 통

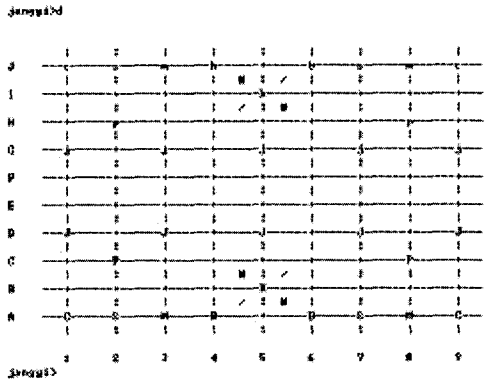

그림 10. 초기의 보드위치
Fig. 10. The initial board position.

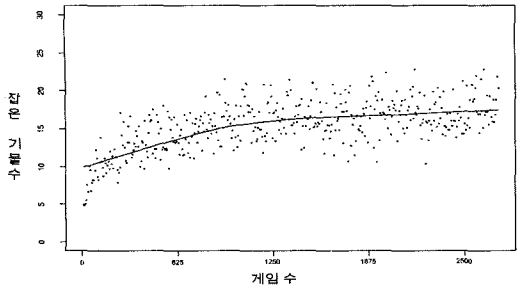

그림 11. 게임종료전 장기의 움직인 기물수
Fig. 11. The number of moves that janggi process had made before the game is over.

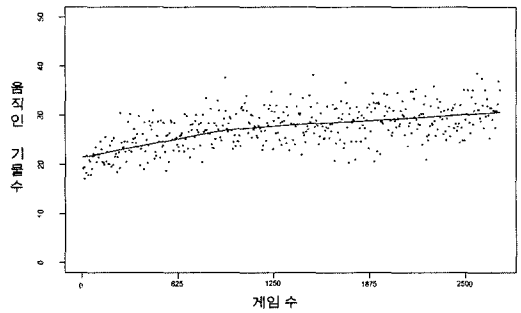

그림 12. 게임종료전 장기의 잡은 기물수
Fig. 12. The number of points that janggi process had captured before the game is over.

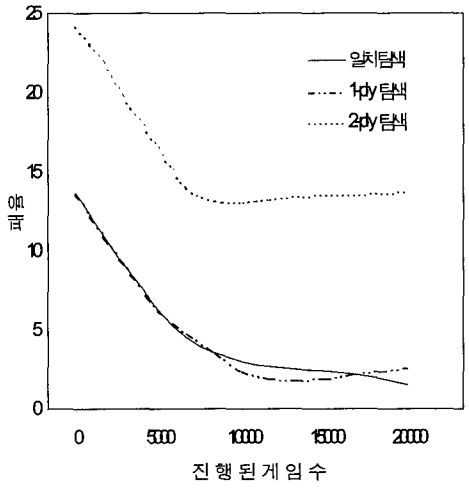

그림 13. 학습과 승률과의 관계
Fig. 13. The relation between learning and winning.

하여 가중치가 가지는 공간에 대한 균일한 학습이 이루어지지 않은 결과로 분석된다.

이러한 점을 보완하기 위한 방법으로는 역전파의 국부해가 가지는 단점을 극복하기 위하여 해에 대한 등판능력이 우수한 유전자 알고리즘을 이용할 수도 있다. 그림 11과 12는 적군에 대하여 게임을 하는 동안에 10게임 단위로 아군이 둔 수의 개수와 잡은 기물의 수에 대한 분포도이다. 그림13에서와 같이 학습과 승률과의 관계를 알아보기 위하여 탐색의 종류를 일치탐색, 1ply탐색과 2ply탐색의 세가지로 구분하여 실험을 수행하였다. 다만 탐색의 깊이에 따라서 시간적인 제약을 고려하기 위하여 탐색의 깊이를 작게하여 수행하였다. 일반적으로 학습이 진행될수록 승률이 좋아짐을 알 수 있었다. 2ply탐색은 충분한 횟수의 게임이 진행될때 까지는 일치탐색보다 비교적 양호한 결과를 보였다. 이것은 일치탐색의 과정이 아주 정확한 평가함수에 대해서는 유리하고 이러한 정확성은 많은 게임에 대해서 만이 얻어짐을 알 수 있었다.

V. 결 론

본 논문에서는 Q학습 알고리즘과 역전파 신경회로망을 이용하여 평가함수에 대한 학습을 통하여 지능을 가지는 장기보드게임을 개발하였다. 제안된 프로그램에서는 탐색과 학습을 융합하여 역전파의 단점을 극복하였다. αβ탐색알고리즘의 골격에 성능을 향상시킬 수 있는 부가기법을 제외하였다. 이는 순수한 학습의 성능을 측정하기 위한 일환이었다. 이러한 부가적인 기능과 많은 수의 학습에 지능의 정도가 비례하였다.

결과적으로 학습의 정도에 따라서 신경망의 가중치의 적응이 부합하였다. 앞으로 유전자알고리즘과 Hill Climbing등을 이용하여 학습의 오차를 보다 더 줄일 수 있는 방향으로의 연구가 병행되어야 할 것이다.

참 고 문 헌

[1] Boyan, J. A. (1992). Modular neural networks for learning. Master's thesis, University of Cambridge. Available via FTP from archive. ohiostate.edu:/pub/neuroprose.

[2] Hecht-Nielsen, R.(1989). Neurocomputing. Addison-Wesley Publishing Company, Inc. Holland, J. H. (1983). Escaping brittleness. In Proceedings of the International Machine Learning

Workshop, pp 92~95.

[3] Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational. abilities. In Proceedings of the National Academy of Sciences USA, volume 79, pp 2554~2558.

[4] Lee, K.-F. and Mahajan, S.(1988). A pattern classification approach to evaluation function learning. Artifical Intelligence, 36,1-25.

[5] McKinsey, J. C. (1952). Introduction to the theory of games. The RAND Series. McGraw-Hill Book Company, Inc.

[6] Minsky, M. and papert, S. (1969). Perceptrons. MIT Press, Cambirdge. Shannon, C. E. (1950). Programming a computer for playing chess. Philosophy Magazine, 41,256-275.

[7] Sutton, R. S. (1984). Temporal credit assignment in reinforcement learning. PhD Thesis, University of Massachusetts, Amherst.

[8] Tom M. Mitchell, (1997). Machine learning, The McGraw-Hill Companies, Inc. pp. 367~387.

저 자 소 개

黃 庠 文(正會員)

1989년 : 원광대학교 전자공학과 졸업(공학사). 1991년 : 원광대학교 대학원전자공학과 졸업(공학석사). 1998년 : 원광대학교 대학원전자공학과 박사과정수료. 1992~현재 : 전주공업대학 전자정보과 <관심분야 : 인공지능, 영상처리, 최적화이론>

朴 仁 圭(正會員)

1985년 : 원광대학교 전기공학과 졸업. 1987년 : 연세대학교 대학원 전산기응용(공학석사). 1997년 : 원광대학교 대학원 졸업(공학박사). 현재 : 중부대학교 컴퓨터과학과 조교수. <관심분야 : 인공지능, 최적화이론, 영상처리>

白 德 洙(正會員)

1988년 : 원광대학교 전자공학과 졸업. 1990년 : 숭실대학교 대학원 졸업(공학석사). 1996년 : 원광대학교 대학원 졸업(공학박사). 현재 : 익산대학 전자정보과 교수. <관심분야 : 신경회로망>

晉 達 福(正會員)

1958년 : 조선대학교 전기과 졸업(학사). 1970년 : 조선대학교 전기과 전임강사. 1972년 : 조선대학교 대학원졸업(석사). 1985년 : 전남대학교 대학원졸업(박사), 1982년~현재 : 원광대학교 전자과 교수. 1983~1987년 : 원광대학교 전자계산소장. 1997~1991년 : 원광대학교 공과대학장. 1991~1993년 : 원광대학교 기획처장, 1997~1999년 : 원광대학교 사회교육원장겸 정보교육원장. (전공)마이크로 프로세서 응용