한국게임학회 논문집
제1권 제1호 2001년 11월

역전파 신경회로망과 강화학습을 이용한 2인용 장기보드게임 개발

박인규, 정광호
중부대학교 정보공학부 전자계산학과

The Development of Two-Person Janggi Board Game Using Backpropagation Neural Network and Reinforcement Learning

In-Kue Park, Kwang-Ho Jung
Division of Information Engineering, Dept. of Computer Science, Joongbu

Abstract

This paper describes a program which learns good strategies for two-person, deterministic, zero-sum board games of perfect information. The program learns by simply playing the game against either a human or computer opponent. The results of the program' s learning of a lot of games are reported.

The program consists of search kernel and a move generator module. Only the move generator is modified to reflect the rules of the game to be played. The kernel uses a temporal difference procedure combined with a backpropagation neural network to learn good evaluation functions for the game being played. Central to the performance of the program is the search procedure. This is a the capture tree search used in most successful janggi playing programs. It is based on the idea of using search to correct errors in evaluations of positions. This procedure is described, analyzed, tested, and implemented in the game-learning program. Both the test results and the performance of the program confirm the results of the analysis which indicate that search improves game playing performance for sufficiently accurate evaluation functions.

1. 개요

기계와 게임과 추론간의 관계는 많은 사람들의 관심의 대상이 되어왔다. 1948년에 Robert Wiener는 기계와 인간의 잠재력사이에 근본적인 차이점을 나타내기 위한 일환으로 체스의 경기수준을 향상시키는데 몰두하였다. 또한 1950년에 Alan Turing은 기계가 생각을 할 수 있는지에 대해 가상게임을 제안하였고, Claude Shannon은 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 Paul Richard와 Marvin Weinberg는 기계가 학습을 할 수 있는 아이디어를 제안하

였고, 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 정보처리등이 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 조합적이라는 사실이다[2,3].

특히 결론을 도출하기 위하여 수많은 조합의 수를 모두 탐색하는 것도 가능 하지만, 그 조합이 지수함수적으로 늘어나기 때문에 실제로는 실현 불가능하다. 예를 들어 Shannon이 추정한 수치를 보면 체스의 경우의 수를 보면 10120경우의 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 하므로써 2인용 보드게임에 대한 학습기능을 가지는 장기보드게임을 구현하는 것을 목표로 하고 있다.

2. 게임트리 탐색

2.1 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 단말노드에 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 Max와 Min의 이름을 부여하면 Max는 점수를 최대로 하는 수를 두게되고 반대로 Min은 점수를 최소화 해주는 수를 두게된다. 이러한 방식으로 트리의 모든 노드는 이득 값인 minimax값을 할당받는다. Max의 가장 우수한 수는 트리의 루트와 같은 minimax값을 가지는 수이다. 따라서 이와 같은 값을 가지는 값을 따라서 트리를 따라 내려 갈 경우에 이 경로가 각 선수에게는 최상의 경로를 나타내며 이를 Principal Variation이라고 한다.

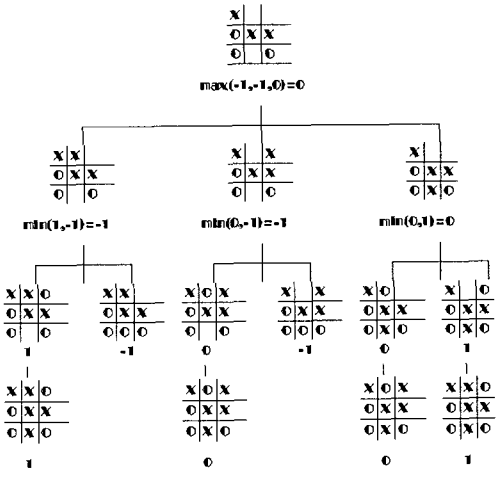


그림 1 Minimax값을 가지는 OX게임트리

Fig. 1 Naughts and crosses game tree with minimax values

2.2 알고리즘

Brudno에 의해서 처음으로 소개된 알고리즘은 minimax 알고리즘을 개량한 것이다. 트리의 각 노드마다 두 개의 한계 와 가 사용된다. 이 값은 깊이탐색에 따라서 인가된다. 각 노드에서 값은 가장 작은 값을 나타내며 트리의 상위노드들의 minimax값에 영향을 줄 수 있다. 반면에 값은 minimax값에 영향을 줄 수 있는 가장 큰 값을 나타낸다. 는 자기 자신의 노드를 포함하여 연결되어 있는 MAX노드들의

평가된 가지들중에서 가장 큰 minimax값을 나타낸다. MAX노드아래의 각 부 트리가 탐색되어 짐에 따라서 는 점차적으로 증가한다. 따라서 탐색경로를 따라 트리를 따라서 내려감에 따라 는 단조적으로 증가한다. 이와 유사하게 도 자기 자신의 노드를 포함하여 그 노드에 연결되어 있는 MIN노드에서 평가된 가지중에서 가장 작은 minmax값을 나타낸다. 가 보다 크거나 같은 위치에 이르면 트리의 루트에 가까운 우수한 경로가 있음을 알 수 있다.

```
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 알고리즘의 negamax구현

Fig 2 The negamax formulation of algorithm

$\alpha \geq \beta$인 노드아래는 더 이상 탐색 할 필요가 없고 바로 부모 노드로 복귀할 수 있다. 결과적으로 minimax값에 영향을 주지 않는 노드들을 삭제하게 된다. 알고리즘은 루트가 탐색창(- , +)으로 탐색이 되어진다면 올바른 minimax값이 반환할 수 있다. negamax알고리즘을 이용한 알고리즘이 그림 2에 나타나 있다. α와 β를 다음 레벨로 패스다운함에 따라서 한계가 항상 α에 유지될 수 있도록 두 개의 매개변수를 반전하고 교환한다. 이 알고리즘에서 트리의 짝수

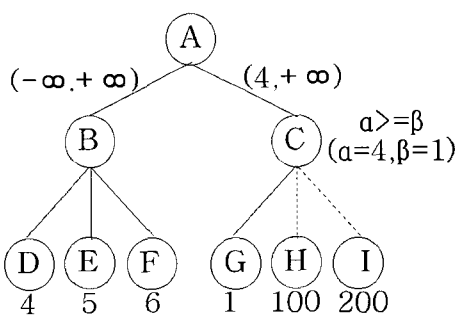


그림 3 의 얕은 차단

Fig 3 shallow cutoff

ply(max 노드)에서는 α가 상승하고, 홀수ply(min 노드)에서는 β가 감소하여 $\alpha\beta$알고리즘의 조건에 부합한다.

그림3은 shallow cutoff를 보여준다. 깊이 탐색에 의하여 트리를 좌에서 우로 진행해 가면 먼저 루트의 좌측가지를 탐색하게 되고 첫째의 수는 최대화측에게는 4의 minimax값을 발생한다. 따라서 우측의 가지를 탐색하게 되면 는 4로 고정이 되어 있는 상태에서 노드G는 1의 minimax값을 가지게 된다. 그리고 는 노드C에서 1로 감소된다. min측은 노드C의 1을 가지고 잇지만 max는 노드B로 이동하므로써 4의 값을 가지게 된다. 따라서 노드C의 다른 자식노드를 탐색할 필요가 없게 된다. max측은 노드C로 가는 것보다 노드B로 이동할 것이다. 따라서 노드H와 I는 탐색에서 제외될 것이다.

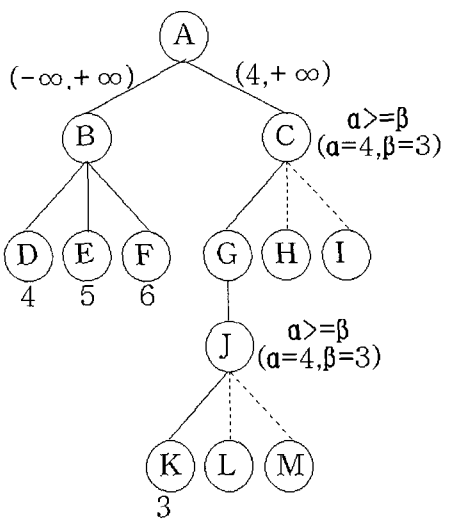


그림 4 의 깊은 차단

Fig 4 deep cutoff

알고리즘은 Deep Cutoff을 발생할 수 있다. Deep Cutoff은 깊이 d에서의 정보에 따라서 깊이 d보다 더 깊은 노드들을 탐색에서 제외시킬 수 있다. 예를 들어 그림4에서 노드B 아래의 부 트리를 탐색하므로써 얻어진 탐색창의 값에 따라서 노드J를 탐색에서 제외할 수 있다.

3. 인공신경회로망

3.1 역전파 알고리즘

역전파의 구조는 지도학습의 일종으로써 출력단의 오차

를 역방향으로 전파하여 다음의 전방향의 계산을 위하여 오차를 줄여 나가는 방식으로 그림5와 같다. 또한 다층의 구조이고 전방향이며 가중치의 연결은 다양한 형태를 취할 수 있다.

규정된 함수에 대한 입출력의 패턴을 학습하여 학습의 과정이 이루어 진다. 출력단의 값과 기대치와의 오차가 허용할 수 있는 범위에 들 때까지 가중치의 적응이 수행된다. 따라서 기존의 알고리즘에 준한다.

3.2 Temporal Difference 학습

Richard Sutton의 Temporal Difference의 학습은 일시적인 신용할당문제를 풀기 위한 좋은 방법을 제시하고 있다[7]. 이는 반복적으로 순차적인 판단의 과정을 관측하여 앞으로의 일을 예측하는 것이 목적이다.

$$E = \frac{1}{2}\sum_{p,j}(t_{jp} - y_{jp})^2$$

$$\triangle w_{ij} = a\sum_{p}((t_{jp} - y_{jp})\frac{\partial y_{jp}}{\partial w_{ij}}$$

$$net_j = \sum_i w_{ji}x_i$$

$$\frac{\partial E_j}{\partial net_j} = (t_{jp} - y_{jp})\frac{\partial y_{jp}}{\partial net_j}$$

$$\triangle w_{ij} = a\sum_{p}\frac{\partial E_j}{\partial net_j}x_i$$

$$\frac{\partial E_j}{\partial net_j} = \sum_k \frac{\partial E_k}{\partial net_k} w_{jk}\frac{\partial y_{jp}}{\partial net_j}$$

그림 5 역전파알고리즘
Fig 5 Backpropagation

이는 신경회로망과 같이 학습알고리즘은 아니지만, 지도학습과 같이 분류되는 알고리즘으로써 임의의 최종 상태 sf로 끝나는 일을 하는 동안에 일련의 상태(s1, s2, · · · ,sf)가 접근된다고 하자. 이러한 과정에서 각각 1이나 0인 이동도 uf로 보상이 이루어 질 수도 있고 이루어지지 않을 수도 있다. 각각의 상태를 최종적인 이동도 uf로 쌍을 이루어 (s1, uf),(s2, uf), · · · , (sf, uf) 과 같은 학습패턴을 만들 수 있다. 비교적으로 Temporal Difference방법은 Pt가 시간 t에서 제어기의 예측을 나타낼 경우에 학습패턴의 쌍을 (s1, Pf),(s2, Pf), · · · , (sf-1, Pf),(sf, uf) 로 사용할 수 있다. 이와 같은 쌍을 TD(0)모드라고 한다. 델타규칙을 지도학습으로 사용하면 갱신된 가중치는 식(3.1)과 같다.

$$\triangle w_t = a(P_{t+1} - P_t)\frac{\partial P_t}{\partial w} \qquad (3.1)$$

최종적인 출력에 대해 직접적으로 각각의 상태에 대한 예측을 하기 전에 바로 다음의 단계에서 발생되는 예측에 대해 가중치가 갱신되어 진다. 따라서 일시적으로 연속적인 예측들간의 차이를 최소화시키기 위한 것이 목적이다. Sutton이 제안한 아이디어는 완전히 새로운 것은 아니지만 이전의 학습알고리즘들의 Temporal Difference 방법을 변형한 것이다.

3.3 장기의 구조

본 논문의 프로그램은 크게 두 개의 부분으로 그림6에서와 같이 구성되어 있다. 첫째부분은 탐색과 학습으로 구성되어 있고 다른 부분은 적법한 수를 발생시키는 부분이다.

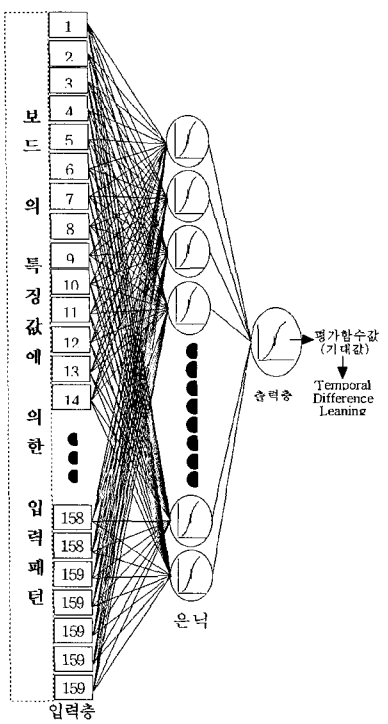


그림 6 평가함수의 학습을 위한 역전파알고리즘
Fig 6 Backpropagation for evaluation function

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, 수 두기는 일단 수가 두어지면 보드의 위치를 갱신한다. 마지막으로 게임 종료는 게임의 승패와 비김을 나타낸다. 그림 3.2는 신경회로망을 이용한 장기의 평가함수의 학습을 위한 네트워크를 보여준다.

본 논문에서는 두 개의 평가함수를 사용하는데 하나는 먼저 두는 선수에 대한 평가함수이고 다른 하나는 두 번째로 두는 선수의 것이다. 첫 번째 선수가 움직일 수 있는 수들을 이용하여 첫 번째 선수의 평가함수를 학습하고 두 번째 선수의 평가함수도 그 선수의 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. 평가함수의 학습의 과정에 탐색의 과정을 추가하여 역전파 신경회로망이 가지는 국부해의 단점을 보완하였다.

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

표 1 특징 값의 구성

Table 1 The formation of feature vectors

게임의 유효한 특징 값들을 발견하기 위하여 평가함수 학습알고리즘을 이용하는 데에는 실제로 게임의 학습에 있어서 아주 많은 학습패턴이 필요하다. 이러한 이유로 표1과 같이 특징 값들의 항목을 이용하여 학습을 수행하였다.

게임을 하는 동안에 수행되는 각각의 수들은 신경회로망을 구성하는 평가함수에 대한 각각의 입력패턴을 형성한다. TD알고리즘은 각각의 패턴에 대한 기대치(target value)로 작용한다. 값은 0으로 바로 인접한 상태만을 고려하도록 하여 다음 보드의 위치에 대한 평가 값이 현재의 보드에 대한 평가 값의 기대치로 작용하도록 하여 역전파 신경회로망의 학습이 이루어진다. 따라서 역전파 알고리즘은 각각의 패턴에 대하여 네트워크의 가중치를 적응시키므로써 지능을 가지게 된다.

4. 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 그림7과 같이 두 프로세스간에 서로 수를 주고 받으며 그림과 같이 게임을 진행하였다. 학습과정에서 적군이 이

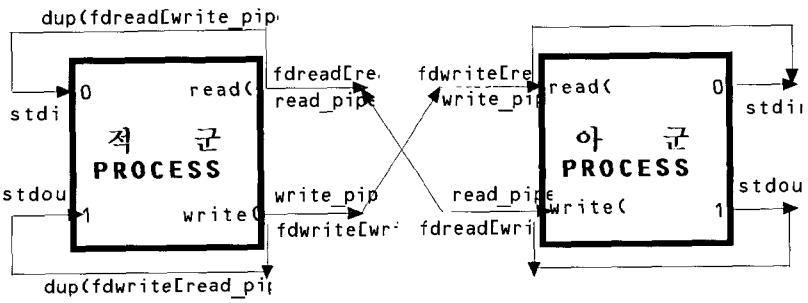


그림7 인터페이스 쉘

Fig. 7 Interface shell

기기 까지 아군이 둔 수는 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. 그림8은 두 프로세스가 서로 메시지를 주고 받기 위한 보드이다. 보드에서 각각의 기물의 값은 졸이 1, 마가 5, 상이 3, 포가 7, 차가 13, 사가 3 그리고 왕은 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.

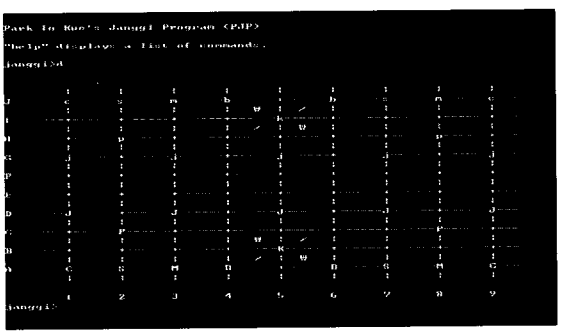


그림 8 초기의 보드위치

Fig. 8 The initial board position

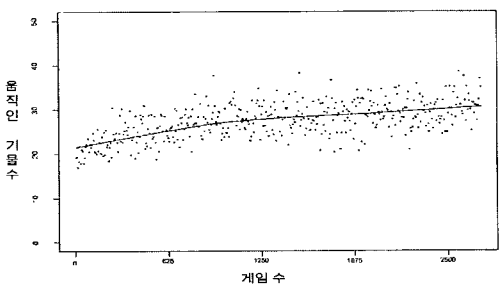


그림 9 게임종료전 장기의 잡은 기물수

Fig. 9 The number of points JANGGI captures before the game is over

2500번의 학습을 수행한 아군의 컴퓨터와 적군(4 ply중급)의 또 다른 컴퓨터에 대하여 사람이 마우스를 이용하여 상호의 수를 두어 50번의 게임을 수행한 결과, 41번을 이기고 9번을 졌다. 이는 보다 많은 학습을 통하여 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 알고리즘등을 이용할 수도 있다.

그림 9과 10은 적군에 대하여 게임을 하는 동안에 10게임 단위로 아군이 둔 수의 개수와 잡은 기물의 수에 대한 분포도이다.

5. 결론

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

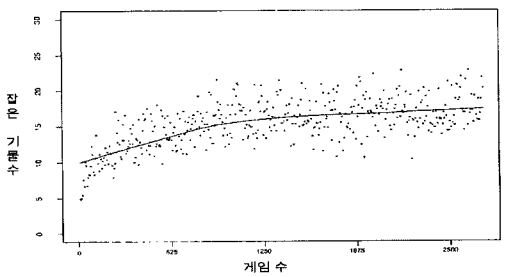


그림 10 게임종료전 장기의 움직인 기물수

Fig. 10 The number of moves JANGGI makes before the game is over

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

참고문헌

[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.

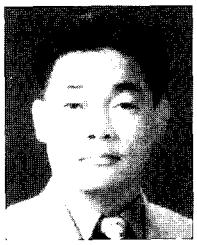
[3] Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational. abilities. In Proceedings of the National Academy of Sciences USA, volume 79, pp 2554-2558.

[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.



박인규

e-mail : ikpark@joongbu.ac.kr
1985년 원광대학교 전기공학과 졸업(공학사)
1987년 연세대학교 대학원 전기공학과 전자계산기응용(공학석사)
1996년 원광대학교 대학원 전자공학과 마이크로프로세서응용(공학박사)
1997년 - 현재 중부대학교 정보공학부 전자계산학과 조교수
관심분야 : 퍼지논리, 신경회로망, 최적화 보드게임이론등

정광호

서울산업대학교 전산기기전공(공학사)
건국대학교 산업대학원 컴퓨터응용(공학석사)
동국대학교 대학원 전산통계학전공(이학박사)
육군통신학교 마이크로웨이브교관(예비역 대위)
중부대학교 전자계산소장,학생처장,사회교육원장등
중부대학교 전자계산학전공 전임강사,조교수,부교수
중부대학교 컴퓨터공학부 게임공학전공 부교수
중부대학교 대학원장(현)
관심분야 : 게임공학, 소프트웨어공학