유전자기반 신경회로망과 Temporal Difference학습: 장기보드게임

박인규
중부대학교 컴퓨터과학과

Genetic Algorithm based Neural Network and Temporal Difference Learning: Janggi Board Game

In-Kue Park
Dept. of Computer Science, JoongBu University
E-mail : ikpark@joongbu.ac.kr

요 약

본 논문은 2인용 보드게임의 정보에 대한 전략을 학습할 수 있는 방법을 유전자기반 역전파 신경회로망과 Temporal Difference학습알고리즘을 이용하여 제안하였다. 학습의 과정은 역전파에 의한 초기학습에 이어 국부해의 단점을 극복하기 위하여 미세학습으로 유전자알고리즘을 이용하였다. 시스템의 구성은 탐색을 담당하는 부분과 기물의 수를 발생하는 부분으로 구성되어 있다. 수의 발생부분은 보드의 상태에 따라서 갱신되고, 탐색커널은 αβ탐색을 기본으로 유전자알고리즘을 이용하여 가중치를 최적화하는 유전자기반 역전파 신경회로망과 TD학습을 결합하여 게임에 대해 양호한 평가함수를 학습하였다. 일반적으로 많은 학습을 통하여 평가함수의 정확도가 보장되면 승률이 학습의 양에 비례함을 알 수 있었다.

1. 서론

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. 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 가

리는데 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.

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.

2 게임트리 탐색

Brudno에 의해서 처음으로 소개된 αβ알고리즘은 최대최소알고리즘을 개량한 것이다. 트리의 노드마다 두 개의 한계 α와 β가 사용된다. 이 값은 깊이탐색에 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.

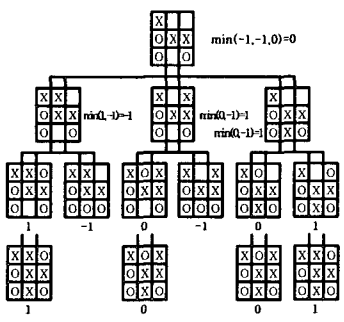

그림 1 OX게임트리

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 탐색 할 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.

```
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

negamax알고리즘을 이용한 αβ알고리즘이 그림2에 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.

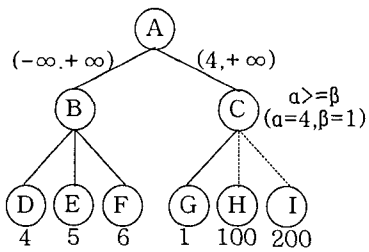

그림 3 αβ 의 얕은 차단

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 α

는 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.

3 인공신경회로망

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

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.

이는 신경회로망과 같이 학습알고리즘은 아니지만, 지도학습과 같이 분류되는 알고리즘으로써 임의의 최종 상태 s_f로 끝나는 일을 하는 동안에 일련의 상태(s_1, s_2, · · ·,s_f)가 접근된다고 하자. 이러한 과정에서 각각 1이나 0인 이동도 u_f로 보상이 이루어 질 수도 있고 이루어지지 않을 수도 있다. 각각의 상태를 최종적인 이동도 u_f로 쌍을 이루어 {(s_1, u_f),(s_2, u_f), · · ·, (s_f, u_f)}과 같은 학습패턴을 만들 수 있다. 일반적으로 Temporal Difference방법은 P_t가 시간 t에서 제어기의 예측을 나타낼 경우에 학습패턴의 쌍을 {(s_1, P_f),(s_2, P_f), · · ·, (s_f-1, P_f),(s_f, u_f)}로 사용할 수 있다. 이와 같은 쌍을 TD(0)모드라고 한다. 델타규칙을 지도학습으로 사용하면 갱신된 가중치는 식(1)과 같다.

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

최종적인 출력에 대해 직접적으로 각각의 상태에 대한 예측을 하기 전에 바로 다음의 단계에서 발생되는 예측에 대해 가중치가 갱신되어진다. 따라서 일시적으로 연속적인 예측들간의 차이를 최소화시키기 위한 것이 목적이다.

3.3 유전자 알고리즘

유전자 알고리즘은 자연세계의 진화과정을 시뮬레이션 함으로써 복잡한 실세계의 문제를 해결하고자 하는 계산 모델로 구조가 간단하고 방법이 일반적이어서 응용범위가 매우 넓으며, 특히 적응적 탐색과 학습 및 최적화를 통한 공학적인 문제의 해결에 많이 응용되고 있다. 유전자 알고리즘은 특정 문제에 대한 가능한 해를 간단한 염색체와 같은 자료구조로 표현하고 그 해에 근접하도록 선택(selection), 재조합(recombination), 변이(mutation)와 같은 연산자들을 적용하는 것이다. 이 모집단에 선택 연산(selection), 재조합 연산(recombination), 변이 연산(mutation)을 적용하여 다음 모집단(P(t))을 생성한다. 현재 모집단에서 다음 모집단까지의 이 과정을 한 세대라고 하며 다음 세대에서는 이 모집단이 다시 현재 모집단이 되어 동일한 과정을 반복한다. 이러한 반복과정은 생성된 모집단이 문제에 대한 해에 수렴할 때 종료하게 된다.

```
Genetic Algorithm():
{
        t <- 0;
        Initialize(P(t));
        Evaluate(P(t));
        while(not Finished()){
                t <- t+1;
                Select P(t) from P(t-1);
                Recombine P(t);
                Mutate P(t);
                Evaluate(P(t));
        }
        solution <- BestOf(P(t));
}
```

4 장기의 구조

본 논문의 시스템은 그림7에서와 같이 구성되어 있다. 게임의 규칙은 수 발생기에 의하여 발생된다. 따라서 수 발생기, 수 두기와 게임의 종료에 의하여 게임을 진행하는데, 수 발생기

는 임의의 주어진 위치에 대하여 기물이 움직일 수 있는 모든 행마를 발생시키며, 수두기는 일단 수가 두어지면 보드의 위치를 갱신한다. 마지막으로 게임종료는 게임의 승패와 비김을 나타낸다.

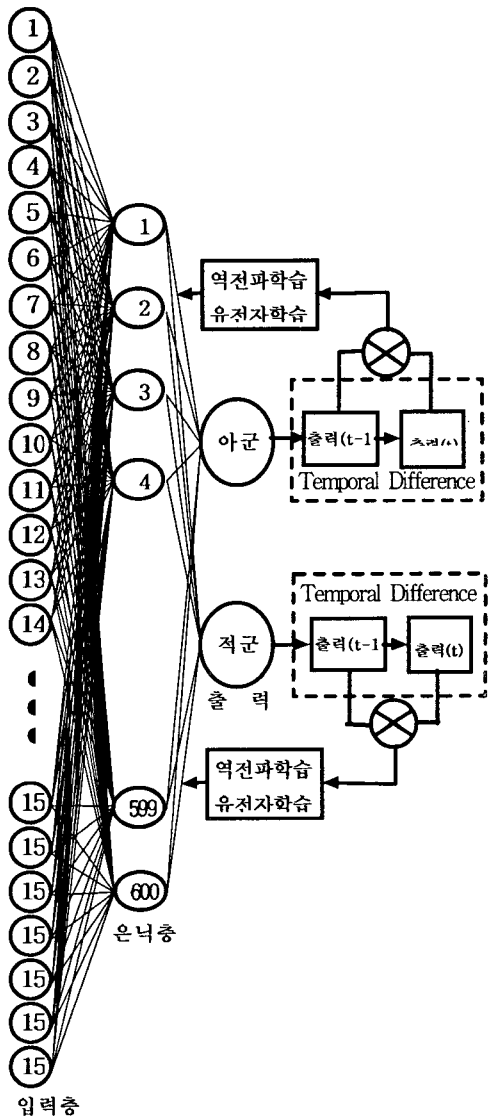

그림 7 장기의 구조

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

```
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

게임의 유효한 특징 값들을 발견하기 위하여 평가함수 학습알고리즘을 이용하는 데에는 실제로 게임의 학습에 있어서 아주 많은 학습 패턴이 필요하다. 따라서 표1과 같이 특징 값들의 항목을 추출하여 학습을 수행하였다.

표 1 특징 값의 구성

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.

게임을 하는 동안에 수행되는 각각의 수들은 신경회로망을 구성하는 평가함수에 대한 각각의 입력패턴을 형성한다.

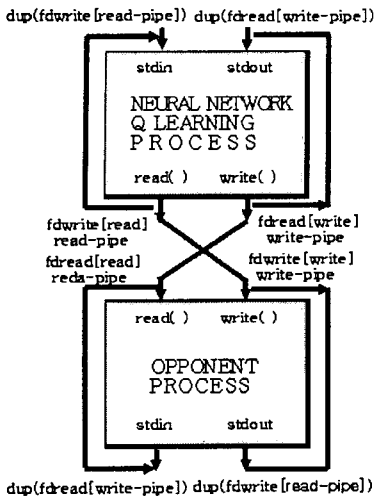

그림9 학습을 위한 인터페이스 쉘

TD학습에 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 fork함수와 pipe를 이용하여 양방향 통신이 가능하도록 하여 구성하였다.

4 결과고찰

본 논문에서 제안한 장기와의 학습을 위하여 신경회로망이 없는 일반적인 장기프로그램(적군 프로세스)을 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 그림 과 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. 그림10은 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.

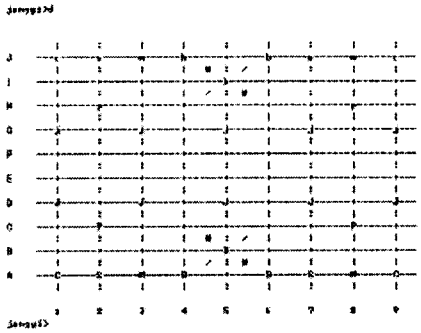

그림 10 초기의 보드위치

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. 그림11은 2인용 보드게임의 탐색은 최소최대(minimax) 탐색알고리즘을 이용하기 때문에 아군의 경우에는 최대노드에 해당하고, 적군의 경우에는 최소노드에 해당한다. 따라서 아군의 경우는 양의 값을 가지고 적군의 경우는 음의 값을 가진다. 학습의 양이 많아질수록 평가함수의 값이

아군의 경우에는 커지게 되고 적군의 경우에는 작아지게 된다. 그림12와 그림13은 아군과 적군의 평균자승오차를 나타낸다. 그림14는 학습과 승률과의 관계를 알아보기 위하여 탐색의 종류를 일치탐색, 1ply탐색과 2ply탐색의 세가지로 구분하여 실험을 수행하였다. 학습의 양이 증가함에 따라서 승률이 증가함을 알았다.

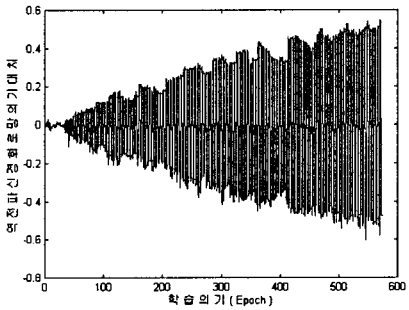

그림 11 역전파 신경회로망의 기대치

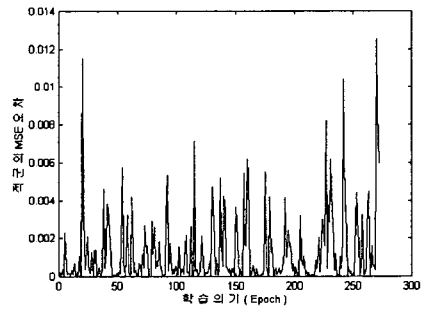

그림 12 적군의 MSE오차

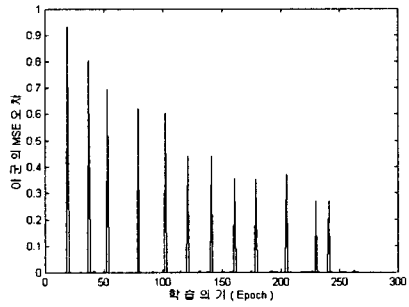

그림 13 아군의 MSE오차

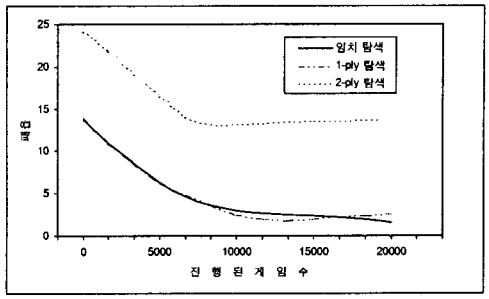

그림 14 학습과 승률과의 관계

5 결론

본 논문에서는 TD학습 알고리즘과 유전자기반 역전파 신경회로망을 이용하여 평가함수에 대한 학습을 통하여 지능을 가지는 장기보드게임을 개발하였다. 역전파의 국부해의 약점을 극복하기 위하여 초기에는 역전파에 의한 학습이 진행된 다음에 미세학습의 과정으로 유전자에 의한 학습을 이용하여 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.

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