ISSN 1975-8359(Print) / ISSN 2287-4364(Online)
The Transactions of the Korean Institute of Electrical Engineers Vol. 67, No. 12, pp. 1657~1664, 2018
http://doi.org/10.5370/KIEE.2018.67.12.1657

ANN 기반 기보학습 및 Minimax 탐색 알고리즘을 이용한 오델로 게임 플레이어의 구현

An Implementation of Othello Game Player Using ANN based Records Learning and Minimax Search Algorithm

전 영 진* · 조 영 완†
(Youngjin Jeon · Youngwan Cho)

Abstract - This paper proposes a decision making scheme for choosing the best move at each state of game in order to implement an artificial intelligence othello game player. The proposed decision making scheme predicts the various possible states of the game when the game has progressed from the current state, evaluates the degree of possibility of winning or losing the game at the states, and searches the best move based on the evaluation. In this paper, we generate learning data by decomposing the records of professional players' real game into states, matching and accumulating winning points to the states, and using the Artificial Neural Network that learned them, we evaluated the value of each predicted state and applied the Minimax search to determine the best move. We implemented an artificial intelligence player of the Othello game by applying the proposed scheme and evaluated the performance of the game player through games with three different artificial intelligence players.

Key Words : Othello game, Minimax search, Artificial neural network, Decision making, Records learning

1. 서 론

최근 복잡한 상황에 대한 최선의 의사결정을 위하여 딥러닝(Deep Learning), 강화학습 등과 같은 기계학습에 대한 연구가 활발히 이루어지고 있다. 그 중 딥러닝은 방대한 데이터의 학습을 토대로 인식, 분류, 게임 등의 다양한 분야에 활용되고 있으며 Google, Facebook, Amazon 등 세계적인 기업들은 지속적인 연구를 통해 실용적인 서비스를 제공하고 있고 몇몇 분야에서는 인간이 하기 어려운 일을 대신하고자 하는 시도를 하고 있다.

딥러닝을 적용한 인공지능은 최근 바둑과 같이 고도의 계산 및 판단 능력을 요하는 지능적 보드 게임에 성공적으로 적용되며 그 응용의 실용성이 확인되었다. AlphaGo-Lee는 2016년 세계 최정상급 프로기사 이세돌을 상대로 4:1로 승리하며 화제가 되었고 이후 여러 번의 개선을 통해 AlphaGo-Zero로 발전하였으며 AlphaGo-Lee와의 대국에서 100%의 승률을 보이며 딥러닝 및 강화학습이 적용된 인공지능의 발전가능성을 보여주었다[1][2].

지능적 보드 게임은 전통적으로 인공지능이 도전하는 대표적인 분야로 바둑 이외에도 체스, 오델로 게임 분야에서 딥블루(Deep Blue)[3], 로지스텔로(LOGISTELLO)[4] 등의 인공지능 알고리즘이 세계 챔피언들을 상대로 승리하였다. 오델로 게임 또한 가능한 모든 상황의 수가 대략 10^{54}개로 바둑과 체스 못지않게 많은 경우의 수가 존재하며 아직까지 완전한 솔루션이 제시되지 않아 인공지능 분야의 도전 대상으로 꾸준히 많은 연구가 진행되고 있다.

IAGO[5]는 최초의 프로기사 수준의 오델로 인공지능 플레이어로 상황에 따른 각기 다른 전략과 가치평가함수를 사용하여 의사결정을 수행하나 계산 기반 가치평가함수를 사용하여 예상하지 못한 상황이나 추세의 변화에 약한 모습을 보이는 단점을 갖는다. BILL[6]은 IAGO와 비슷한 전략의 가치평가함수를 가지는 알고리즘으로 학습을 통해 구한 가치평가함수를 사용함으로써 IAGO의 한계를 극복했다는 평가를 받는다. 또한 Logistello[4]는 분류된 패턴에 대해 선형회귀를 적용하여 가치평가함수를 결정하고 Zebra[7]는 가치평가함수의 해시 테이블(Hash table)을 기보를 통해 학습하여 적용하였다.

Logistello와 Zebra가 우수한 성능을 가지기는 하나 모두 테이블 형식의 가치평가함수를 사용하고 있어 가능한 모든 상황에 대한 평가 테이블을 구성하기는 어려우므로 한계를 지니고 있다고 할 수 있다. 이와 같은 문제를 해결하기 위해 가치평가 함수를 테이블로 구성하지 않고 인공신경망으로 근사하여 표현하거나[8, 9] 강화학습(Reinforcement Learning)이나 딥강화학습(Deep Reinforcement Learning) 같은 자가 학습방법을 이용해 의사결정을 하려는 연구 또한 최근 활발히 진행되고 있다[10-12].
본 논문에서는 오델로 게임의 인공지능 플레이어 구현을 위해 게

† Corresponding Author : Dept. of Computer Eng., Seokyeong University, Korea.
E-mail: ywcho@skuniv.ac.kr
* Dept. of Computer Eng., Seokyeong University, Korea.
Received : November 6, 2018; Accepted : November 14, 2018

임의 각 진행 상태에서 최선의 수를 선택하기 위한 의사결정 방법을 제안한다. 제안하는 의사결정 방법은 현 상태에서 게임이 몇 단계 진행되었을 경우의 여러 가능한 상태를 예측하고 이러한 상태들에 대해 승패의 유·불리에 대한 평가를 수행하며 이 평가를 기반으로 최선의 수를 탐색하여 결정한다. 본 논문에서는 프로기사들의 실제 기보를 상태별로 분해하고 가치를 부여하여 누적함으로써 학습 데이터를 생성하였고 이를 인공신경망(Artificial Neural Network)을 통해 학습함으로써 모든 가능한 경우에 대해 가치평가함수를 구성할 수 있는 방법을 제안하였으며 이를 통해 예측된 각 상태에 대해 가치 평가를 수행하고 이를 바탕으로 최선의 수를 결정하기 위하여 최대최소 탐색을 적용하였다.

2. ANN 기반 기보학습

본 논문에서는 오델로 게임의 각 진행 상황에서 형세 판단을 위해 프로 기사들의 기보를 ANN을 통해서 학습하고 이 결과를 탐색에 적용하여 현 상황에서 최적의 수를 결정하는데 이용한다. 학습시키고자 하는 신경망은 오델로 게임의 특정 상황을 입력으로 하며 이에 대해 형세(승패의 예측 확률)를 출력하는 구조이다. 신경망의 학습을 위해 하나의 기보는 진행되는 각 수별로 상황을 분해하여 각 진행 상황에 대해 승패의 결과를 1(승)과 0(패)로 부여하며 전체 기보 데이터에 대해 이를 누적함으로써 수많은 다양한 상황에 대한 학습데이터를 구성하여 사용하는 방법을 제안하여 이용한다.

2.1 ANN 학습데이터 생성 및 지도학습

오델로, 바둑, 장기, 체스와 같은 지능적 보드 게임에서 플레이어는 게임의 각 진행 단계에서 최선의 선택을 위해 게임이 몇 단계 진행되었을 경우의 상황을 예측하고 이러한 가능성 있는 여러 가지 예측 상황에서 승패의 유·불리(형세)에 대한 평가를 수행하여 이를 바탕으로 최선의 수를 결정하게 된다. 본 논문에서는 게임이 진행되었을 경우 발생할 수 있는 여러 가지 예측 상황에서의 형세를 평가하는 모델로서 인공신경망을 사용하고 이를 바탕으로 최선의 수를 결정하기 위해 탐색 알고리즘을 적용한다.

본 논문에서 제안하여 사용하는 인공신경망은 예측된 특정 게임 상황이 주어졌을 경우 이에 대한 형세를 평가하여 제공하는 역할을 하므로 게임의 특정 상황을 입력으로 하고 형세를 나타내는 척도로서 승패의 유·불리에 대한 확률을 학습결과로 출력하는 구조를 갖는다. 이를 위해 본 논문에서는 게임이 진행되는 각 상황을 상태 s로 정의하고 상태 s에 대한 승패의 유·불리 정도를 가치함수 $v(s)$로 정의하여 사용한다.

그림 1은 게임의 특정 상황을 상태 s로 표현하는 과정을 나타낸다. 본 논문에서는 8×8 크기의 보드에서 이루어지는 오델로 게임을 대상으로 하고 있으며 게임의 상태 s는 8×8 공간의 각 위치에 부여된 64개의 원소 e로 표현된다. 게임의 상황은 8×8 공간에서 흑돌, 백돌, 돌이 놓이지 않은 곳의 상대적 위치에 의해 결정되므로 이는 상태 원소 e에 돌이 없는 곳은 0, 흑돌이 놓인 곳은 1, 백돌이 놓인 곳은 2의 값을 부여하여 그림 1에서와 같이 행렬 형태로 표현할 수 있고 이는 1차원 벡터 형식으로 변환하여 인공신경망의 입력으로 사용한다.

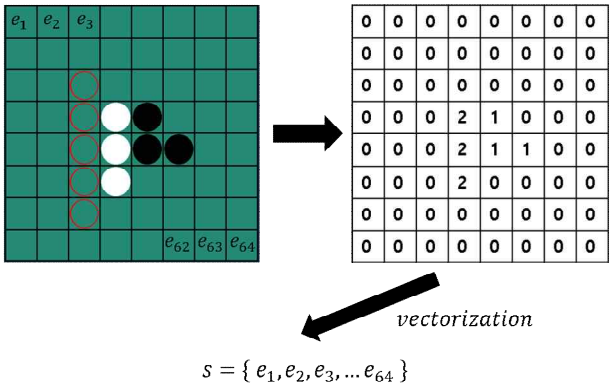

그림 1 상태의 표현
Fig. 1 Representation of the state

본 논문에서는 인공신경망의 학습을 위한 학습 데이터로 프로 기사들의 실제 대국을 기록한 기보를 사용하였다. 본 논문에서 사용한 기보 데이터는 1977년부터 2018년까지 진행된 모든 세계대회에서의 프로기사 대국의 기록으로 약 15만개의 기보를 사용하였으며 기보의 두 가지 예와 그 구성을 제시하면 표 1과 같다.

표 1 학습에 사용된 기보의 구성
Table 1 Configuration of records used for learning

기 보	결 과
F5F6E6F4F3D6C6D3G6E7G5D7C7H5C3 H6E3B6C5D8C8B8A6B5G4D2E1C1F2H4 A5E2F8B3F7G1H3H2G3A4C4B4C2B1A3 A7A8E8B7B2A1A2D1G8H8G2G7H7F1H1	흑 승
F5F6E6F4F3C5G6E3D3G5B6C4C3D6H6 G4H3D2E2H5H4C2E1D1C1B3E7C6G3D7 C8F1F2B1B4A5C7B5A6F8A4A3A2D8E8 B8B2G1G2A1B7A7A8F7G8H1G7H2H7H8	백 승

기보는 대국의 진행에 따라 돌이 놓여진 위치를 순서대로 기록한 것으로 오델로 보드의 가로축 위치를 나타내는 알파벳과 세로축 위치를 나타내는 숫자의 조합을 한 쌍으로 하여 진행된 수에 따라 순차적으로 나열한 것이다. 대국은 기본형에서 시작하여 최대 60수까지 진행되므로 기보는 최대 120개의 문자열로 구성되어 있다.

본 논문에서 사용하는 인공신경망은 게임의 상태 s에 대해 형세를 나타내는 가치함수 $v(s)$를 학습결과로 출력하는 구조이므로 기보를 그 자체로서 그대로 학습데이터로 사용하지 않고 게임의 진행 단계에 따라 상태별로 분해하고 승패에 대한 값을 부여하며 이러한 데이터를 전체 기보에 대해 통합하여 각 상태별 학습데이터를 생성하여 학습한다.

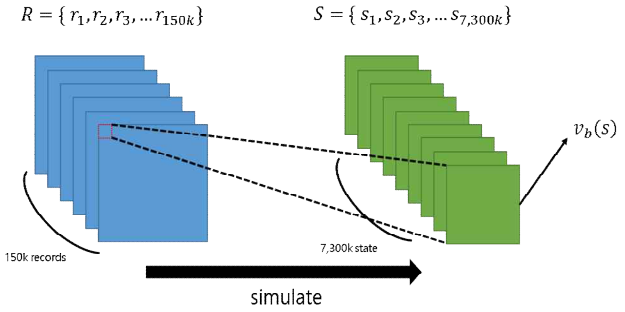

그림 2 학습데이터 생성과정
Fig. 2 Process of learning data generation

그림 2는 본 논문에 적용된 학습데이터의 생성과정을 나타낸다. 약 15만개의 기보로 이루어진 집합 R을 상태에 따라 분해하고 조합하여 약 730만개의 상태 s에 대한 학습 데이터의 집합 S를 구성하여 사용하였다.

상태별로 분류된 각각의 학습데이터를 사용하여 상태 s에 대한 승패의 기보 확률 $v_b(s)$는 다음 식 (1)과 (2)와 같이 구한다.

$$z_k(s)=\begin{cases}1, win\\0, lose\end{cases} \tag{1}$$

$$v_b(s)=\frac{1}{n}\sum_{k=1}^{n}z_k(s) \tag{2}$$

식 (1)에서 z는 승패여부를 판단하여 누적하기 위한 값으로 흑을 기준으로 승: $z_k(s)=1$, 패: $z_k(s)=0$로 설정하여 사용하였다. 식 (2)에서 n은 상태 s가 기보에서 등장한 총 회수를 나타낸다. 따라서 상태 s에 대한 승패의 기보 확률 $v_b(s)$는 기보에서 등장한 상태 s에 대해 흑을 기준으로 승리한 경우의 확률을 의미하므로 상태 s에 대한 승패의 유·불리 즉, 형세를 나타내는 척도로 사용하였다.

본 논문에서 사용한 기보 데이터들을 분석해 본 결과 비기는 경우는 많지 않았기 때문에 학습에서 제외시켰으며 게임 플레이어가 백일 때의 상태 가치 함수 $v_w(s)$는 다음 식 (3)과 같이 구할 수 있다.

$$v_w(s)=1-v_b(s) \tag{3}$$

표 2는 기보 데이터를 통해 구한 상태별 빈도, 승리 회수, 가치함수에 대한 몇 가지 예를 나타낸다. 표 2의 상단의 4개 상태들은 오델로 게임의 진행 초반을 나타내는 것으로 출현빈도 (n)가 높고 승패의 기보 확률 $v(s)$(흑일 경우 $v_b(s)$, 백일 경우 $v_w(s)$)가 0.5에 가까운 것을 알 수 있고 하단의 3개 상태들은 비교적 초기 진행 상태에서 흑의 승률이 높았던 상태들을 보여주고 있다.

오델로 게임 플레이어는 게임 진행의 각 단계에서 최선의 선택을 위해 탐색하는 과정에서 탐색의 말단 노드에서의 형세 즉, 상태에 대한 가치함수의 값을 기반으로 최적의 경우를 선택하게

표 2 기보로 생성한 가치평가 테이블.
Table 2 Evaluation table generated from game records

State vector	n	win	$v(s)$
0000000000000000000000000021000 0001110000000000000000000000000	122250	55336	0.452
0000000000000000000000000021000 0001210000000200000000000000000	41221	19155	0.465
0000000000000000000000000021000 0001110000001200000000000000000	40498	18941	0.478
0000000000000000000000000022200 0001120000001200000000000000000	40498	18941	0.478
0000000000000000000000000022200 0001110000000000000000000000000	1384	864	0.624
0000000000000000000010000021200 0001110000000000000000000000000	1241	803	0.647
0000000000000000000000000021000 0011210000020200000000000000000	335	257	0.767

된다. 이러한 경우 가능한 모든 상태에 대해 가치함수가 정의되어야 탐색이 가능하며 가치함수의 정확도는 플레이어의 게임 성능을 결정하는 주요한 요소가 된다.

기보 데이터를 통해 표 2와 같이 상태에 대해 일대일 대응이 되는 가치 평가 테이블을 구성할 수는 있으나 8×8 공간을 가지는 오델로 게임의 모든 가능한 상태의 수는 대략 10^{54} 개로 모든 상태에 대한 가치 함수를 테이블로 표현하여 사용하기는 어렵다. 따라서 본 논문에서는 기보 데이터를 표 2와 같은 형태의 학습 데이터로 표현하여 이를 토대로 인공신경망을 학습하여 가치함수를 근사한다.

본 논문에서 가치평가함수를 근사하기 위해 사용한 인공신경망의 구성은 그림 3과 같다. 은닉층은 총 3개 계층으로 구성되어 있고 각각 128, 256, 64개의 노드를 가지며 활성화 함수는 모두 ReLu 함수를 사용하였다. 출력층은 가치 함수를 출력하는 1개의 노드로 구성되어 있고 0~1사이의 값을 가지는 승률을 표현하기 때문에 sigmoid 함수를 활성화 함수로 사용하였다.

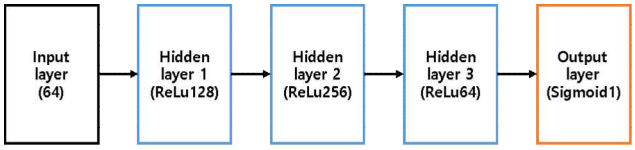

그림 3 학습에 사용된 인공신경망의 구조
Fig. 3 Structure of Artificial Neural Network used in Supervised-Learning

ANN 학습과정에서의 비용함수(Cost Function)는 식 (4)와 같이 전체 학습 데이터 개수 N에 대한 상태 s의 상대적 출현 빈도 $n(s)$를 가중한 MSE(Mean Squared Error)를 사용하였고 이를 통해 인공신경망의 가치평가함수 $v_\theta(s)$가 기보를 통해 구성

한 가치평가 테이블 $v(s)$를 근사할 수 있도록 학습하였다.

$$C(\theta)=\frac{1}{2M}\sum_{s}\frac{n(s)}{N}(v(s)-v_{\theta}(s))^2 \qquad (4)$$

3. ANN 및 탐색 기반 의사결정

본 논문에서는 인공지능 오델로 게임 플레이어의 구현을 위해 제안한 인공신경망의 학습을 통해 구한 가치평가함수 $v_{\theta}(s)$를 최소최대탐색 알고리즘에 적용하여 게임 플레이어의 의사를 결정한다.

3.1 최소최대탐색 알고리즘

최소최대탐색 알고리즘은 예상되는 최대의 손실을 최소화하기 위한 의사결정 방법으로써 바둑이나 오델로 게임과 같이 1:1로 상대방과 상호작용을 하는 경우에 유효한 탐색 방식으로 알려져 있다.[15]

최소최대탐색 알고리즘은 상대방과 자신이 항상 최선의 수를 둔다는 가정 하에 진행되며 기본적으로 트리탐색의 구조를 가진다. 본 논문에서는 트리를 하나의 상태에서 선택 가능한 경우의 수들로 확장하는데 확장은 트리의 깊이(depth)까지 진행되며 말단 노드(leaf node)에 도달했을 때 해당 노드의 상태에 대한 가치평가를 하고 상위 노드(parent node)로 그 값을 반환한다. 그 상위 노드가 자신의 차례라면 하위 노드(child node) 중 최댓값을 취하고, 상대방의 차례라면 하위 노드 중 최솟값을 취해 그 노드의 상위 노드로 반환하게 된다. 이 과정을 통해 전체 트리의 평가를 하게 되고 평가를 끝마친 트리의 루트 노드(root node) 아래 하위 노드 중 최대가 되는 노드가 최종적인 의사결정으로 선택된다.

그림 4는 최소최대탐색 알고리즘의 의사코드이다. 재귀형태의 함수로 구성되어 있으며 첫 번째로 호출할 함수의 파라미터 깊이(depth)에 따라 앞으로 몇 수를 내다볼지를 설정한다. 최대화 행위자(maximizing player)는 최대(최선의 수)를, 최소화 행위자(minimizing player)는 최소(최대화 행위자 입장에서는 최악의 수)를 선택함으로써 깊이에서 설정된 수까지 진행되었을 경우를 고려하여 최선의 수를 선택할 수 있도록 한다.

그림 5는 최소최대탐색 알고리즘의 선택과정을 예시로 나타낸 것이다. 그림에서 원은 최대화할 노드(자신)를 나타내고 사각형은 최소화할 노드(상대방)를 나타낸다. 그림의 예에서 지역적 트리에서의 선택은 파선으로 표시하였으며 알고리즘을 통해 최종적으로 선택된 현재 트리의 출력은 점선으로 표시하였다.

```
function minimax(node, depth, maximizingPlayer) is
    if depth = 0 or node is a terminal node then
        return the heuristic value of node
    if maximizingPlayer then
        value := -∞
        for each child of node do
            value := max(value, minimax(child, depth -1, FALSE))
        return value
    else (*minimizing player *)
        value := ∞
        for each child of node do
            value := min(value, minimax(child, depth -1, TRUE))
        return value
```

그림 4 최소최대탐색 알고리즘 의사코드

Fig. 4 Pseudo code of Minimax Search

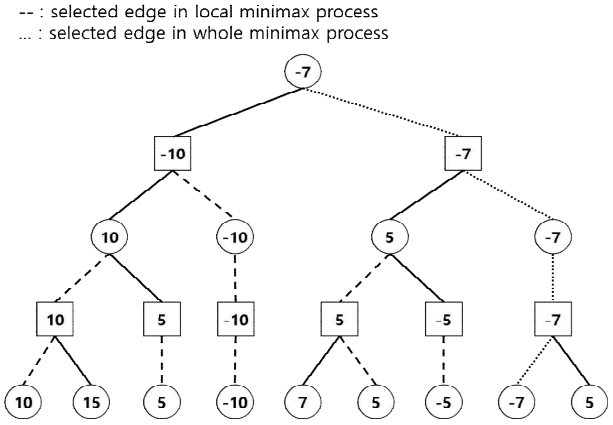

그림 5 최소최대탐색 알고리즘의 선택 과정

Fig. 5 Selection process of Minimax search algorithm

최소최대탐색은 루트 노드에서 발생 가능한 모든 경우에 대해 탐색을 시행하기 때문에 깊이(depth)가 커짐에 따라 연산 속도가 기하급수적으로 느려지게 문제점이 있다. 이에 본 논문에서는 최소최대탐색의 연산 속도를 개선하기 위해 Alpha-Beta Pruning을 사용하여 의사결정을 수행한다.

3.2 Alpha-Beta Pruning

최소최대탐색은 깊이(depth) 내의 가능한 모든 경우를 탐색하므로 탐색이 필요하지 않은 경우를 포함하는 경우 시간복잡도가 높다. Alpha-Beta Pruning은 가지치기를 통해 불필요한 부분은 탐색을 하지 않으면서도 최소최대탐색과 같은 결과를 가진다.[16]

```
function alphabeta(node, depth, maximizingPlayer, α, β) is
    if depth = 0 or node is a terminal node then
        return the heuristic value of node
    if maximizingPlayer then
        value := -∞
        for each child of node do
            value := max(value, alphabeta(child, depth -1, FALSE , α, β))
            α := max(α, value)
            if β ≤ α then
                break (* β cut-off *)
        return value
    else (*minimizing player *)
        value := ∞
        for each child of node do
            value := min(value, alphabeta(child, depth -1, TRUE , α, β))
            β := min(β, value)
            if β ≤ α then
                break (* α cut-off *)
        return value
```

그림 6 Alpha-Beta Pruning 의사코드

Fig. 6 Pseudo code of Alpha-Beta Pruning

그림 6은 Alpha-Beta Pruning의 의사코드이다. 최소최대탐색과 비교해 추가된 파라미터로 α, β가 있다. 이미 탐색한 영역에서 α, β는 반환된 값 중 각각 최댓값, 최솟값을 가진다. α는 최대화 행위자일 때 최소영역의 범위를 설정해 주고, β는 최소화 행위자일 때 최대영역의 범위를 지정해 준다.

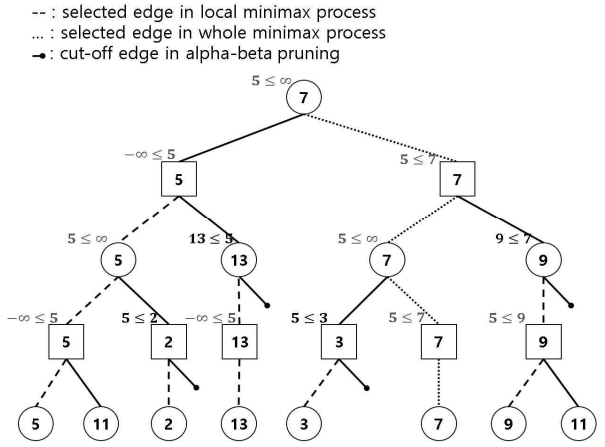

그림 7 Alpha-Beta Pruning의 선택 및 소거 과정
Fig. 7 Selection and Eliminating process of Alpha-Beta Pruning

alpha-beta pruning은 이와 같이 탐색의 최소, 최대 범위를 지정하여 탐색이 필요 없는 경우의 탐색을 제거함으로써 최소최대탐색과 동일한 결과를 얻으면서도 평균적인 연산 시간을 줄이는 장점을 갖는다.

3.3 게임 플레이어의 의사결정 방법

본 논문에서는 게임의 진행 상태 s에서 플레이어의 의사결정을 위하여 최소최대탐색의 말단 노드(leaf node) s_l에 대해 학습된 인공신경망의 출력인 $v(s_l)$을 적용하여 탐색 결정을 통해 의사결정을 수행한다. 본 논문에서 최소최대탐색과 근사화한 가치평가 네트워크를 통한 의사결정과정 $\pi_\theta(s,a)$는 식 (6)과 같이 정의한다.

$$\pi_\theta(s,a) \doteq minimax(s,a,\theta) \tag{6}$$

식 (6)에서 s는 플레이어가 의사를 결정할 현재의 게임 상태를 의미하고 a는 의사결정과정에 의해 결정된 행동을 의미하며 θ는 인공신경망의 학습 파라미터를 나타낸다.

그림 8은 본 논문에서 사용된 오델로 인공지능 플레이어의 의사결정 과정을 도식화한 것이다. 오델로 시뮬레이터에서 진행상황 $k(k=1,2,3,\cdots,60)$에 따른 상태 s_k가 최소최대탐색에 입력되면 최소최대탐색은 탐색 트리에 따라 탐색된 말단 노드 상태인 s_l에 대한 형세의 판단을 위해 학습된 인공신경망으로부터 가치평가 함수값 $v_\theta(s_l)$을 제공 받는다. 말단 노드의 각 상태에서 평가된 형세의 가치함수를 근거로 하여 최소최대탐색은 현재 상태 s에 대한 최선의 선택을 하며 선택 결과인 a_k는 현재 상태에서 취할 행동으로 결정된다. 이와 같이 본 논문에서 제안하는 게임 플레이어의 의사결정과정 π_θ는 최소최대탐색과 value network의 상호작용이라고 할 수 있으며 이러한 일련의 과정을 통해 의사결정을 수행한다.

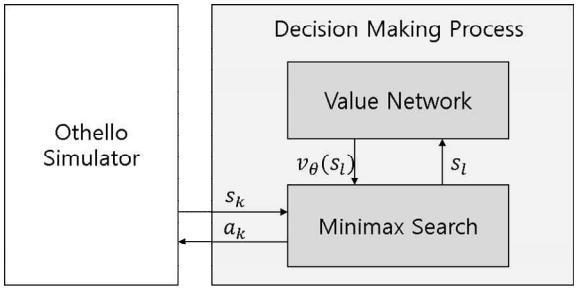

그림 8 제안하는 오델로 게임 플레이어의 의사결정 구조
Fig. 8 Block Diagram of the Proposed Decision-Making Scheme of the Othello Game Player

그림 9는 본 논문에서 사용한 의사결정 과정의 한 예를 도시한 것이다. 본 논문에서 최소최대탐색의 깊이(depth)는 5로 설정하였다. 이는 현재의 상태 s(root node)에서 5수 이후까지의 진행 상황을 학습된 인공신경망을 통해 예측하고 예측된 결과들에 대해 최선의 선택을 하도록 하였음을 의미한다.

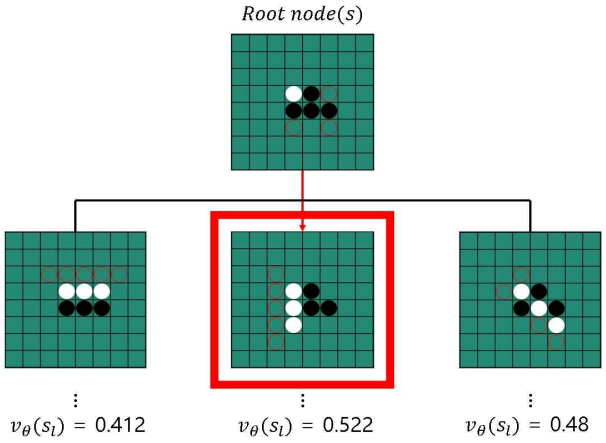

그림 9 의사결정의 예
Fig. 9 Example of Decision-Making

4. 실험 및 결과

4.1 실험 환경

본 논문의 실험은 CPU: Intel i7-7700k 4.2GHz, GPU: NVIDIA GeForce GTX 1080 Ti 사양의 하드웨어 및 Window OS 환경에서 Python으로 구현하여 진행되었다. 오델로 시뮬레이

터 구축에는 pygame, numpy 등을 사용하였고 인공신경망 생성에는 파이썬 기반의 머신러닝 오픈소스 라이브러리인 tensorflow와 keras를 사용하였으며 학습데이터 생성 및 분류를 위해 scikit-learn를 사용하였다.

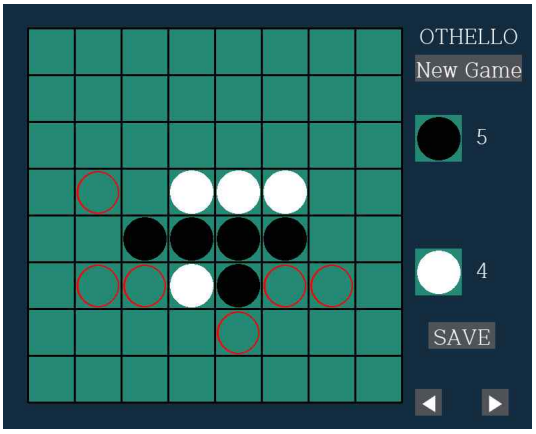

그림 10 오델로 시뮬레이터
Fig. 10 Othello Simulator

그림 10은 본 논문에서 구현한 오델로 게임 플레이어의 사용자 인터페이스 화면 구성을 나타낸다. 기본적으로 오델로 게임의 규칙에 기반하여 진행하도록 설계되었고 착수 가능한 위치를 표시하였으며 진행된 대국 결과를 기보화하는 기능 등 학습과정에 필요한 기능들을 부가적으로 포함하고 있다.

4.2 기보학습 인공신경망의 구조 및 파라미터

인공신경망의 기보 학습 및 평가는 생성된 학습데이터 7,314,328개를 8:2의 비율로 training data 5,851,462개와 test data 1,462,866개로 나누어 진행하였다.

표 3 제안된 인공신경망의 구조와 파라미터
Table 3 Structure and Parameter of the proposed ANN

Layer	Node #	Activation Function	Parameter #
1	128	ReLU	8,320
2	256	ReLU	33,024
3	64	ReLU	16,448
4	1	Sigmoid	65

표 3은 가치평가 함수를 학습하기 위한 인공신경망의 구조와 파라미터를 나타내는 것으로 3개의 은닉층과 하나의 출력층으로 구성되어 있으며 총 57,857개의 파라미터를 포함하고 있다.

표 4는 기보 데이터의 학습을 위해 사용한 인공신경망의 파라미터를 나타낸다. 본 학습에서 학습률은 0.001로 설정하였고 과적합을 방지하기 위해 Adam-Optimizer와 정규화를 적용한 early-stopping을 사용하였다. 표 3과 표 4에서 제시한 구조와

표 4 기보 학습에 사용된 Hyper Parameter
Table 4 Hyper Parameters used for Records Learning

Hyper Parameter	Value
loss function	mean squared error
optimizer	Adam
learning rate	0.001
Adam_beta1	0.9
Adam_beta2	0.99
epoch	300
early stopping	True
patience	10
batch size	1024

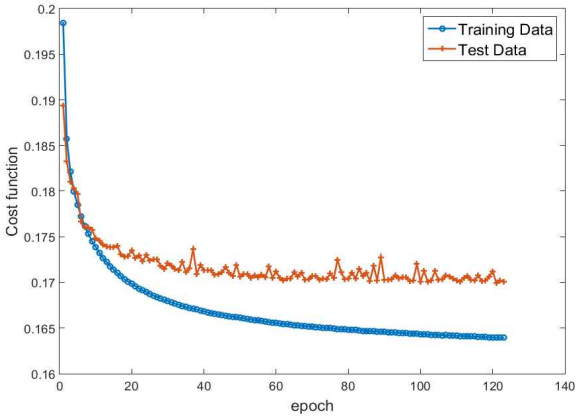

그림 11 학습 진행에 대한 비용함수 그래프
Fig. 11 Cost function graphs in terms of learning stage

파라메터를 적용하여 학습한 인공신경망의 학습 진행 상황에 따른 비용함수 변화를 그림 11에 제시하였다. 본 논문에서 제안한 인공신경망은 학습의 진행에 따라 비용함수의 값이 수렴하고 있어 적절한 학습이 이루어진 것을 확인할 수 있다.

4.3 오델로 인공지능 플레이어의 성능 평가

본 논문에서는 제안한 의사결정 방법을 사용하여 오델로 게임의 인공지능 플레이어를 구현하였으며 구현된 플레이어의 성능 평가를 위해 기존의 서로 다른 3가지 알고리즘의 지능 플레이어 ULTRA, JONHAFISH, LITE와 실제 대국을 수행하여 결과를 제시하였다. ULTRA는 최소최대탐색을 기반으로 하고 있고 JONHAFISH는 패턴으로 분류된 상황에 따라 전략을 달리하는 알고리즘이며 평가 대상으로 사용된 알고리즘 중에서는 JONHAFISH의 성능이 가장 우수하고 LITE는 상대적으로 취약한 것으로 알려져 있다.

본 실험에서는 제안 인공지능 플레이어에서 사용하는 최소최대탐색의 깊이(depth)를 50번째 수를 기준으로 탐색 경우의 수에 따라 5와 7의 가변깊이를 적용하였다. 각기 다른 알고리즘에 대해 흑과 백을 기준으로 각각 50회의 대전을 시행하였으며 구체적인 결과는 표 5와 표 6에 제시하였다.

표 5 다른 AI와의 대국 결과(흑)
Table 5 Match Results with other A.I.(black)

vs. Algorithm	대전 횟수	승리
ULTRA	50	42
JOHNAFISH	50	46
LITE	50	50
summary	150	138

표 6 다른 AI와의 대국 결과(백)
Table 6 Match Results with other A.I.(white)

vs. Algorithm	대전 횟수	승리
ULTRA	50	41
JOHNAFISH	50	47
LITE	50	50
summary	150	138

실험결과 표 5와 표 6에 제시한 바와 같이 제안한 플레이어가 흑인 경우 세 알고리즘에 대해 각 승률은 84%, 92%, 100%를 나타냈으며 백인 경우는 각각 82%, 94%, 100%의 승률을 기록하여 종합적으로 약 92%의 승률을 나타내 기존의 비교 알고리즘에 비해 우수한 결과를 나타냄을 확인할 수 있었다.

5. 결 론

본 논문에서는 오델로 게임의 인공지능 플레이어 구현을 위해 인공신경망의 학습 방법을 제안하고 이를 최대최소 탐색에 적용하여 게임의 각 진행 상태에서 최선의 수를 선택하기 위한 의사결정 방법으로 사용하였다. 이를 위해 본 논문에서는 프로기사들의 실제 기보를 게임의 진행 단계에 따라 상태별로 분해하고 승패에 대한 가치를 부여하여 전체 기보 데이터에 대해 이를 누적함으로써 학습 데이터를 생성하였고 인공신경망을 통해 학습함으로써 모든 가능한 경우의 상태에 대해 가치평가함수를 구할 수 있는 방법을 제안하여 적용하였다.

본 논문에서 제안한 오델로 게임 플레이어는 기존의 서로 다른 세 가지 알고리즘의 플레이어와 실제 대국을 통해 성능을 평가하였다. 실험결과 제안한 플레이어는 흑, 백으로 둔 경우 모두 세 알고리즘에 대해 종합적으로 92%의 승률을 나타내 기존의 비교 알고리즘에 비해 우수한 결과를 나타냄을 확인할 수 있었다.

실험 결과 본 논문에서 제안한 인공지능 플레이어는 게임의 초·중반에 강한 특성을 나타내었는데 이는 기보에서 등장하는 상태의 상대적 빈도를 고려한 비용함수를 적용함으로써 인공신경망이 게임의 초·중반에 실제 자주 등장하는 패턴의 형세를 적절하게 학습한 결과인 것으로 분석된다. 또한 실험 결과 기보 상에서 상대적으로 출현빈도가 낮은 실제 게임의 후반부 상태에서도 플레이어가 게임의 주요전략들에 부합하는 수준 높은 수들을 선택함을 확인할 수 있었는데 이는 제안 인공신경망이 많은 수의 데이터 학습을 통해 전략적 측면의 패턴에 대해서도 잘 학습된 결과를 나타내는 것으로 볼 수 있다.

본 논문에서는 기보의 상태를 학습하기 위해 단순 구조의 인공신경망을 사용하고 최선의 수 선택을 위해 최대최소 탐색을 적용하였으나 최근 영상인식 분야에 폭넓게 사용되고 있는 CNN 등 보다 복잡한 구조의 신경망을 사용하거나 강화학습을 적용하여 학습된 신경망을 자가 학습으로 보완하면 보다 개선된 결과를 얻을 수 있을 것으로 기대된다.

감사의 글

본 연구는 2018학년도 서경대학교 교내연구비 지원에 의하여 이루어졌음.

References

[1] Silver, D. et al., "Mastering the game of Go with deep neural networks and tree search", *Nature 529*, 484-489, 2016.

[2] Silver, D. et al., "Mastering the game of Go without human knowledge", *Nature 550*, 354-359, 2017.

[3] M. Campbell, A.Joseph Hoane, Feng-hsiung Hsu, "Deep Blue", *Artificial Intelligence*, Volume 134, Issues 1-2, 57-83, 2002.

[4] M. Buro, "LOGISTELLO-A Strong Learning Othello Program", *NEC Research Institute, Princeton*, NJ, 1997.

[5] P. S. Rosenbloom, "A World-Championship-Level Othello Program", *Artificial Intelligence 19*, 279-320, 1982.

[6] K.-F. Lee, S. Mahajan, "The Development of a World Class Othello Program", *Artificial Intelligence 43*, 21-36, 1990.

[7] J. Schaeffer, H.J. van den Herik, "Chips Challenging Champions: Games, Computers and Artificial Intelligence", 135, 2002.

[8] Gunawan, Hendrawan Armanto, Joan Santoso, Daniel Giovanni, Faris Kurniawan, Ricky Yudianto, Steven, "Evolutionary Neural Network for Othello Game", *Procedia -Social and Behavioral Sciences*, Volume 57, Pages 419-425, 2012.

[9] P. Liskowski, W. M. Jaskowski and K. Krawiec, "Learning to Play Othello with Deep Neural Networks", *in IEEE Transactions on Games*, 2018

[10] R. S. Sutton, A. G. Barto, "Reinforcement Learning : An Introduction", *MIT Press, Cambridge, MA*, 1998.

[11] N.J. van Eck and M. van Wezel, "Reinforcement learning and its application to othello", *Technical Report EI 2005-47, Econometric Institute Report*, 2005.

[12] M. van der Ree and M. Wiering, "Reinforcement

learning in the game of Othello: Learning against a fixed opponent and learning from self-play", *2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL), Singapore*, pp. 108-115, 2013.

[13] J. P. Fishburn, "Appendix A: Some Optimizations of α-β Search", *Analysis of Speedup in Distributed Algorithms, UMI Research Press*, pp. 107-111, 1984.

[14] S. H. Lee, D. Y. Kim, and Y. W. Cho, "Image Recognition based on Deep Learning Using Multi- Dimensional Neural Network and Decision Making Neural Network", *Journal of Korean Institute of Intelligent Systems*, vol. 28, no. 1, pp. 34-40, 2018.

저 자 소 개

전 영 진 (Youngjin Jeon)

2017년: 서경대학교 컴퓨터공학과 공학사
2017년~현재: 서경대학교 전자컴퓨터공학과 석사과정 재학 중
관심분야: 딥러닝 및 강화학습, 인공지능

조 영 완 (Youngwan Cho)

1991년: 연세대학교 전자공학과 공학사
1993년: 연세대학교 전자공학과 공학석사
1999년: 연세대학교 전자공학과 공학박사
2000년~2003년: 삼성전자 책임연구원
2003년~현재: 서경대학교 컴퓨터공학과 교수
관심분야 : 지능제어시스템, 무인이동체제어, 딥러닝 및 강화학습