Journal of Korea Game Society 2018 Aug; 18(4): 99-106
http://dx.doi.org/10.7583/JKGS.2018.18.4.99

게임 인공지능

미니맥스 알고리즘을 이용한 학습속도 개선을 위한 Q러닝

신용우
동아방송예술대학교 창의융합교양학부
ywshin@dima.ac.kr

Q-learning to improve learning speed using Minimax algorithm

YongWoo Shin
Division of Creative Convergence Education, Dong-Ah Institute of Media and Arts

요 약

보드게임에서는 많은 경우의 수의 말들과 많은 상태공간들을 가지고 있다. 그러므로 게임은 학습을 오래 하여야 한다. 본 논문에서는 Q러닝 알고리즘을 이용했다. 그러나 강화학습은 학습초기에 학습속도가 느려지는 단점이 있다. 그러므로 학습을 하는 동안에 같은 최선의 값이 있을 때, 게임트리를 고려한 문제영역의 지식을 활용한 휴리스틱을 사용하여 학습의 속도향상을 시도하였다. 기존 구현된 말과 개선하여 구현된 말을 비교하기 위하여 보드게임을 제작했다. 그래서 일방적으로 공격하는 말과 승부를 겨루게 하였다. 개선된 말은 게임트리를 고려하여 상대방 말을 공격하였다. 실험결과 개선하여 구현된 말이 학습속도적인 면에서 향상됨 것을 알 수 있었다.

ABSTRACT

Board games have many game characters and many state spaces. Therefore, games must be long learning. This paper used reinforcement learning algorithm. But, there is weakness with reinforcement learning. At the beginning of learning, reinforcement learning has the drawback of slow learning speed. Therefore, we tried to improve the learning speed by using the heuristic using the knowledge of the problem domain considering the game tree when there is the same best value during learning. In order to compare the existing character the improved one. I produced a board game. So I compete with one-sided attacking character. Improved character attacked the opponent's one considering the game tree. As a result of experiment, improved character's capability was improved on learning speed.

Keywords : Gonu game(고누게임), Minimax algorithm(미니맥스 알고리즘), Q Learning (Q러닝), Learning speed(학습속도)

Received: Jul. 18. 2018 Revised: Aug. 12. 2018
Accepted: Aug. 20. 2018
Corresponding Author: YongWoo Shin(Dong-Ah Institute of Media and Arts)
E-mail: ywshin@dima.ac.kr

ISSN: 1598-4540 / eISSN: 2287-8211

1. 서 론

2016년 4분기 게임 산업의 매출액은 3조 4,855억 원이며 이는 전년 동기 대비 21.7% 증가한 수준이며, 전체 콘텐츠산업 매출액에서 차지하는 비중은 11.5% 이다[1].

게임 산업이 나날이 발전하고 대작들이 많이 제작되므로, 인공지능도 많은 발전을 이루어 부가가치를 만들어야 한다. 게임시나리오와 기획이 게임을 재미있게 한다. 그러나 새로운 게임 인공지능 기법이 도입된다면 다양하고 재미있는 게임의 구현이 가능할 것이다.

인공지능 기법 중에서는, 학습 알고리즘을 적용하여야 한다. 강화학습은 온라인으로 학습하여 캐릭터를 발전시킨다. 특정목적을 달성하였을 때마다 보상을 받는 작업을 반복되다보면 캐릭터가 학습하게 된다[2].

강화학습으로 연구된 분야에는 상태공간의 문제, 캐릭터의 지능화의 문제가 있었는데, 상태공간의 효율적인 사용의 문제는 여러 논문에서 다루었다. 그러므로 본 논문에서는 캐릭터의 지능화를 다루고자 한다. 기존의 캐릭터의 지능화를 다룬 논문들은 서양 보드게임(Board game)인 오델로, 틱택토 등을 다루었으나[3,4] 본 논문에서는 줄고누게임을 다루고자 한다.

논문[5] 에서는 강화학습을 이용하여 줄고누 게임을 구현하였다. 오델로 게임의 경우에는 말의 위치가 정해지면 움직이지 않지만, 줄고누게임의 경우는 말의 위치가 계속 바뀐다는 점에서 구현하기가 상대적으로 까다롭다. 그러나 강화학습을 위해 Q러닝(Q-learning) 알고리즘을 사용한 기존 논문[5]에서는, 학습과정에서 학습이 많지 않은 초반부의 학습 자료의 부족으로 학습속도가 느려지는 단점이 있었다. 본 논문의 선행 연구를 다룬 영향력 분포도 논문[6]에서는 이러한 단점을 없애기 위하여, 학습의 초반부에서 학습속도가 느려질 때, 영향력분포도를 이용하여 학습속도를 단축시켰다.

본 논문에서 제안하는 방법은 학습과정에서 최선의 값을 산출하는 부분에서 동일한 값이 나왔을 때, 미니맥스 알고리즘(Minimax algorithm)을 이용하여 유리한 값을 선택하도록 하였다. 실험 결과 논문[6] 에서의 게임보다 미니맥스를 적용한 말이 우수한 게임을 하는 것을 알 수 있었다.

본 논문의 구성으로는, 1장 서론에 대해 논하며 2장에서는 관련연구에 대해 알아보고 3장에서는 지능형 보드게임의 구현을 논하며 4장에서는 실험 및 결과에 대해 논한다. 마지막으로 5장에서는 최종적인 결론을 맺는다.

2. 관련연구

2.1 Q러닝

환경에는 목적달성에 필요하거나, 필요치 않은 많은 상태들이 존재한다. 에이전트는 상태를 경험하여 목적을 달성하는 경우, 환경으로부터 보상을 받는다. 목적을 달성할 수 없는 상태인 경우에는 보상을 받을 수 없다. 많은 시행착오를 겪을 수 있고, 모든 상태를 경험하여야 한다.

Q러닝은 미리 모델을 설정하거나, 학습할 필요가 없고, 상태공간을 충분히 경험하기만 하면 된다.

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t)+\alpha(r_{t+1}+\gamma \max_{a} Q(s_{t+1},a')-Q(s_t,a_t))$$

[Fig. 1] Q Learning Algorithm

상태 s 의 행위 a 의 보상 r 을 얻을 수 있도록 식이 있다[2]. t 는 이전시간, t+1 은 이후시간을 말한다. max 란 최대값이고, α 는 학습속도 매개변수로서 0 에서 1 사이이며 γ 는 할인계수로서 0 에서 1 사이를 말한다.

모든 가능한 상태공간의 특정 상태 s 의 행위 중 가장 좋은 보상값 r 을 저장하고 산출한다. 처음에는 모든 상태 하나하나에 대하여 보상을 얻기 위해 마음대로 행위를 하여 경험을 쌓게 된다. 그러나 경험한 상태가 많을수록 보상을 토대로 불필

요한 경험을 하지 않는다. 모든 상태를 경험하여 학습이 완료되어 지능적인 동작이 가능하게 된다.

2.2 고누게임(Gonu game)

장기와 바둑의 원초적 형태를 띄고 있어서 고대 중국의 초나라와 한나라 때 생긴 장기놀이가 우리나라에 들어와 재창작된 것으로 짐작 된다[7].

고누의 종류는 여러 가지 특징이 있으나 우물고누, 줄고누, 밭고누, 곤질고누, 자동차고누, 참고누, 호박고누, 패랭이고누, 장수고누, 팔팔고누, 포위고누, 왕고누로 구분해 볼 수 있다[8].

고누의 종류는 대부분 말판의 모양에 따라서 명칭이 붙여진 것이다.

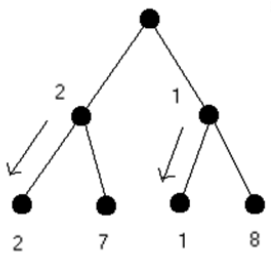
줄고누게임의 규칙은 다음과 같다. 상대방 말을 포획하려면, 상대방 말 옆에 우리 말 둘이 일직선으로 있으면 된다. 또한 우리 말을 포획하려면, 상대방 말의 일직선상에 들어가면 포획 당하게 된다. 게임을 진행하는 방법은 한 번에 한 칸씩만 직선으로 이동한다.

2.3 미니맥스 알고리즘

여기서는 비공식적이지만 한 경기자의 한 행동을 그의 수(Move) 라 하자. 관습적으로 양의 숫자는 한 경기자에겐 이득으로, 음의 숫자는 상대방에서 이득이 있는 것으로 가정하자[9].

[Fig. 2]에서 아래쪽 노드들과 루트 노드(Root node) 가 최대화 레벨(Maximum level) 이고, 가운데 노드들은 최소화 레벨(Minimum level) 이다. 아래 그림과 같은 상태에서 단말 노드들의 값 중 최소값은 최소화 레벨에서 택하고 루트노드에서는 그 중 최대값을 택하게 된다.

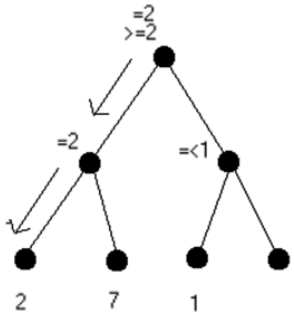
이와 같이 두 수를 내다봄으로 경기자는 해당되는 값을 가지는 놀이상황을 선택하게 된다.



[Fig. 2] Minimax algorithm

2.4 알파베타 가지치기

미니맥스 알고리즘으로 만들어지는 모든 트리 중 최대값 또는 최소값을 만드는데 불필요한 트리(Tree) 는 사전에 가지치기함으로서 알고리즘의 전체속도와 효율성을 높여주는 프로시져(Procedure) 가 알파베타 가지치기이다[9].



[Fig. 3] Alpha beta Pruning

[Fig. 3] 의 최소화레벨에서 최소값은 2 로 결정되고 알파 값 또한 2 로 결정 되었다. 1의 값을 가지는 우측 단말노드에 의해 최소화레벨이 1로 결정되었을 때 우측 마지막 단말노드 까지 가보지 않아도 루트노드는 2 를 선택할 것이다. 최소화레벨이 1 보다 작거나 같기 때문이다. 이와 같은 것이 알파베타 가지치기이다. 1의 값을 가지는 노드의 우측 노드를 0 과 3 으로 각각 하나씩 대입해 보면 확실히 원리를 알 수 있을 것이다.

3. 지능형 보드게임의 구현

본 논문에서는 상대방 말과의 대국을 학습하는 인공지능을 구현한다.

3.1 제안하는 알고리즘

강화학습에서의 상태는 계속되는 게임의 결과에 따라, 보상값들이 상태에 누적되게 된다. 게임이 일정기간 진행되면 누적된 값으로 인해 우리 말에게 가장 좋은 값을 선택할 수 있다.

게임의 초기에는 0으로 초기화되어 모두 같은 값들을 갖고 있다. 게임이 조금 진행된 초기에는 보상값이 누적되지 않아서 동일한 값들을 많이 가지고 있다. 이 때 무작위로 그중 하나의 값을 선택하게 된다. 그래서 적절한 위치로 이동하지 못한다. 상대방 말에게 포획되거나 잘못된 방향으로 움직이게 된다.

본 논문에서는 이러한 단점을 없애기 위해, [Fig. 4] 와 같이 동일한 값들이 추출될 때, 미니맥스를 고려하여 유리한 방향으로 움직이도록 Q러닝 알고리즘을 개선하였다.

```
procedure Q_learning
{
    Find Max from QTable
    If all Q values are equal,
            Apply minimax algorithm
    Select player character's action from QTable

    Go player character

    Go enemy character from e_action()
    If (catch)
        Create plus reward
        If (be captured)
                Create minus reward
    Update QTable
}
```

[Fig. 4] QMM_learning Algorithm

현재 학습의 초창기라면 보상 값이 많이 누적되지 않아 동일한 0의 값들을 갖고 있다. 그러므로 우리 말이 진행할 방향을 미니맥스 알고리즘에서 결정한 곳으로 전달받아 상대방 말을 공격한다.

본게임은 우리 말 둘이 모여서 상대방 말을 공격하게 되어있다.

3.2 이동생성

본 고누게임은 게임트리(Game tree)를 이용하였다. 게임트리는 각 노드마다 무수히 많은 해결책(경우의 수) 과 함께 두 경기자의 모든 상황을 표현해 주는 수학적 도구이다. 그리고 게임트리에는 이동생성기, 노드평가함수, 그리고 탐색이 있다. [Fig. 2] 는 두 수 앞을 내다보는 게임트리이다.

이동생성 과 탐색에 적용할 수 있는 기본적인 알고리즘으로 넓이우선(Breadth-first) 탐색과 깊이우선(Depth-first) 탐색이 있다. 넓이우선 탐색은 [Fig. 2] 와 같이 하위 레벨로 가기 전에 현재 레벨에 있는 모든 경우의 수를 탐색 하는 것을 말하며, 깊이우선 탐색은 그래프의 왼쪽면만 트리의 끝까지 또는 미리 정한 레벨까지 수를 탐색하고 그 다음 왼쪽 면으로 이동하여 경우의 수를 탐색하는 것을 말한다.

본 고누 게임에서는 넓이우선 방법을 기준으로 사용한다. 이동생성은 넓이우선 탐색과 같은 방법으로 위에서 아래로 내려오며 노드들을 생성한다. 탐색은 넓이우선 탐색을 변형한 형태로 모든 경우의 수가 나타난 게임트리가 완성되었을 때 현재의 노드에서 가장 적합한 수로 이동하기 위하여 게임트리의 맨 끝부분, 즉 고누 게임의 승부가 결정되는 노드로부터 평가함수를 적용하며 현 위치로 거슬러 올라간다.

3.3 평가함수

고누게임을 컴퓨터로 구현하기 위해서는 현재 게임의 상황에서 몇 수 앞을 내다보고, 여러 가지 둘 수 있는 수 중 유리한 수들을 평가해 본다. 평

가된 값들 중 현재 나에게 이로운 값들을 판단하기 위해서는 평가함수(Evaluation function) 가 있어야 한다.

이 가치평가는 게임에 영향을 미칠 수 있는 다양한 특성에 근거를 둔다[10].

삼목놀이의 간단한 예에 미니맥스를 적용 해보자. 삼목놀이란, 3개의 말을 가지고 서로 한수씩 번갈아 대국한다. 그래서 먼저 대각선이나 가로로 3개의 말을 이으면 승리하는 게임이다.

맥스는 가위표(X), 미니는 동그라미(O) 로 표시하고 제일 먼저 맥스부터 시작한다 하자. 2 단계까지의 모든 노드를 발생시키는 넓이우선 탐색 을 행한 후 정적 평가함수를 끝단 노드들에 적용한다. 노드의 위치 p 에 대한 평가함수 e(p) 는 다음과 같이 주어진다[10].

e(p) = (MAX 가 표시하여 이길 가능성을 가지는 행이나 열 혹은 대각 의 수) - (MIN 이 표시하여 이길 가능성을 가지는 행이나 열 혹은 대각의 수)

본 논문에서의 고누게임에서도 역시 이길 가능성을 가지는 방법으로 평가함수를 제안하였다.

즉, 현재 이동할 수 있는 모든 말 각각에서 상대방의 말을 제거할 수 있는 경우와 다음 수에서 제거할 수 있는 경우를 계산하는 등의 경우를 말한다. 고누게임에서의 평가함수의 공식은 [Fig. 5] 와 같다.

$$Ef = \sum_{i=1}^{N} (e1*w1 + t1) - \sum_{i=1}^{N} (e2*w2 + t2)$$

[Fig. 5] Evaluation function

전체 공간에서의 상대방의 말을 제거할 수 있는 경우에 가중치를 곱한 값과, 제거할 가능성이 있는 경우를 더한 값을 우리 말에서는 양의 값으로, 상대방 말에서는 음의 값으로 계산하여 나온 값을 평가 값으로 한다.

Ef 는 평가함수를 의미한다. N 은 전체공간에서 둘 수 있는 모든 경우의 수를 의미한다. e1 과 e2 는 각각 서로 다른 상대방 말을 의미한다. e 는 상대방 말을 포획하는 경우이다. w 는 가중치이다. 가중치는 여러 가지 값을 가질 수 있으며, e 를 w 에 곱했으므로 e 와 t 의 상대적 중요성에 따라 결정된다.

4. 실험 및 결과

4.1 게임소개 및 실험환경

캐릭터들은 우리 말과 상대방 말로 각각 둘이 존재한다.

[Fig. 6] 과 같이, 우리 말의 공격은 둘이 일직선으로 인접해있을 때, 그 옆에 상대방 말이 있으면 상대방 말을 포획하게 된다. 일직선상에 놓이지 않으면 방어가 된다.

상대방 말은 우리 말 쪽으로 움직여서 공격하게 되며, 한 수를 두어서 우리 말을 포획할 수 있을 때는 정확히 우리 말을 포획하게 되어있다. 즉, 우리 말 하나가 상대방 말에 인접되었다면, 상대방 말은 우리 말을 포획할 수 있도록, 일직선이 되기 위한 지점으로 정확히 이동하도록 되어있다.

또한, 실수로 상대방 말의 일직선상에 들어갔다면 상대방 말에게 포획 당하게 된다. 상대방 말의 경우에도, 우리 말의 일직선상에 들어오면 포획 당하게 된다.

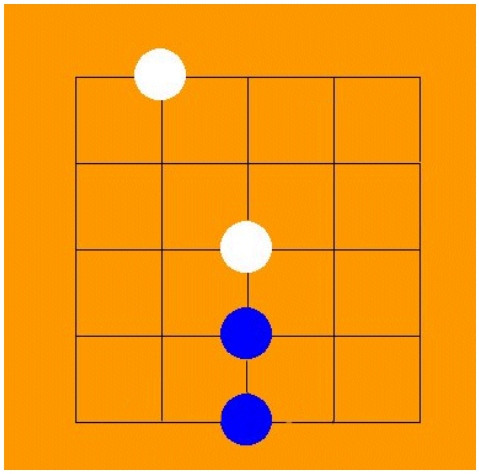
우리 말과 상대방 말의 각각의 위치는 메모리에 각각의 상황에서의 학습점수가 저장된다. 우리 말이 움직일 때에는 선택할 수 있는 점수 중에서 유리한 것을 선택하게 된다. 그러나 학습의 초기에는 저장된 데이터가 많지 않아서 좋은 선택을 못하게 된다.

우리 말이나 상대방 말은 공격과 방어를 하게 되고, 누가 먼저 상대방 말을 포획하느냐가 승부의 관건이 된다. 상대방 말은 우리 말 방향으로 움직

여서 우리 말을 공략하고, 우리 말은 보상점수에 의해 가장 좋은 방향으로 이동한다.

우리 말이 상대방 말을 포획한 경우에는 양의 보상이 주어진다. 우리 말이 포획된 경우에는 음의 보상이 주어진다.

상대방 말의 경우, 초기에는 학습이 되지 않은 우리 말을 쉽게 포획하게 된다. 우리 말의 경우에는, 초기에는 학습이 되지 않았기 때문에 우리 말이 상대방 말을 포획하기는 어렵다. 그러나 학습이 된 후에는 우리 말도 상대방 말을 포획할 수 있게 되고, 최종적으로 상대방 말이 전혀 우리 말을 포획하지 못하게 된다.



[Fig. 6] State in the case of captured between opponents

4.2 실험결과

실험은 우리 말과 상대방 말이 각각 둘일 경우로 실험하였다. 우리 말은 학습 되었으므로 현재의 위치에서 최적의 지점으로 이동하도록 설계되었으며, 상대방 말의 공격 알고리즘은 다음과 같다. 우리 말이 있는 곳의 위치를 파악하여, 최단거리를 선택하여 이동한 후 공격한다.

[Table 1] Experiment results in existing reinforcement learning

Battle / Learn	Win 1	Win 2	Lose 1	Lose 2	Tie
1000	24	8	80	1	0
2000	89	27	132	1	1
3000	266	80	138	1	3
4000	462	135	138	1	6
5000	675	186	138	1	7

[Table 1] 에서의 Win1 과 Win2 는 Q러닝으로 학습된 말이 이긴 경우를 말한다. 그리고 Lose1 과 Lose2 는 공격위주의 말이 이긴 경우이다. 이긴 경우가 두 가지가 있는 것은 정상적으로 이겼을 경우와, 상대방 말이 실수로 함정에 들어와서 이긴 경우 때문이다. Win1 과 Lose1 은 정상적으로 이긴 경우이고, Win2 와 Lose2 는 실수로 이긴 경우를 말한다. 학습을 하는 말과 공격위주의 말의 실험결과는 현격한 차이를 보임을 알 수 있다.

공격위주의 말과 학습된 말과의 포획횟수를 비교해보면, 초반 2000회 까지는 공격위주의 말이 앞서지만 3000회 부터는 학습된 말의 승리횟수는 큰 격차를 보이며 늘어나는 것을 알 수 있다. 또한 학습된 말이 지는 경우는 138 번으로 멈춘 것을 볼 수 있고, 학습된 말이 실수하여 지는 것은 단 1회 뿐인 것을 알 수 있다.

[Table 2] Experiment results in improved reinforcement learning

Battle / Learn	Win 1	Win 2	Lose 1	Lose 2	Tie
1000	162	5	0	0	27
2000	338	9	0	0	50
3000	511	17	0	0	73
4000	653	20	0	0	105
5000	815	26	0	0	133

[Table 2] 에서 보면 알 수 있듯이 미니맥스를 고려하여 강화학습을 하는 말과 일방적인 공격위주

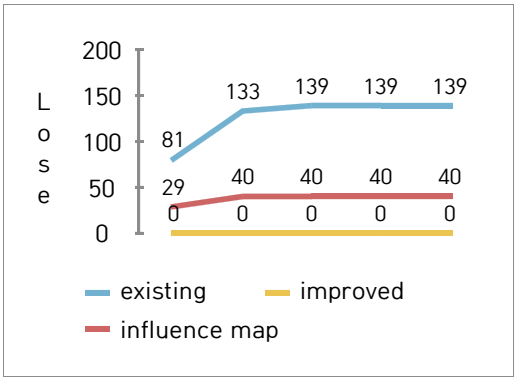
로 움직이는 말은 차이가 있다는 것을 알 수 있다.

학습을 하는 경우에 학습속도적인 측면에서 본다면, 우리 말 은 이미 초반 1000회 에서부터 상대방 말을 승리횟수에서 앞서가는 것을 알 수 있다. 이후로도 승리횟수는 큰 차이를 보이는 것을 알 수 있다. 패배의 경우에는 정상적으로 지거나 실수로 진 것이 단 한 번도 없었다. 이것이 시사 하는 바는, Q러닝의 학습초기에서 보상 값이 쌓이기 전에는, 정확하게 학습할 수 없다는 단점을 완전히 보완하는 것이다.

중요한 점은, 계속 실험을 한다면, 우리 말의 패배 횟수가 0인 상태에서 우리 말의 승리 횟수가 증가될 것으로 예상된다.

[Table 3] Experiment results of existing reinforcement learning, influence map, and improved reinforcement learning

Battle / Learn	existing	influence map	improved
1000	81	29	0
2000	133	40	0
3000	139	40	0
4000	139	40	0
5000	139	40	0



[Fig. 7] Comparison between existing reinforcement learning, influence map, and improved reinforcement learning

[Table 3] 에서 기존의 강화학습, 영향력 분포도 그리고 개선된 후의 학습의 실험결과를 볼 수 있고, [Fig. 7] 에서 직접 비교해 보았다. 세 경우 모두 일방 공격형 말과 대국할 경우에 승리횟수는 꾸준히 증가하는 것을 알 수 있다.

그러나 패배횟수에서는 차이를 보이는 것을 알 수 있다. 가장 많은 패배횟수를 기록하는 것은 139번까지 기록하는 기존 강화학습이다. 영향력 분포도의 경우에는 최대 40번까지 지게 된다[6]. 그러나 개선된 학습에서는, 단 한 번도 지지 않게 된다.

5. 결 론

시나리오가 게임의 흥미를 만들어내지만, 캐릭터의 지능화는 여전히 중요한 문제라고 할 수 있다. 캐릭터의 지능화로 인해 게임의 재미가 더해질 수 있고, 그것은 강화학습을 포함한 여러 가지 학습알고리즘들을 효과적으로 이용할 때, 가치가 더해진다.

일반적인 강화학습에서는 일정기간동안 보상 값이 쌓이기 전에는 정확한 값을 산출할 수가 없다. 그러므로 학습초기에는 동일한 값이 산출되었을 때 무작위로 난수 값을 발생시켜서 순서를 부여한다. 이러한 단점을 개선하기 위하여 영향력 분포도를 이용한 학습에서 단점을 개선하였으나 완전히 개선하기는 어려웠다. 그러나 본 논문은 이러한 단점을 개선하기 위하여 동일한 값이 산출될 때, 게임트리를 활용한 미니맥스 알고리즘을 활용하였다. 그 결과 학습속도가 향상되었다. 학습속도가 향상됨으로 우리 말이 전략적으로 움직임으로 패배횟수가 없다.

구현된 우리 말이 잘 움직이는지 확인하기 위하여, 보드게임을 제작하였다. 기존의 강화학습으로 움직이는 방법, 영향력 분포도로 움직이는 방법 그리고 개선된 강화학습을 적용한 방법의 실험결과를 직접 비교 하였다. 실험결과, 기존방법 보다는 영향력 분포도를 활용한 방법이 패배횟수의 감소가 있었다. 그러나 미니맥스를 활용한 방법은 패배횟

수가 0에 불과했다.

본 논문은 고누게임을 구현했다는 점에서 큰 의미를 부여할 수 있지만, 실제 게임에서도 응용할 수 있다. 많은 수의 지능적인 캐릭터들이 등장하는 경우에, 학습을 시키기 위해서는 시간적으로 부족할 수 있다. 그러나 본 논문의 알고리즘으로는 상대적으로 빠르게 학습이 가능하다.

향후 논문으로는, 개선된 방법의 실험에서는 패배횟수는 없지만 무승부의 횟수가 증가하는 것을 알 수 있었다. 무승부의 횟수를 감소시키는 방법을 연구하여야 하겠다.

REFERENCES

[1] Korea Creative Content Agency, “Content Industry Trend Analysis Report for 4Q 2016 (Game Industry)”, 2017

[2] Richard Sutton, Andrew G. Barto, "Reinforcement Learning :An Introduction", MIT Press, Cambridge, MA, 1998.

[3] Imran Ghory, "Reinforcement learning in board games.", available at http://www.cs.bris.ac.uk/Publications/Papers/2000100.pdf, 2004.

[4] Nee Jan van Eck, Michiel van Wezel., "Reinforcement Learning and its Application to Othello", available at http://www.few.eur.nl/few/people/mvanwezel/rl.othello.ejor.pdf, 2004

[5] Yongwoo Shin, “Artificial Engine Development through Reinforcement Learning on Jul-Gonu Game ”, Journal of Internet Computing and Services, Vol 10, No 1, pp93-99, 2009

[6] Yongwoo Shin, “An improvement of the learning speed through Influence Map on Reinforcement Learning”, Journal of Korea Game Society, Vol 17, No 4, pp109-116, 2017

[7] Woosung Sim, “50 traditional games Korean folk play”, Nonghyup, 1996

[8] Woosung Sim, “Korean folk play”, Dongmoonsun, 1996

[9] Patrick Henry Winston, "Artificial Intelligence", Addison Wesley, 1993

[10] Sukin You, “Artificial Intelligence Fundamentals”, Kyohaksa, 1988

[11] Tozour, Paul, "Influence Mapping", Game Programming Gems 2, Charles River, 2001

[12] Laramee, Francois Dominic, "A Rule-based Architecture Using Dempster-Shafer Theory", AI Game Programming Wisdom, Charles River Media, 2002.

[13] Mommersteeg, Fri, "Pattern Recognition with Sequential Prediction", AI Game Programming Wisdom, Charles River Media, 2002.

[14] Steve Woodcock, "Game AI : The State of the Industry", Game Developer Magazine, 2000.

[15] Steve Rabin, AI Game Programming Wisdom 2, Charles River Media, 2003

[16] Steve Rabin, AI Game Programming Wisdom, Charles River Media, 2002

[17] Laramee, Francois Dominic, "Using N-Gram Statistical Models to Predict Player Behavior", AI Game Programming Wisdom, Charles River Media, 2002.

[18] Andrew Kirmse, Game Programming Gems 4, Delmar Thomson Learning, 2004.

[19] Mark Deloura, Game Programming Gems 3, Charles River Media, 2002.

[20] Mark Deloura, Game Programming Gems 2, Charles River Media, 2001.

신 용 우 (YongWoo Shin)

약 력 : 2014 경희대학교 컴퓨터공학과 박사
1990-2000 프리랜서 게임프로그래머, LG U+
2000-현재 동아방송예술대학교 교수

관심분야 : 게임인공지능, VR/AR/MR게임