강화학습에 기반한 오델로 게임의 설계 및 구현

이동훈° 우종우
국민대학교 컴퓨터 학부
{natak°, cwwoo}@cs.kookmin.ac.kr

Design and Implementation of Othello game Based on Reinforcement Learning

Donghun Lee° Chongwoo Woo
°School of Computer Science, Kookmin University

요 약

최근 인공지능의 기법을 도입한 게임에 관한 연구가 활발히 진행되고 있다. 특히 신경망의 역 전파 알고리즘을 적용한 게임은 구현이 용이하고 학습이 완료되면 비교적 실행이 빨라서 많은 연구가 진행되고 있지만, 기본적인 학습시간이 길고 최적화에 관한 문제점이 존재하고 있다. 이러한 문제점을 개선하고자 본 논문에서는 기존의 역 전파 알고리즘과 강화학습의 Q-learning알고리즘을 오델로 게임에 적용하여 비교 분석 하였다. 실험은 단순한 min-max 알고리즘과 각각 대결하여 승수 와 승율을 중심으로 비교하였고 실험의 결과는 강화학습의 알고리즘이 역 전파 알고리즘에 비하여 비교적 우수한 결과를 제시하였다.

1. 서 론

최근 컴퓨터 게임은 그래픽과 사운드를 중시하는 관점에서 벗어나, 보다 자연스럽고 재미있는 기능에 대한 관심이 고조 되고 있으며, 이러한 사용자들의 요구를 인공지능의 여러 기법들을 적용 함으로서 해결하려는 연구가 진행되고 있다. 전통적으로 이러한 인공지능의 기법들은 FSM (Finite State Machine)으로부터, Fuzzy State Machine, Decision Tree등을 적용하여 연구되었다 [1].

이러한 전통적인 기법들은 게임 설계자의 경험적 규칙에 주로 의존하게 되어, 체계적이지 못한 점과 방대한 데이터의 처리 등이 문제로 제기되었다. 이에 따른 해결방안으로 최근에는 기계학습의 알고리즘들을 적용하여 성능향상을 하려는 연구가 진행되고 있다 [2][3].

이러한 연구의 대표적인 예는 신경망의 역 전파 알고리즘의 적용이며, 역 전파 알고리즘을 적용하게 되면, 구현이 쉽고, 학습이 완료되면 비교적 실행속도가 빠른 반면, 초기 학습시간이 길며 최적화된 결과가 제공되지 못하는 문제점이 있다[4]. 본 논문에서는 이러한 문제점을 개선하고자 강화학습 알고리즘을 오델로 게임에 적용하여 시스템을 설계 및 구현하였다. 실험은 단순한 min-max알고리즘을 역 전파 알고리즘과 강화학습 알고리즘과 각각 대결하여 승율 및 승수를 비교하였다.

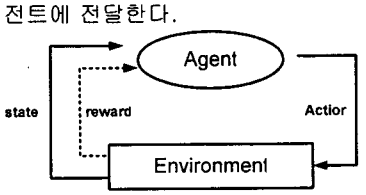
2. 관련연구

강화학습에는 다양한 접근방법이 있으며, 본 논문에서는 Q-learning알고리즘을 중심으로 기술하였다.

2.1 강화학습

강화 학습은 환경과 에이전트와의 상호정보교환을 통하여 에이전트의 행동을 개선해 나가는 학습방법이다. 강화학습에서의 환경모델은 마르코프 속성을 만족하는 MDPs Markov Decision Process)에 형식화 되어있다. 이는 시각 t+1에서의 환경반응은 오직 시각 t에서의 상태와 행동에 의존하는 속성을 말한다[5]. 일반적인 교사학습은 주어진 상황에서의 대응방법을 제시하는 반면, 강화학습은 주어진 상태에 대해 자유로운 선택을 하고 선택의 결과를 제시 함으로서 차후에는 보다 나은 선택을 할 수 있게 유도하게 된다 [6][7].

[그림1]은 강화 학습의 학습 과정을 간략히 도식화 한 것이다. 즉, 강화 학습은 현재의 시간(t)에 대한 환경의 정보(s_t)를 통해 현재 행동 가능한 모든 행동($A(s_t)$)에 대한 정보를 에이전트에 전달한다.



[그림 1] 강화 학습

에이전트는 최적의 행동(a)을 선택하고, 이로부터 변경되는 새로운 상태 정보(s_{t+1})와 그로부터 나오는 보상(r_t)을 받는다. 즉, 학습한 지식과 받은 보상간의 차이를 고려하여 학습하지만, 대부분의 강화 학습 에이전트는 최적의 행동(a)을 선택할 때 보상(r_t)을 같이 구하여 학습한다.

강화 학습은 이러한 과정을 반복하여 수행하며, 중요한 점은 보다 나은 방향으로 학습이 되어야 한다는 것이다. 즉, 강화학습 에이전트는 하나의 최종상태를 가지는 MDP들 을 위한 보상의 총합($r_0+r_1+...+r_n$)이나 최종 상태가 아닌 MDP 들을 위한 보상의 총합($\Sigma_i\gamma^i r_i$)이 최대화 하는 하나의 정책 π:S→A 를 개발해야 한다 (여기서 γ 은 0.0 과 1.0 사이의 Learning rate를 말한다.).

2.2 Q-learning 알고리즘

Q-learning 알고리즘은 Q(s,a)의 값을 통해 최적의 정책(policy)을 찾아내는 강화 학습 알고리즘이다. Q(s,a)의 값은 주어진 상황에서의 행동(a)로 인해 얼마나 좋은지를 수치화 한 것이다. Q(s,a)는 Q-function으로 불리기도 한다.

Q-function은 Q-table을 가지고 있는데 Q-table은 행동 가능한 모든 경우의 행동에 대한 Q-function의 값을 저장하고 있다. 현재 환경의 상태 정보에서 행동 가능한 모든 행동에 대한 Q-function 값을 Q-table에서 얻어, 그 중 최대값을 최적의 행동으로 선택한다. 이렇게 해서 얻은 최적의 행동을 통해 얻어지는 보상 값을 이용해서 기존의 Q-table을 수정함으로써 차후에는 보다 좋은 선택을 할 수 있게 도와준다.

3. 설계

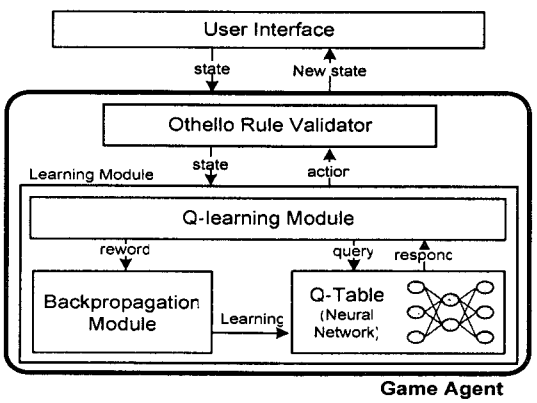
본 시스템은 크게 User interface와 Game 에이전트로 나눌 수 있다. User interface는 강화 학습의 환경 모델에 해당하는 부분이고, Game 에이전트는 강화학습의 에이전트에 해당 된다 [그림 2].

[그림 1

3.1 모듈 별 기능

시스템의 User Interface부분은 게임의 상태 정보를 가지고 있으며 사용자나 에이전트의 필요에 의해 상태 정보를 제공한다.

Game 에이전트는 user inferface 이외에 게임의 진행에 관련된 모든 모듈들을 포함한다. Game 에이전트는 크게 Othello Rule Validator와 Learning 모듈로 구성되며, Learning 모듈은 세부적으로 Q-learning 모듈 Q-table, Backpropagation 모듈로 나누어 진다.



[그림 2] 시스템 구조

각 모듈별 기능은 다음과 같이 설명된다.

- Othello Rule Validator : 오델로 게임의 규칙에 관한 적법성을 판별하게 된다.
- Learning Module : 게임의 전반적인 지능에 관련된 기능을 목적으로 다음 세부모듈들을 가진다.
  - Q-learning Module : 상태정보에서 다음 동작을 추론하고 Q-table의 학습데이터를 작성한다.
  - Q-table : Q-function의 데이터 값을 저장하는 저장공간으로, 신경망으로 구축되어 있다.
  - Backpropagation Module : Q-learning Module로부터 Q-table의 학습 데이터를 받아 역 전파 알고리즘을 이용하여 Q-table을 학습한다.

3.2 학습 알고리즘

Game 에이전트에 사용되는 강화 학습 알고리즘은 Q-learning 알고리즘이다. Q-learning 알고리즘은 Q-table을 통해 얻어지는 데이터를 통해 다음 행동을 선택하기 때문에 Q-table이 반드시 필요하며, 가능한 모든 경우에 대한 데이터가 저장 되어야 한다. 그러나 오델로 게임의 경우 상태 공간은 대략 10^{28}으로 매우 크며, 이것을 Q-table로 구성할 경우 매우 많은 양의 데이터 공간이 필요하다.

본 연구에서는 데이터 공간의 문제를 해결하기 위해 신경망 알고리즘을 Q-table의 역할로 사용하여 해결하였다. 본 시스템에서의 신경망은 모든 상태 및 행동에 대한 Q-function 값을 매핑 하고, 시스템 구조는 입력층, 은닉층, 출력층으로 구성되며, 64개의 뉴론을 가지고 있다. [그림 3]은 신경망을 적용한 Q-learning 에 대한 알고리즘을 표현한 것이다.

Initialize all neural network (NN) weights to small random numbers
Repeat (for each trial)
  Initialize the current state s_t
  Repeat (for each step of trial)
    Observe the current state s_t
    For all actions a' in s use the NN to compute $Q(s_t, a')$
    Select an action a using a policy π
    $Q_{output} \leftarrow Q(s_t, a)$
    Execute action a
    Receive an immediate reward r
    Observe the resulting new state s s_{t+1}
    For all actions a' in s_{t+1} use the NN to compute $\hat{}Q(s_{t+1}, a')$
    Compute Q_{target} Q(s, a)
    Adjust the NN by backpropagating the error (Q_{target} - Q_{output})
    $s_t \leftarrow s_{t+1}$
  Until s is a terminal state

[그림 3] 신경망을 적용한 Q-learning 알고리즘

3.3 수행 시나리오

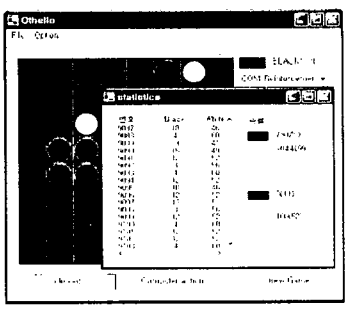
본 논문의 시스템은 게임이 수행이 될 동안 다음과 같은 과정으로 학습되는데, 학습과정을 사용자와 시스템의 두 부분으로 분리하여 보면 다음과 같이 설명된다.

1. 사용자의 차례 일 경우 사용자가 동작을 할 경우 현재 게임의 상태 정보 및 사용자의 동작 정보를 Othello Rule Validator에서 받아 게임의 적법성 여부를 판별하고, 그 결과를 다시 user interface에 보낸다.
2. 컴퓨터의 차례일 경우 현재 게임의 상태 정보를 user interface에서 제공받아 Rule Validator에서 게임의 적합성 여부를 판별한 후 다음과 같이 진행된다.
   가. 현재 게임 상태가 오델로의 게임규칙에 적합한 상태일 경우 learning 모듈에 게임의 상태 정보를 보낸다.
   나. learning 모듈의 Q-learning 모듈은 현재 상태에서 가능 한 모든 행동에 대한 Q-function 값을 Q-table을 통해 구한다.
   다. Q-learning 모듈은 다음 동작상태에 관한 보상값을 계산하여 Othello Rule Validator에게 다음 동작상태를 전달하고, 이 값을 이용하여 새로운 Q-table 데이터 값을 구한 후 Backpropagation 모듈에 전달한다.
   라. Othello Rule Validator 모듈은 Learning 모듈로부터 다음 동작의 상태를 받아 적법성 여부를 판별하여 User interface에 다음 동작에 대한 정보를 준다.
   마. Backpropagation 모듈은 새로운 Q-table 정보를 받아 기존의 Q-table을 학습 시킨다.

4. 구현 및 실험

본 논문의 시스템은 C# 언어를 사용하여 Windows XP, Microsoft .NET Framework 7.1 위에서 구현하였다. [그림 4]는 실험을 수행 후 실험 결과에 대해 분석한 화면이다. 분석 화면은 게임을 상황을 보여주는 보드 판, 게임의 경기 결과를 보여 주는 결과 판으로 구성되어 있다.

실험은 본 본문의 시스템과 동일한 신경망을 가진 역 전파 알고리즘으로 구현된 모듈과 강화 학습 모듈을 학습이 아직 안된 상태에서, 각각 depth 3인 min-max 알고리즘과의 대결을 통하여 측정하였다. 학습의 성공여부는 승수 및 승율로 판단할 수 있게 된다.



[그림 4] 프로그램 수행 화면

4.1 실험 결과

실험은 1회차에 5000번씩 대결을 하며, 모두 5회차로 수행하여 그 결과를 [표1]에 요약 정리하였다. 표에서 앞의 숫자는 승수를 말하며, 괄호안의 숫자는 승율을 나타낸다. 학습 시간은 두 알고리즘 모두 비슷하였으며, 강화 학습의 경우 약 13~37%의 승율인 반면, 역 전파 알고리즘의 경우 9~19%의 승율로 강화 학습이 비교적 우수하게 나타났다. 또한 역 전파 알고리즘은 학습이 진행 갈수록 패턴이 고착화 되어가는 반면 강화 학습은 역 전파 알고리즘에 비해 패턴이 불규칙하였다.

[표 1] 실험 결과

	Reinforcement Learning	Backpropagation
1	942(18.86%)	829(16.58%)
2	1227(26.54%)	305(6.10%)
3	713(14.26%)	972(19.44%)
4	1853(37.06%)	790(15.80%)
5	659(13.18%)	679(13.58%)

5. 결론 및 향후 과제

본 연구에서는 최근 인공기능 게임분야에서 확산되고 있는 강화 학습 알고리즘으로 게임을 설계 및 구현하였다. 본 연구의 시스템을 분석하기 위하여 실험 대상으로는 오델로 게임을 선정하였고, 성능측정을 위하여 신경망의 역 전파 알고리즘과 비교.분석 하였다. 두 알고리즘의 단순비교가 어렵기 때문에 비교의 대상으로 min-max알고리즘을 구현하여 두 알고리즘과 각각 대결하는 방식을 수행하였고, 실험의 결과는 강화학습기반의 시스템이 비교적 우수한 결과를 제공하였다. 특히 강화 학습 알고리즘은 역 전파 알고리즘에 비해 그 학습 율과 패턴의 불규칙성이 커 게임 적용시 보다 자연스러운 결과를 제시할 수 있을 것이다.

향후 연구로는 오델로 이외의 다양한 게임에의 적용과 강화학습 알고리즘의 학습율 향상에 관한 연구가 진행되어야 할 것 이다.

참고문헌

[1] 이만재, "게임에서의 인공지능 기술", 정보처리 학회지, 제 9권 3호, p69~p76, 2002.

[2] Johannes Fürnkranz, "Machine Learning in Game Playing: A Survey" In J. Fürnkranz and M. Kubat (eds.), *Machines that Learn to Play Games*, pp.11-59, Nova Science Publishers, 2001

[3] Michael Pfeiffer, "Reinforcement learning of strategies for settlers of CATAN" available at http://eprints.pascal-network.org/archive/00000425/01/LAG-37_pfeiffer_2004.pdf, 2004

[4] R. Reed, "Pruning algorithms—A survey," IEEE Trans. Neural Networks, vol. 4, pp. 740 747, Sept. 1993.

[5] G.J. Tesauro, "Temporal Difference Learning and TD-Gammon", Communications of the ACM 38, 58-68, 1995.

[6] 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.

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