미로 환경에서 최단 경로 탐색을 위한 실시간 강화 학습

김 병 천[†] · 김 삼 근[††] · 윤 병 주[†††]

요 약

강화 학습(reinforcement learning)은 시행-착오(trial-and-error)를 통해 동적 환경과 상호작용하면서 학습을 수행하는 학습 방법으로, 실시간 강화 학습(online reinforcement learning)과 지연 강화 학습(delayed reinforcement learning)으로 분류된다. 본 논문에서는 미로 환경에서 최단 경로를 빠르게 탐색할 수 있는 실시간 강화 학습 시스템(ONRELS : ONline REinforcement Learning System)을 제안한다. ONRELS는 현재 상태에서 상태전이를 하기 전에 선택 가능한 모든 (상태-행동) 쌍에 대한 평가 값을 갱신하고 나서 상태전이를 한다. ONRELS는 미로 환경의 상태 공간을 압축(compression)하고 나서 압축된 환경과 시행-착오를 통해 상호 작용하면서 학습을 수행한다. 실험을 통해 미로 환경에서 ONRELS는 TD-오류를 이용한 Q-학습과 $TD(\lambda)$를 이용한 $Q(\lambda)$-학습보다 최단 경로를 빠르게 탐색할 수 있음을 알 수 있었다.

Online Reinforcement Learning to Search the Shortest Path in Maze Environments

Byung-Cheon Kim[†] · Sam-Keun Kim[††] · Byung-Joo Yoon[†††]

ABSTRACT

Reinforcement learning is a learning method that uses trial-and-error to perform learning by interacting with dynamic environments. It is classified into online reinforcement learning and delayed reinforcement learning. In this paper, we propose an online reinforcement learning system (ONRELS : ONline REinforcement Learning System). ONRELS updates the estimate-value about all the selectable (state, action) pairs before making state-transition at the current state. The ONRELS learns by interacting with the compressed environments through trial-and-error after it compresses the state space of the maze environments. Through experiments, we can see that ONRELS can search the shortest path faster than Q-learning using TD-error and $Q(\lambda)$-learning using $TD(\lambda)$ in the maze environments.

1. 서 론

M. L. Minsky에 의해 소개된 강화 학습(reinforcement learning)은 심리학(psychology) 분야에서 동물의 학습을 연구하는 과정에서 기원하였으며[1, 2], 동적 프로그래밍(dynamic programming)과 교사 학습을 혼합한 형태의 학습 방법으로서 학습을 수행하는 에이전트(agent)는 에이전트의 외부에 존재하는 환경과 시행-착오(trial-and-error)를 통해 상호 작용(interaction)하면서 학습한다[3, 4]. 그러므로 강화 학습은 동적 환경에서 학습을 하기 위해 널리 이용되고 있으며[5-7], 강화 학습을 위해 제시된 많은 강화 학습들은 현재 상태에서 선택한 (상태-행동) 쌍의 값을 언제 갱신할 것인가에 따라 실시간 강화 학습(online reinforcement learning)과 지연 강화 학습(delayed reinforcement learning)으로 분류된다[8].

실시간 강화 학습은 현재 상태에서 상태전이를 하기 전에 현재 상태에서 선택한 (상태-행동) 쌍에 대한 평가 값을 갱신하는 학습 방법이다. 그러나 실시간 강화 학습을 위해 제안된 대부분의 강화 학습들은 현재 상태에서 선택된 (상태-행동) 쌍에 대한 평가 값만을 갱신하기 때문에 진정한 의미의 실시간 강화 학습이라 할 수 없다[9]. 지연 강화 학습은 현재 상태에서 상태전이를 하기 전에 현재 상태에서 선택한 (상태-행동) 쌍에 대한 평가 값을 목표 상태를 발견하고 나서 갱신하는 학습 방법이다. 그러므로 지연 강화 학습은 최적 값-함수(optimal value-function)에 매우 느리게 수렴하

† 종신회원 : 한경대학교 컴퓨터공학과 교수
†† 정 회 원 : 한경대학교 컴퓨터공학과 교수
††† 종신회원 : 명지대학교 컴퓨터공학과 교수
논문접수 : 2001년 9월 28일, 심사완료 : 2002년 2월 8일

는 단점이 있다[10].

본 논문에서는 미로 환경에서 최단 경로를 빠르게 탐색할 수 있는 실시간 강화 학습 시스템(ONRELS : ONline REinforcement Learning System)을 제안하였다. ONRELS는 미로 환경에서 최단 경로를 탐색하기 위해 먼저, 미로 환경을 압축하여 압축된 미로 환경과 시행-착오를 통해 상호 작용하면서 최단 경로가 될 수 있는 후보 상태들을 탐색하여 미로 환경과 사상(mapping)시킨다. 그리고 나서 사상된 상태들에 대해서만 미로 환경과 시행-착오를 통해 상호 작용하면서 학습을 수행하여 최단 경로를 탐색한다. ONRELS의 학습 방법은 현재 상태에서 다른 상태로 상태전이를 하기 전에 현재 상태에서 선택 가능한 모든 (상태-행동) 쌍에 대한 평가 값을 갱신하고 나서 다음 상태로 상태전이를 하는 실시간 강화 학습 시스템이다.

R. S. Sutton이 제안한 미로 환경[11]에 대해 실험한 결과 ONRELS는 TD-오류(Temporal Difference error)를 이용한 Q-학습(Quality-learning)[12]과 $TD(\lambda)$를 이용한 $Q(\lambda)$-학습[13]보다 최단 경로를 빠르게 탐색할 수 있음을 알 수 있었다.

본 논문은 5장으로 구성되어 있으며 각 장의 주요 내용은 다음과 같다. 2장에서는 강화 학습을 위해 가장 널리 이용되고 있는 Q-학습과 $Q(\lambda)$-학습의 학습 방법에 대하여 기술하고, 3장에서는 본 논문에서 제안된 ONRELS의 학습 방법과 특징에 대하여 설명한다. 4장에서는 강화 학습의 학습 성능을 평가하기 위해 널리 이용되고 있는 R. S. Sutton이 제안한 표준 문제(test-bead)인 미로 환경에 대하여 ONRELS와 TD-오류를 이용한 Q-학습 그리고 $TD(\lambda)$를 이용한 $Q(\lambda)$-학습 등과 학습 성능을 비교 분석한다. 그리고 5장에서는 결론과 함께 향후 연구 과제를 제시한다.

2. 관련 연구

동적 환경에서 학습을 하기 위해 가장 널리 이용되고 있는 강화 학습은 TD-오류를 이용한 Q-학습과 $TD(\lambda)$를 이용한 $Q(\lambda)$-학습 등이 있다.

2.1 Q-학습

C. J. C. H. Watkins가 제안한 Q-학습은 강화 학습을 위해 가장 널리 이용되고 있는 학습 방법으로서 통계적 동적 프로그래밍(stochastic dynamic programming)에 근거한 학습 방법이다. Q-학습은 현재 상태(s_t)에서 어떤 행동(a_t)를 수행하였을 때 받은 강화 값(reinforcement value)에 대한 근사 값(approximation value)을 (상태-행동) 쌍에 대한 Q-함수 $Q(s_t, a_t)$에 할당한다. 그리고 나서 다음 상태 s_{t+1}의 (상태-행동) 쌍에 대한 Q-함수 $Q(s_{t+1}, a_{t+1})$가 최대가 되는 행동 a_{t+1}을 선택하여 현재 상태의(상태-행동) 쌍에 대한 Q-함수 값과의 차(difference) 즉, TD-오류를 이용하여 식 (2.1)과 같이 갱신하면서 학습한다.

$$Q(s_t, a_t) = (1-\alpha) \cdot Q(s_t, a_t) + \alpha \cdot \delta_t \tag{2.1}$$

식 (2.1)에서 α는 학습율(learning rate)이고, δ_t는 현재 상태에서 선택한 (상태-행동)에 대한 TD-오류로서 식 (2.2)와 같이 계산된다.

$$\delta_t = r_{t+1} + \gamma\{\max_{a \in A(s_t)} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)\} \tag{2.2}$$

(여기에서, r_{t+1}은 강화 값이고, $\gamma(0 < \gamma < 1)$는 할인율(discount rate)이다)

식 (2.2)에서 $A(s_t)$는 학습을 수행하는 에이전트가 상태 s_t에서 선택 가능한 행동들의 집합이다.

2.2 $Q(\lambda)$-학습

$Q(\lambda)$-학습은 TD-오류와 적합도(eligibility factor)를 이용한 학습 방법으로서 적합도란 현재 상태에서 선택한 (상태-행동) 쌍이 얼마나 접합한가를 의미하는 척도이다[14]. $Q(\lambda)$-학습은 현재 상태에서 선택한 (상태-행동) 쌍에 대한 Q-함수 $Q(s_t, a_t)$를 TD-오류와 적합도를 이용하여 식 (2.3)과 같이 갱신한다.

$$Q(s_t, a_t) = (1-\alpha)\, Q(s_t, a_t) + \alpha \delta_t e(s_t, a_t) \tag{2.3}$$

식 (2.3)에서 δ_t는 TD-오류이고, $e(s_t, a_t)$는 현재 상태 s_t에서 행동 a_t를 선택하였을 때의 적합도이다. 현재 상태에서 선택한 (상태-행동) 쌍에 대한 적합도 $e(s_t, a_t)$는 식 (2.4)d와 같이 계산된다.

$$e(s_t, a_t) = \begin{cases} \gamma\lambda e(s_{t-1}, a_{t-1}) + 1 & \\ \qquad \text{if } Q(s_t, a_t) = \max\limits_{a \in A(s_t)} Q(s_t, a) & \\ 0 \qquad \textit{otherwise} & \end{cases} \tag{2.4}$$

식 (2.4)에서 $\gamma\lambda(0<\gamma<1\,,\ 0<\lambda<1)$는 감소율(decay factor)을 의미하며, 적합도 $e(s_t, a_t)$는 에이전트가 탐색 과정에서 선택한 (상태-행동) 쌍이 얼마나 좋은가에 대한 기록이라 할 수 있으며, 만일 현재 상태에서 선택 가능한 (상태-행동) 쌍들 중에서 가장 큰 Q-값을 갖는 (상태-행동) 쌍을 선택한 경우 적합도를 이전 상태에서 선택한 (상태-행동) 쌍에 대한 접합도보다 1만큼 증가시키고, 그렇지 않은 경우 적합도를 0으로 한다.

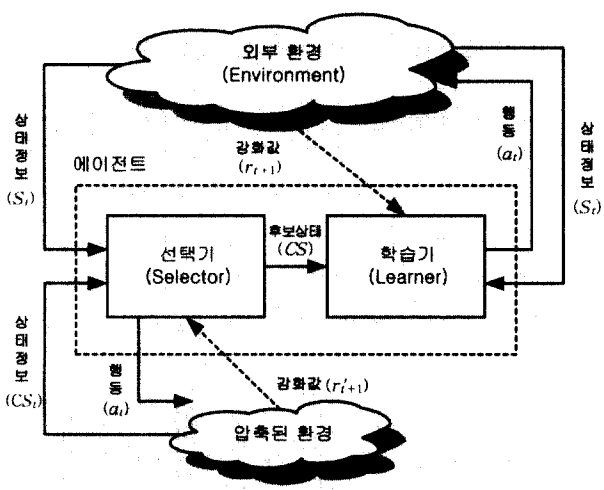
일반적으로 강화 학습에서 상태전이를 하기 위해 가장 적당한 행동을 선택하는 방법은 식 (2.5)와 같은 볼쯔만(Boltzmann) 확률 분포에 따라 특정 행동을 선택하는 방법이 널리 이용되고 있다.

$$p(\bar{a} \mid s) = \frac{e^{\frac{Q(s,\bar{a})}{T}}}{\sum_{a \in A(s)} e^{\frac{Q(s,a)}{T}}} \quad (2.5)$$

식 (2.5)에서, T는 행동 선택의 임의성(randomness) 정도를 제어하는 온도변수(temperature variable)이다.

3. 실시간 강화 학습 시스템(ONRELS)

미로 환경에서 시작 상태에서 목표 상태까지의 최단 경로를 빠르게 탐색하기 위해 본 논문에서 제시된 실시간 강화 학습 시스템 ONRELS는 (그림 3.1)과 선택기와 학습기로 구성되어 있다.



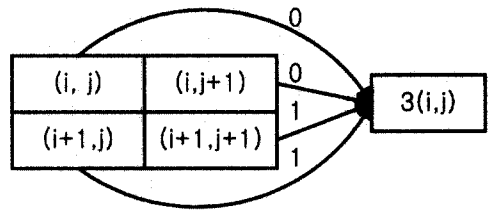
(그림 3.1) ONRELS의 구조

3.1 선택기

선택기는 미로 환경을 압축하여 압축된 미로 환경으로 재구성하는 압축 모듈(compression module)과 압축된 미로 환경과 시행-착오를 통해 상호 작용하면서 실시간 강화 학습을 수행하여 최단 경로가 될 수 있는 후보 상태들을 선택하는 학습 모듈(learning module)로 구성되어 있다.

3.1.1 압축 모듈

압축 모듈은 주어진 미로 환경의 상태 공간 $S(n \times n)$을 $S'(n/2 \times n/2)$으로 축소된 미로 환경으로 압축하는 모듈이다. 압축 방법은 (그림 3.2)와 같이 주어진 미로 환경을 좌에서 우로, 위에서 아래로 (2×2)의 상태 공간을 장애물의 유무에 따라 (1×1)로 압축하여 새로운 상태 값을 할당한다.



(그림 3.2) 압축 방법

압축된 환경의 각 상태 값은 다음 상태로 상태전이 할 수 있는가 또는 할 수 없는가에 대한 정보를 나타낸다. 예를 들어 현재 상태의 상태 값이 12(■)인 경우 에이전트는 북쪽 방향으로는 상태전이를 할 수 없고, 동쪽 방향은 다음 상태 값이 0, 2, 4, 6, 8, 10, 12 또는 14인 경우에만 상태전이를 할 수 있다. 그리고 남쪽 방향은 다음 상태 값이 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 또는 11인 경우에만 상태전이를 할 수 있고, 서쪽 방향은 다음 상태 값이 0, 1, 2, 3, 4, 5, 8, 9, 12 또는 13인 경우에만 상태전이를 할 수 있다.

3.1.2 학습 모듈

학습 모듈은 최단 경로가 될 수 있는 후보 상태들을 선택하기 위해 압축된 미로 환경에서 실시간 강화 학습을 수행하는 학습 모듈이다. 압축된 미로 환경에서 학습 모듈은 먼저, (식 3.1)을 이용하여 현재 상태에서 선택 가능한 모든 (상태-행동) 쌍에 대한 $CQ'(s_t, a_t)$-값을 갱신한다. 그리고 나서, 현재 상태에서 CQ'-값이 최대인 (상태-행동) 쌍을 식 (3.2)와 같이 갱신한다.

For all s_t, a $(a \in A(s_t))$

$$CQ'(s_t, a_t) = (1-\alpha)CQ'(s_t, a_t) + \alpha[r'_{t+1} + \gamma\{\max_{a \in A(s_t)} CQ'(s_{t+1}, a_{t+1}) - CQ'(s_t, a_t)\}] \quad (3.1)$$

식 (3.1)에서 r'_{t+1}는 압축된 미로 환경으로부터 받는 강화 값이고, $CQ'(s_t, a_t)$는 압축된 미로 환경에서 에이전트가 현재 상태에서 선택한 (상태-행동) 쌍에 대한 평가 값이다.

$$CQ'(s_t, a_t) = (1-\alpha)CQ'(s_t, a_t) + \alpha\delta_t ce(s_t, a_t) \quad (3.2)$$

식 (3.2)에서 δ_t는 현재 상태에서 선택한 (상태-행동) 쌍과 다음 상태의 (상태-행동) 쌍과의 TD-오류로서 식 (3.3)과 같이 계산된다.

$$\delta_t = r'_{t+1} + \gamma\{\max_{a\in A(s_t)} CQ'(s_{t+1}, a_{t+1}) - CQ'(s_t, a_t)\} \quad (3.3)$$

식 (3.3)에서 $ce(s_t, a_t)$는 압축된 상태 공간에서 선택한 (상태-행동) 쌍이 얼마나 적합한가를 나타내는 적합도이며, 식 (3.4)와 같이 갱신된다.

$$ce(s_t, a_t) = \begin{cases} 1 & \text{if } s = s_t \text{ and } a = a_t \\ \gamma\lambda\, ce(s_{t-1}, a_{-1}) & otherwise \end{cases} \quad (3.4)$$

식 (3.4)는 에이전트가 탐색 과정에서 선택한 현재 상태의 (상태-행동) 쌍이 이미 선택된 (상태-행동) 쌍일 경우 현재 상태에서 선택한 (상태-행동) 쌍에 대한 적합도를 1로 하고, 그렇지 않은 경우 현재 상태에서 선택한 (상태-행동) 쌍에 대한 적합도를 $\gamma\lambda$만큼씩 감소시키는 역할을 한다. 그러므로 매우 긴 에피소드를 갖는 에피소딕 환경(episodic environment)뿐만 아니라 어떤 에피소드도 갖지 않는 지속적인 환경(continual environment)에서도 학습이 가능하다. 압축된 미로 환경에서 TD-오류와 적합도를 이용하여 최단 경로가 될 수 있는 후보 상태를 선택하기 위한 실시간 강화 학습을 수행하는 학습 모듈의 학습 절차는 (그림 3.3)과 같다.

```
Learning_Module()
{
  Compress_Environment() ;
  Initialize CQ'(s, a) and ce(s, a) arbitrarily, for all s, a;
    Repeat {
    Repeat {
      Update_Qvalue (s_t);
      Take an action a_t;
      if (observable s_{t+1}) {
        Observe reward r'_{t+1}, next state s_{t+1};
        Update_EQvalue (s_t, a_t, r_{t+1}, s_{t+1});
      }
      s_t = s_{t+1}; a_t = a_{t+1};
    } Until( s_t is goal state)
  } Until(a certain number of episodes)
  Select_Candidate_States() ;
  Mapping() ;
}
```

(그림 3.3) 학습 모듈의 학습 알고리즘

(그림 3.3)에서 Update_Qvalue(s_t) 함수는 식 (3.1)과 같이 압축된 환경에서 선택 가능한 모든 (상태-행동) 쌍에 대한 평가 값을 갱신하는 함수이며, Update_EQvalues(s_t, a_t, r_{t+1}, s_{t+1}) 함수는 식 (3.2)를 이용하여 현재 상태에서 선택된 (상태-행동) 쌍에 대한 평가 값을 갱신하는 함수이다. 그리고 Select_Candidate_States() 함수는 압축된 환경에서 충분히 학습한 후 최단 경로가 될 수 있는 예상 경로를 식 (3.5)와 같이 결정한다.

For all s_t, a

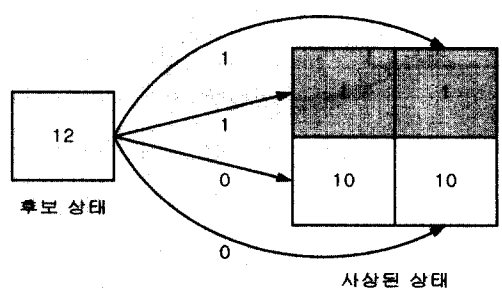
$$CS(s_t, a_t) = \max_{a\in A(s_t)} CQ'(s_t, a_t) \quad (3.5)$$

식 (3.5)에서 $CS(s_t, a_t)$는 압축된 상태 공간에서 최단 경로가 될 수 있는 후보 상태로서 현재 상태에서 선택 가능한 모든 (상태-행동) 쌍들 중에서 $CQ'(s_t, a_t)$ 값이 최대인 (상태-행동) 쌍을 선택한다. 학습 모듈이 예상 경로를 선택하기 위한 처리 절차는 (그림 3.4)와 같다.

```
While( s_t ≠ goal state) {
  For all s_t, a
    Search the max_{a∈A(s_t)} CQ'(s_t, a_t) value
  Observe s_{t+1};
  s_t = s_{t+1}; a_t = a_{t+1};
}
```

(그림 3.4) Select_Candidate_States() 함수

(그림 3.3)의 mapping() 함수는 압축된 상태 공간에서 선택된 후보 상태들을 미로 환경과 사상(mapping)시키는 함수이다. 사상 과정은 압축된 상태 공간에서 선택된 (1×1)이 압축되지 않은 상태 공간의 (2×2)로 (그림 3.5)와 같이 사상되며, 사상된 상태들은 장애물의 유무에 따라 특정 상태 값(장애물 : 1, 비 장애물 : 10)을 갖는다.



(그림 3.5) 후보 상태들에 대한 사상

3.2 학습기

학습기는 미로 환경에서 사상된 상태들 즉, 상태 값이 10 (s_t=10)인 상태들에 대해서만 실시간 강화 학습을 수행하여 시작 상태에서 목표 상태까지 최단 경로를 탐색하는 역할을 한다. 학습기는 먼저, 현재 상태에서 선택 가능한 모든 (상태-행동) 쌍에 대한 평가 값을 식 (3.6)과 같이 갱신한다.

For all s_t, a

$$Q(s_t, a_t) = (1-\alpha)Q(s_t, a_t) + \alpha[r_{t+1} + \gamma\{\max_{a \in A(s_t)} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)\}] ; \quad (3.6)$$

그리고 나서 식 (3.7)과 같이 TD-오류와 적합도를 이용하여 현재 상태의 (상태-행동) 쌍에 대한 평가 값을 갱신하면서 실시간 강화 학습을 수행한다.

```
Initialize Q(s, a) and e(s, a) arbitrarily for all s, a;
Repeat {
  Set the start state s_t;
    Repeat {
      Update_Qvalues (s_t);
      Take an action a_t ;
      Observe s_{t+1}, r_{t+1};
      Choose a_{t+1} from s_{t+1};
      Update_EQvalues ( s_t, a_t, r_{t+1}, s_{t+1});
      s_t = s_{t+1}; a_t = a_{t+1};
    } Until( s_t is goal state)
} Until(a certain number of episodes)
```

(그림 3.6) 학습기의 학습 알고리즘

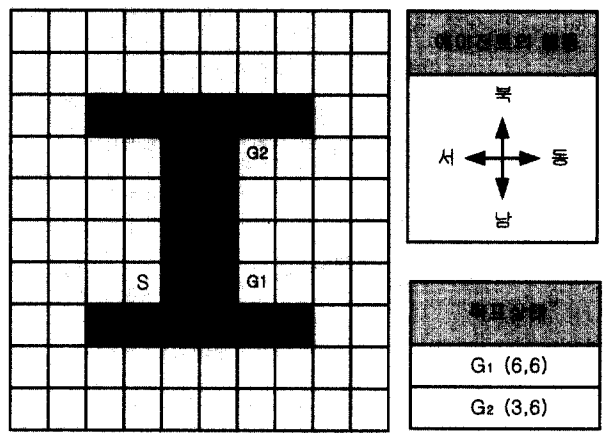
$$Q(s_t, a_t) = (1-\alpha)Q(s_t, a_t) + \alpha\delta_t e(s_t, a_t) \quad (3.7)$$

식 (3.7)에서 δ_t는 TD-오류이고, $e(s_t, a_t)$는 현재 상태에서 선택한 (상태-행동) 쌍에 대한 적합도로서 학습 모듈의 적합도와 같이 계산된다. 학습기는 식 (3.5)와 식 (3.6)을 이용하여 미로 환경에서 최단 경로를 탐색하기 위해 (그림 3.6)과 같이 실시간 강화 학습을 수행한다.

4. 실험 및 분석

ONRELS의 학습 성능을 평가하기 위해 R. S. Sutton이 제안한 (그림 4.1)과 같은 (10×10) 미로 환경을 이용하였다. 미로 환경에서 에이전트의 학습 목표는 시작 상태(S)에서 에이전트가 수행할 수 있는 행동 ($a \in A(s_t)$)들은 {동, 서, 남, 북}이고 이들 행동들 중에서 특정 행동을 선택하여 목표 상태(G)에 도달하는 최단 경로를 탐색하는 것이다. (그림 4.1)에서 S는 시작 상태, G_1과 G_2는 서로 다른 목표 상태이며, 음영 부분은 상태 값 1을 갖는 장애물, 음영 부분이 아닌 부분은 상태전이가 가능한 상태로서 상태 값은 0이다. 본 논문에서 제안한 ONRELS의 학습 성능을 평가하기 위해 TD-오류를 이용한 Q-학습과 $TD(\lambda)$를 이용한 $Q(\lambda)$-학습 그리고 ONRELS의 학습율을 0.8 (α=0.8), 할인율을 0.9 (γ=0.9), 그리고 감소율(decay factor)을 0.9 (λ=0.9)로 고정하여 각 학습 방법들을 20회씩 실행한 후 결과를 비교 분석하였다.

ONRELS가 후보 상태를 선택하기 위해 선택기의 압축 모듈이 (그림 4.1)과 같이 주어진 미로 환경을 압축한 결과는 (그림 4.2)와 같다.



(그림 4.1) 미로 환경

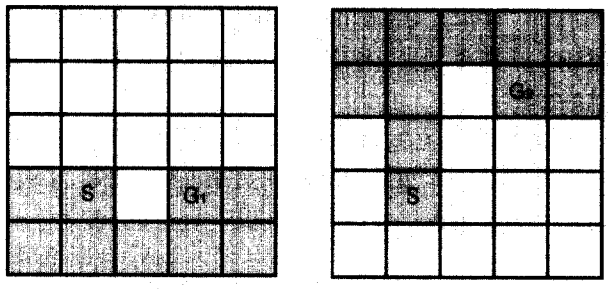
(그림 4.2)와 같이 미로 환경을 압축하고 나서, 압축된 미로 환경에서 선택기의 학습 모듈이 목표 상태 G_1과 G_2까지 최단 경로가 될 수 있는 후보 상태를 선택하기 위해 실시간 강화 학습을 수행한 결과는 (그림 4.3)과 같다.

0	0	0	0	0
0	12	15	12	0
0	0	15	0	0
0	12	12	12	0
0	0	0	0	0

(그림 4.2) (그림 4.1)의 미로 환경에 대한 압축 결과

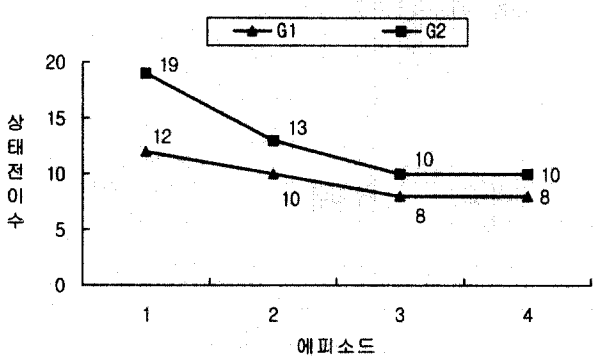
학습 모듈은 (그림 4.3)의 목표 상태 G_1과 G_2까지 최단 경로가 될 수 있는 후보 상태를 선택하기 위해 (그림 4.4)와 같이 각각 22번과 32번의 상태전이가 발생하였다.

선택기의 학습 모듈이 학습을 완료한 후 선택된 (그림 4.3)과 같은 후보 상태들을 미로 환경과 사상시키며, 사상 결과는 (그림 4.5)와 같다.

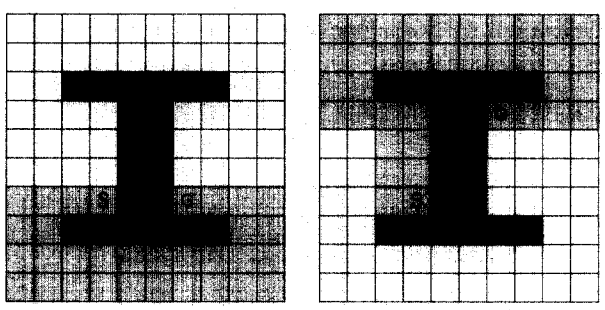


(a) G_1에 대한 예상 경로 (b) G_2에 대한 예상 경로

(그림 4.3) 선택기의 학습 결과

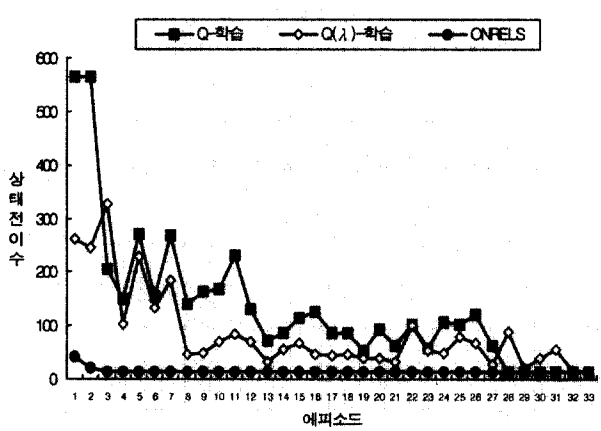


(그림 4.4) 학습 모듈의 상태전이 수



(a) G_1에 대한 사상 결과 (b) G_2에 대한 사상 결과

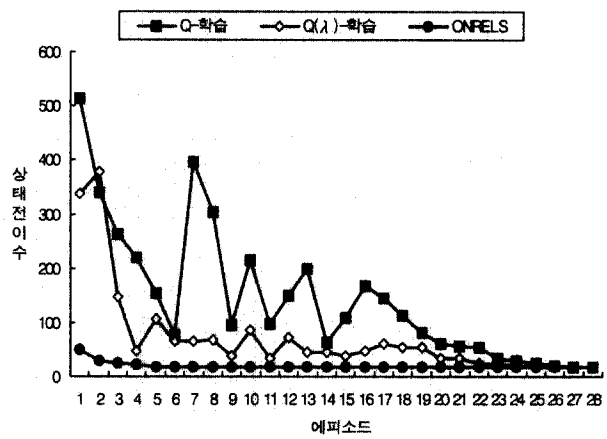
(그림 4.5) 선택기의 사상 결과



(그림 4.6) 목표 상태 G_1에 대한 학습 결과

학습기는 최단 경로를 탐색하기 위해 (그림 4.4)와 같이 사상된 상태들에 대해서만 학습을 수행한다. ONRELS와 Q-학습, 그리고 $Q(\lambda)$-학습들이 목표 상태 G_1까지 최단 경로를 탐색하기 위한 학습 결과는 (그림 4.6)과 같다.

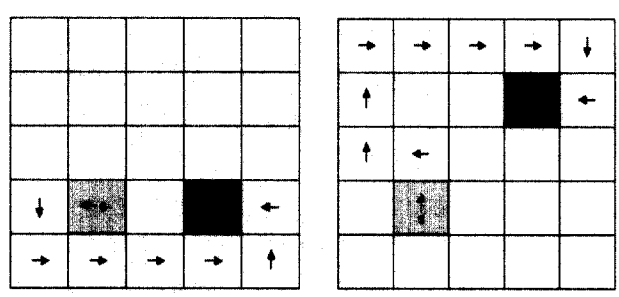
(그림 4.5)에 나타난 것처럼 Q-학습은 4417번, $Q(\lambda)$-학습은 2,853번의 상태 전이 후에 시작 상태에서 목표 상태 G_1까지의 최단 경로를 탐색할 수 있었다. 그러나 ONRELS는 단지 62번의 상태 전이 후에 최단 경로를 탐색할 수 있었다. ONRELS와 Q-학습, 그리고 $Q(\lambda)$-학습들이 목표 상태 G_2까지 최단 경로를 탐색하기 위한 학습 결과는 (그림 4.7)과 같다.



(그림 4.7) 목표 상태 G_2에 대한 학습 결과

(그림 4.7)에 나타난 것처럼 Q-학습은 3977번, $Q(\lambda)$-학습은 2,585번의 상태 전이 후에 시작 상태에서 목표 상태 G_1까지의 최단 경로를 탐색할 수 있었다. 그러나 ONRELS는 단지 124번의 상태 전이 후에 최단 경로를 탐색할 수 있었다.

미로 환경에서 ONRELS는 목표 상태 G_1과 G_2까지 최단 경로를 탐색하기 위해 실시간 강화 학습을 수행한 결과는 (그림 4.8)과 같다.



(a) G_1에 대한 학습 결과 (b) G_2에 대한 학습 결과

(그림 4.8) G_1과 G_2에 대한 학습 결과

5. 결론 및 향후 연구방향

본 논문에서 제안한 ONRELS는 미로 환경에서 <표 5.1>과 같이 목표 상태 G_1까지의 최단 경로를 탐색하기 위해 먼저, 후보 상태를 선택하기 위해 압축된 미로 환경에서 22번의 상태 전이가 발생하였다. 그리고 나서 미로 환경에서 최단 경로를 탐색하기 위해 62번의 상태 전이가 발생하였다. 그러므로 ONRELS는 G_1까지의 최단경로를 탐색하기 위해 총 84번의 상태 전이가 발생하였다. 또한 목표 상태 G_2까지의 최단 경로를 탐색하기 위해 압축된 미로 환경에서 후보 상태를 선택하기 위해 32번의 상태전이가 발생하였고, 최단 경로를 탐색하기 위해 124번의 상태전이가 발생하였다. 그러므로 ONTELS는 G_2까지의 최단 경로를 탐색하기 위해 총 156번의 상태전이가 발생하였다.

〈표 5.1〉 미로 환경에서의 학습 결과에 대한 비교

학습방법 \ 목표상태	G_1	G_2
Q-학습	4,417	3,977
$Q(\lambda)$-학습	2,853	2,585
ONRELS	62	124

본 논문에서 제안한 ONRELS는 미로 환경에서 최단 경로를 탐색하기 위해 TD-오류를 이용한 Q-학습과 $TD(\lambda)$를 이용한 $Q(\lambda)$-학습보다 최단 경로를 매우 빠르게 탐색할 수 있음을 알 수 있었다. 이는 상태 공간을 압축하여 학습을 수행하기 때문이다.

본 연구를 통해 상태 공간을 압축하여 학습할 경우, 학습을 수행하는 에이전트가 탐색해야하는 상태 공간과 에이전트가 참조해야하는 기억 공간이 줄어들기 때문에 최단 경로를 매우 빠르게 탐색할 수 있음을 알 수 있었다. 또한 에이전트가 상태전이를 하기 전에 현재 상태에서 선택 가능한 모든 (상태-행동) 쌍에 대한 평가 값을 갱신하고 나서 상태전이를 하는 실시간 강화 학습이 목표 상태에 빠르게 수렴함을 알 수 있었다. 제안된 방법은 단지 미로 환경에서 효율적으로 학습을 수행하기 위해 상태 공간을 압축하였기 때문에 다른 환경에 그대로 적용하기에는 무리가 따른다는 단점이 있다. 그러나 제안된 실시간 강화 학습의 학습 방법은 미로 환경이 아닌 다른 동적 환경에도 적용 가능하다.

참 고 문 헌

[1] M. L. Minsky, *Theory of Neural-Analog Reinforcement Systems and Application to the Brain-Model Problem*, Ph.D. Thesis, Princeton University, Princeton, 1954.

[2] M. L. Minsky, "Steps towards artificial intelligence," In Proceedings of the Institute of Radio Engineers, 49, pp.8-30, 1961.

[3] A. G. Barto, D. A. White and D. A. Sofge, "Reinforcement learning and adaptive critic methods," Handbook of Intelligent Control, pp.469-491, 1992.

[4] A. W. Moore and C. G. Atkeson, "Prioritized sweeping : Reinforcement Learning with less data and less real time," Machine Learning, 13, pp.103-130, 1993.

[5] C. W. Anderson, "Learning to control an inverted pendulum using neural networks," IEEE Control Systems Magazine, pp.31-37, 1989.

[6] F. S. Ho, "Traffic flow modeling and control using artificial neural networks," IEEE Control Systems, 16(5), pp.16-26, 1996.

[7] R. H. Crites and A. G. Barto, "Improving Elevator Performance Using Reinforcement Learning," Advances in Neural Information Processing Systems, 8, MIT Press, Cambridge MA, 1996.

[8] G. Rummery and M. Niranjan, "On-line Q-learning using connectionist systems," Technical Report CUED/F-INFENG-TR 166, Cambridge University, U.K., 1994.

[9] J. Peng and R. Williams, "Incremental multi-step Q-learning," Machine Learning, 22, pp.283-290, 1996.

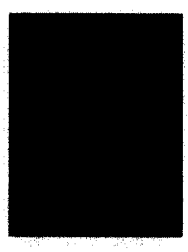
[10] P. Dayan, "Navigating through temporal difference," In Advances in Neural Information Processing Systems, 3, Morgan Kaufmann, 1991.

[11] R. S. Sutton and A. G. Barto, *An Introduction to Reinforcement Learning : An Introduction*, MIT Press, 1998.

[12] G. A. Rummery, *Problem Solving with Reinforcement Learning*, Ph.D. thesis, Cambridge University, 1995.

[13] P. Cichosz, "Truncating temporal differences : On the efficient implementation of $TD(\lambda)$ for reinforcement learning," Journal of Artificial Intelligence Research, 2, pp.287-318, 1995.

[14] S. P. Singh and R. S. Sutton, "Reinforcement Learning with Replacing Eligibility Traces," Machine Learning, 22, pp. 123-158, 1996.



김 병 천

e-mail : bckim@hnu.hankyong.ac.kr

1988년 한남대학교 전자계산학과(공학사)

1990년 숭실대학교 전자계산학과(공학석사)

1999년 명지대학교 컴퓨터공학과(공학박사)

1991년~현재 한경대학교 컴퓨터공학과 부교수

관심분야 : Machine Learning, Knowledge-Based System, Computer Vision

김 삼 근

e-mail : skim@ce.hankyong.ac.kr

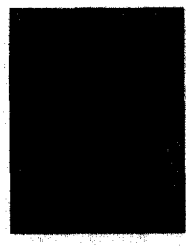
1985년 부산대학교 계산통계학과(이학사)

1988년 숭실대학교 전자계산학과(공학석사)

1998년 숭실대학교 전자계산학과(공학박사)

1992년~현재 한경대학교 컴퓨터공학과 부교수

관심분야 : Neural Network, Data Mining, Web Computing



윤 병 주

e-mail : yoonbj@wh.myongji.ac.kr

1975년 경북대학교 수학과(학사)

1982년 한국과학기술원 전산학과(석사)

1994년 Florida State University 전산학과(박사)

1982년~현재 명지대학교 컴퓨터공학과 교수

관심분야 : Machine Learning, Knowledge-Based System, Hybrid Intelligent Systems