목표상태 값 전파를 이용한 강화 학습

김 병 천† · 윤 병 주††

요 약

동적 환경에서 학습을 수행하기 위해 Q-학습, $TD(0)$-학습, $TD(\lambda)$-학습 등과 같은 강화학습 알고리즘들이 제안되었다. 그러나 대부분의 강화학습 알고리즘들은 목표상태에 도달하였을 때 강화값이 주어지기 때문에 학습 속도가 매우 느린 단점이 있다.

본 논문에서는 미로 환경에서 목표상태에 빠르게 수렴할 수 있는 강화학습 방법을 제안하였다. 제안된 강화학습 방법은 전역학습(global learning)과 지역학습(local learning)으로 분리하여 학습을 수행한다. 전역학습은 replacing eligibility trace 방법을 이용하여 목표상태를 탐색하기 위한 학습이다. 지역학습은 전역학습을 통해 탐색된 목표상태 값을 인접 상태에 전파시킨 후 인접 상태에서 목표상태를 탐색하기 위한 학습이다. 제안한 강화학습 방법은 Q-학습, $TD(0)$-학습, $TD(\lambda)$-학습 등과 같은 강화학습 방법보다 최적 해에 더 빠르게 수렴할 수 있음을 실험을 통해 알 수 있었다.

Reinforcement Learning using Propagation of Goal-State-Value

Byung-Cheon Kim† · Byung-Joo Yoon††

ABSTRACT

In order to learn in dynamic environments, reinforcement learning algorithms like Q-learning, $TD(0)$-learning, $TD(\lambda)$-learning have been proposed. However, most of them have a drawback of very slow learning because the reinforcement value is given when they reach their goal state.

In this thesis, we have proposed a reinforcement learning method that can approximate fast to the goal state in maze environments. The proposed reinforcement learning method is separated into global learning and local learning, and then it executes learning. Global learning is a learning that uses the replacing eligibility trace method to search the goal state. In local learning, it propagates the goal state value that has been searched through global learning to neighboring states, and then searches goal state in neighboring states. We can show through experiments that the reinforcement learning method proposed in this thesis can find out an optimal solution faster than other reinforcement learning methods like Q-learning, $TD(0)$-learning and $TD(\lambda)$-learning.

1. 서 론

강화학습(reinforcement learning)은 학습을 수행하는 에이전트(agent)가 동적 환경(dynamic environment)에 대해 시도와 오류(trial-and-error)에 의해 상호 작용하면서 학습을 수행하기 때문에 상호 작용에 의한 학습(learning by interaction)이라고도 한다[1]. 에이전트는 학습을 수행하는 동안 주어진 환경에서 취할 수 있는 행동(action)을 시도(trial)하며, 그 행동에 대한 스칼라(scalar)형의 강화값(reinforcement value)

※ 이 연구는 일부 정보통신부의 정보통신 우수시범학교 지원사업에 의하여 수행된 것임.
† 정 회 원 : 명지대학교 대학원 컴퓨터공학과/정보통신교육연구센터
†† 종신회원 : 명지대학교 컴퓨터공학과 교수/정보통신교육연구센터
논문접수 : 1999년 1월 5일, 심사완료 : 1999년 2월 27일

을 받아 강화(strengthen)된다. 이와 같은 강화학습은 동적 환경에서도 효율적인 학습이 가능하기 때문에 Cart-Pole 균형 문제[2], 교통신호 제어[3], 엘리베이터 운행 제어[4], 그리고 로봇 이동(robot navigation)[5] 등에 널리 이용되고 있다. 지금까지 제안된 대표적인 강화학습 알고리즘으로는 Q-학습(Quality-learning)[6], $TD(0)$-학습(Temporal-Difference learning)[7], $TD(\lambda)$-학습[8] 등이 있지만, 제안된 대부분의 강화학습 알고리즘들은 언제까지 학습을 해야 최적 해(optimal solution)에 수렴하는가에 대해 명확하지 않을 뿐만 아니라, 학습을 수행하는 에이전트에게 주어지는 강화값은 에이전트가 의도된 목표에 도달할 때에만 주어지기 때문에 최적 해에 매우 느리게 수렴하는 단점이 있다[9].

본 논문에서는 미로 환경(maze environment)에서 사용할 수 있는 강화학습 방법의 한가지로서 위의 결점들을 보완할 수 있는 방법을 제안한다. 제안된 강화학습은 전역학습(global learning)과 지역학습(local learning)으로 분리하여 학습을 수행한다. 전역학습은 replacing eligibility traces 방법[10]을 이용하여 목표상태를 탐색하기 위한 학습이다. 지역학습은 전역학습에서 탐색된 목표상태의 상태 값을 인접 상태에 전파시키고 나서 목표상태와 인접한 상태에서 목표상태를 탐색하기 위해 학습을 수행한다. 제안된 강화학습 방법을 서술의 편의를 위해 GP-강화학습(Goal-state-value Propagation reinforcement learning)이라 명명하였다. GP-강화학습은 학습 수행 기간을 명시하므로 수렴 여부 확인이 용이하며, 전역학습과 지역학습으로 분리하여 탐색하기 때문에 기존의 강화학습 알고리즘 보다 빠르게 수렴하는 특징이 있다. 본 논문의 구성은 다음과 같다. 2장에서는 대표적인 강화학습 알고리즘인 Q-학습, $TD(0)$-학습, $TD(\lambda)$-학습을 요약하며, 3장에서는 본 논문에서 제안한 GP-강화학습에 대하여 설명한다. 4장에서는 GP-강화학습 알고리즘과 Q-학습, $TD(0)$-학습, $TD(\lambda)$-학습을 미로 환경에 적용하여 각 강화학습 알고리즘에 대한 성능을 분석하며, 5장에서 결론 및 향후 연구 과제를 제시한다.

2. 관련 연구

미로 환경을 유한 상태(finite state)의 MDP(Markov Decision Problem)라 할 수 있으며[11], 이런 환경에서 최단 경로를 찾기에 적합한 강화학습 방법은 Q-학습, $TD(0)$-학습, $TD(\lambda)$-학습 등의 방법이 제안되었다.

2.1 Q-학습

Watkins가 제안한 Q-학습은 강화학습을 위해 널리 이용되는 학습 방법이다. Q-학습의 본질은 학습을 수행하는 에이전트가 현재 상태에서 수행 가능한 각 행동 ($a \in A$)에 관련된 예측(prediction)을 표현하는 Q-함수를 학습하는데 있으며, Q-함수는 식(1)과 같이 표현된다.

$$Q(x_t, a_t) = (1-\alpha)Q(x_t, a_t) + \alpha[r_t + \gamma V(x_{t+1})] \quad (1)$$

여기서, $Q(x_t, a_t)$는 현재 상태의 상태-행동 쌍에 대한 Q 값이고, α는 학습율(learning rate), r_t는 강화값, γ는 할인율(discount rate), 그리고 $V(x_{t+1})$는 다음 상태에 대한 평가(estimate)이다. 식(1)에서 Q-함수의 목표는 외부 환경으로부터 받는 강화값을 최대화하는 것이다. 그러므로 시간 t 에서 어떤 상태에 대한 $V(x_t)$의 평가(estimate)는 식(2)와 같다.

$$V(x_{t+1}) = \max_{a \in A(x_{t+1})} Q(x_{t+1}, a) \quad (2)$$

식(2)를 이용한 Q-학습에서 Q-함수 값을 저장하기 위해 look-up 테이블을 사용하였을 때 유한 상태의 MDP에 수렴하게 되는데, Q-함수가 수렴하면 최적의 정책(optimal policy)은 각 상태에서 가장 큰 Q-값을 갖는 행동을 선택하여 주어진 목표상태를 찾을 수 있다. 식(1)과 식(2)를 이용하는 Q-학습 알고리즘은 (그림 1)과 같다.

1. Initialize $Q(x_t, a)$ value functions arbitrarily
2. Initialize the environment : set a current state x_t
3. Select an action following a certain policy (e.g. Bolzmann probability distribution)
4. Take an action a_t and observe reinforcement value r_t
5. Update the estimate values $Q(x_t, a)$ as follows :
5.1 $V(x_{t+1}) = \max_{a \in A(x_{t+1})} Q(x_{t+1}, a)$
5.2 $Q(x_t, a_t) = (1-\alpha)Q(x_t, a_t) + \alpha[r_t + \gamma V(x_{t+1})]$
6. Let $x_t = x_{t+1}$

7. Go to step 3 until the state x_t is a goal state
8. Repeat steps 2 to 7 for a certain number of episodes

(그림 1) *Q*-학습 알고리즘

2.2 TD(0)-학습

TD(0)-학습은 Monte Carlo 방법과 동적 프로그래밍(Dynamic Programming)의 각 장점을 혼합한 형태의 강화학습 알고리즘이라 할 수 있다[12]. 즉, *TD*(0)-학습은 Monte Carlo 방법과 같이 동적 환경에 대한 모형(model)을 요구하지 않고 학습할 수 있으며, 동적 프로그래밍처럼 최종 결과를 기다리지 않고 다른 상태의 학습된 평가를 근거로 현재 상태를 갱신함으로서 학습할 수 있다.

TD(0)-학습은 현재 상태에 대한 예측과 다음 상태에 대한 예측과의 차이(difference)를 이용하여 현재 상태의 *Q*-함수 값을 식(3)과 같이 갱신한다.

$$Q(x_t, a_t) = (1-\alpha)Q(x_t, a_t) + \alpha \cdot TD\,error \quad (3)$$

여기서, α는 학습율(learning rate)이고, $TD\,error$는 식(4)와 같이 계산한다.

$$TDerror = r_t + \gamma[\max_{a \in A(x_{t+1})} Q(x_{t+1}, a_{t+1}) - Q(x_t, a_t)] \quad (4)$$

식(3)과 식(4)를 이용하는 *TD*(0)-학습 알고리즘을 표현하면 (그림 2)와 같다.

1. Initialize $Q(x_t, a)$ value functions arbitrarily
2. Initialize the environment : set a current state x_t
3. Select an action following a certain policy (e. g. Bolzmann probability distribution)
4. Take an action a
Observe reward r,
Find the next state x_{t+1},
and Select the next action a_{t+1}
5. Update the estimate values $Q(x_t, a)$ as follows :
5.1 $TDerror = r_t + \gamma[\max_{a \in A(x_{t+1})} Q(x_{t+1}, a_{t+1}) - Q(x_t, a_t)]$
5.2 $Q(x_t, a_t) = (1-\alpha)Q(x_t, a_t) + \alpha \cdot TD\,error$
6. Let $x_t = x_{t+1}$
7. Go to step 3 until the state x_t is a goal state
8. Repeat steps 2 to 7 for a certain number of episodes

(그림 2) *TD*(0)-학습 알고리즘

TD(0)-학습은 4-튜블(x_t, a_t, x_{t+1}, a_{t+1})을 이용하여 현재 상태의 $Q(x_t, a_t)$값에 대한 평가를 갱신하면서 학습한다.

2.3 *TD*(λ)-학습

강화학습에서 temporal-credit의 할당 문제 즉, 학습을 수행하는 에이전트가 어떤 행동을 선택하여 어떤 상태에 도달하였을 때 그 행동에 대해 어떻게 보상할 것인가에 대한 문제는 강화학습의 중요한 과제라 할 수 있다. 이를 해결하기 위한 방법으로 eligibility trace를 사용하는 방법이 제안되었다[13]. Eligibility trace란 어떤 행동을 선택하거나 어떤 상태를 방문하는 것과 같은 사건(event)의 발생에 대한 기록이라 할 수 있는데, *TD*(λ)-학습 알고리즘은 강화 기법으로 eligibility trace 값과 trace-decay factor $\lambda(0 < \lambda < 1)$를 사용하는 강화학습 알고리즘이다. Eligibility trace를 이용한 *TD*(λ)-학습은 현재 상태에 대한 평가를 식(5)와 같이 갱신한다.

$$Q(x_t, a_t) = (1-\alpha)Q(x_t, a_t) + \alpha \cdot \delta_t \cdot e(x_t, a_t) \quad (5)$$

식(5)에서 δ_t는 식(6)과 같이 계산되며, 모든 상태와 행동들에 대한 $e(x_t, a_t)$는 식(7)과 같이 계산된다.

$$\delta_t = r_t + \gamma[\max_{a \in A(x_{t+1})} Q(x_{t+1}, a_{t+1}) - Q(x_t, a_t)] \quad (6)$$

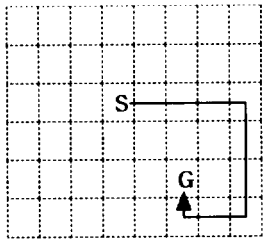
$$e(x_t, a_t) = \begin{cases} \gamma\lambda e(x_{t-1}, a_{t-1}) + 1 & \text{if } x_{t-1} = x_t \text{ and } a_{t-1} = a_t \\ \gamma\lambda e(x_{t-1}, a_{t-1}) & otherwise \end{cases} \quad (7)$$

Eligibility trace를 이용한 *TD*(λ)-학습은 상태 전이가 발생할 때마다 $e(x_t, a_t)$ 값이 누적되며, 알고리즘은 (그림 3)과 같이 표현된다.

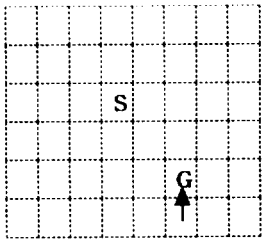
1. *Initialize* $Q(x_t, a)$ *value functions arbitrarily and* $e(x_t, a) = 0.0$ *for all* x, a
2. *Initialize the environment : set a current state* x_t
3. *Select an action following a certain policy (e. g. Bolzmann probability distribution)*
4. *Take an action* a_t, *Observe reward* r_t, *Find the next state* x_{t+1}, *and Select the next action* a_{t+1}
5. *Update the estimate values* $Q(x_t, a)$ *as follows :*
   5.1 $\delta_t = r_t + \gamma[\max_{a \in A(x_{t+1})} Q(x_{t+1}, a_{t+1}) - Q(x_t, a_t)]$
   5.2 *if* $(x_{t-1} = x_t$ *and* $a_{t-1} = a_t)$
   $e(x_t, a_t) = \gamma\lambda e(x_{t-1}, a_{t-1}) + 1$
   else (for all x, a*)*
   $e(x_t, a_t) = \gamma\lambda e(x_{t-1}, a_{t-1})$
6. $Q(x_t, a_t) = (1-\alpha)Q(x_t, a_t) + \alpha \cdot \delta_t \cdot e(x_t, a_t)$
7. *Let* $x_t = x_{t+1}$, $a_t = a_{t+1}$
8. *Go to step 3 until the state* x_t *is a goal state*
9. *Repeat steps 2 to 8 for a certain number of episodes*

(그림 3) $TD(\lambda)$-학습 알고리즘

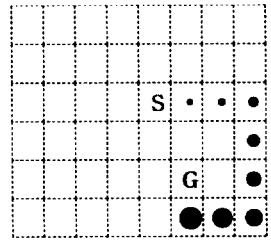
미로 환경에서 eligibility trace를 이용하면 학습 성능을 개선할 수 있으며, 그 이유는 (그림 4)와 같이 설명할 수 있다.



(a) 한 번의 episode 결과



(b) $TD(0)$-학습의 결과



(c) TD(λ)-학습의 결과

(그림 4) $TD(0)$-학습과 TD(λ)-학습과의 차이

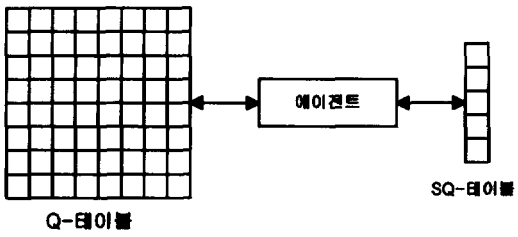
(그림 4)의 미로 환경에서, 에이전트가 미로 환경으로부터 받는 강화값은 에이전트의 다음 상태가 목표상태(G)인 경우 양의 값(positive value)을 그렇지 않은 경우 0이다. (그림 4) (a)는 미로 환경에서 시작 상태(S)에서 목표상태(G)까지 한 번의 에피소드에서 에이전트에 의해 탐색된 경로를 의미한다. (그림 4) (b)는 $TD(0)$-학습의 결과로서 목표상태를 찾기 위한 일련의 행동들 중 가장 큰 강화값을 받는 마지막 행동에 대해서만 강화되는 특징이 있다. 반면에 $TD(\lambda)$-학습은 (그림 4) (c)와 같이 시작 상태에서 목표상태를 찾기 위한 일련의 행동들이 강화된다. 그러므로 시작 상태에서 목표상태를 찾기 위해 여러 번의 에피소드를 수행할 경우 $TD(\lambda)$-학습이 $TD(0)$-학습보다 더 빠르게 최단 경로를 찾을 수 있다. 그러나 Q-학습 알고리즘의 단계 8, $TD(0)$-학습 알고리즘의 단계 8, 그리고 $TD(\lambda)$-학습 알고리즘의 단계 9에서 표현된 것처럼 몇 번의 에피소드가 발생한 후에 최단 경로를 찾을 수 있는지에 대해 명확하게 표현하지 못하는 단점이 있다.

3. GP-강화학습

본 논문에서 제안한 GP-강화학습 시스템은 (그림 5)와 같이 주어진 미로 환경에 대해 전역학습을 수행하기 위한 Q-테이블(table), 학습을 수행하는 에이전트, 지역학습을 수행하기 위한 SQ-테이블(Sub_Q-table)들로 구성되어 있다.

미로 환경에서 전역학습을 위한 Q-테이블은 $|O| \times |A|$의 크기를 갖는 행렬이며, $|O|$는 관찰 가능한 상태들의 수이고, $|A|$는 에이전트가 수행할 수 있는 행동들의 수이다. Q-테이블의 각 $Q_t(x_i, a_j)$는 시간 t에서

상태 x_i 에 있는 에이전트가 행동 a_j를 수행한 평가 값이다. 지역학습을 위한 SQ-테이블은 $|O|\times|A|$로 구성되어 있으며, $|O|$는 목표상태와 목표상태에 인접한 상태들의 수이고, $|A|$는 에이전트가 수행할 수 있는 행동들의 수이다.



(그림 5) GP-강화학습 시스템 구조

3.1 전역학습

전역학습은 에이전트가 시작 상태에서 특정 상태 값을 가지고 있는 목표상태를 탐색하기 위한 학습이다. 미로 환경에서 학습을 수행하는 에이전트가 최단 경로를 탐색하기 위한 전역학습은 Sutton이 제안한 식(8)과 같은 replacing eligibility traces를 이용하였다.

$$Q(x_t, a_t) = (1-\alpha)\,Q(x_t, a_t) + \alpha\delta e(x_t, a_t) \quad (8)$$

식(8)에서 α는 학습율(learning rate)이고, δ는 TD-오류를 의미한다. TD-오류는 식(4)에서와 같이 계산되고, $e(x_t, a_t)$는 식(9)와 같이 계산된다.

$$e(x_t, a_t) = \begin{cases} \gamma\lambda e(x_{t-1}, a_{t-1}) + 1, & \text{if } x_{t-1} = x_t \text{ and } a_{t-1} = a_t \\ 0, & \text{if } x_{t-1} = x_t \text{ and } a_{t-1} \neq a_t \\ \gamma\lambda e(x_{t-1}, a_{t-1}), & \text{if } x_{t-1} \neq x_t \end{cases} \quad (9)$$

식(9)에서 λ는 trace-decay factor이고, $\gamma\lambda$는 어떤 상태가 최근에 방문되었을 때를 의미한다. 그러므로 $e(x_t, a_t)$는 가장 최근에 방문한 상태 x_t에서 행동 a_t를 수행하는 것이 얼마나 적합한가(eligible)에 대한 정도를 나타낸다. 전역학습에서 현재 상태에서 선택 가능한 행동들의 집합 $A(x_t)$에 속해 있는 특정 행동 $(\bar{a})$을 선택하기 위한 방법은 식(10)과 같은 볼쯔만 확률 분포(Bolzmann probability distribution)를 이용하였다.

$$p(\bar{a}) = \frac{e^{\frac{Q(x_t, \bar{a})}{T}}}{\sum_{a \in A(x_t)} e^{\frac{Q(x_t, a)}{T}}} \quad (10)$$

식(10)에서 T는 임의성 정도(randomness)를 제어하는 온도(temperature) 상수이며, 전역학습을 위한 알고리즘은 (그림 6)과 같다.

1. Initialize $Q(x_t, a)$ value functions arbitrarily and $e(x_t, a) = 0.0$ for all x, a
2. Determine a current state(x_t),
goal state,
and goal state value(gv)
3. Select an action following a certain policy
(e. g. Bolzmann probability distribution)
4. Take an action a_t
Observe reward r_t,
Find the next state x_{t+1},
and Select the next action a_{t+1}
5. TD-error = $r_t + \gamma \cdot [\max_{a \in A(x_{t+1})} Q(x_{t+1}, a_{t+1}) - Q(x_t, a_t)]$
6. Update the eligibility value $e(x_t, a_t)$
7. Update the estimate value of $Q(x_t, a_t)$ value function
8. if (state value of x_{t+1} = gv)
local_learn(x_{t+1})
else $x_t = x_{t+1}$
9. Go to step 3 until (start state value = gv)

(그림 6) 전역학습 알고리즘

전역학습의 단계 9는 2장에서 제시된 강화학습의 문제점 즉, 언제까지 학습을 수행해야 최적 해를 찾을 수 있는가를 보완하기 위한 단계이다. 장애물 상태의 상태 값은 1, 장애물이 아닌 상태의 상태 값은 0, 그리고 목표상태의 상태 값은 0과 1이 아닌 값을 갖는 미로 환경에서 시작 상태의 값이 목표상태의 값이 같을 때까지만 학습을 수행한다는 의미이다.

3.2 지역학습

지역학습은 전역학습 과정에서 탐색된 목표상태의 상태 값을 목표상태와 인접한 상태들로 전파시키고 나서, 인접 상태에서 목표상태를 탐색하기 위한 학습이다. 지역학습은 에이전트가 SQ-테이블과 상호 작용하면서 학습을 수행한다. SQ-테이블은 목표상태와 목표상태에

인접한 상태들에 대한 정보를 가지고 있는 테이블이다. 목표상태에 인접한 상태란 목표상태에서 해밍 거리(Hamming distance)가 1인 상태들을 의미한다. 전역학습이 수행되고 나서 에이전트가 찾은 목표상태의 위치와 상태 값은 SQ-테이블의 첫 번째 요소에 저장되고 목표상태에 인접한 상태들은 SQ-테이블의 나머지 요소에 저장된다. 그리고 나서 지역학습은 식(11)을 이용하여 SQ-테이블에 저장된 인접 상태에서 SQ-테이블의 목표상태 값을 가진 상태를 찾기 위해 학습을 수행한다.

$$Q'(x_t, a_t) = \max[r_t + Q'(x_t, a)] \qquad (11)$$

식(11)은 현재 상태에서 가장 큰 강화값을 주는 다음 상태를 탐색한다는 의미이다. 식(11)에서 discount factor(γ)를 삭제한 이유는 지역학습은 목표상태와 인접한 상태에서만 학습을 수행하기 때문에 많은 상태 전이를 발생하지 않기 때문이다. 식(11)을 이용하여 지역학습을 수행하는 지역학습 알고리즘은 (그림 7)과 같다.

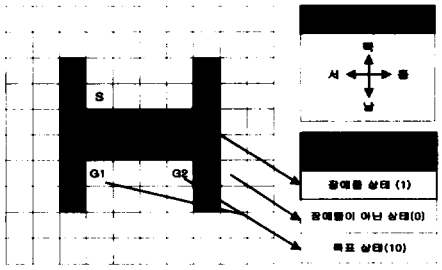
```
local_learn(x_t)
{
  1. Initialize Q'(x_t, a) of coordinates in SQ-table
  2. Each adjacent coordinate in SQ-table
     2.1 Take action a_t
         Observe reward r_t ,
         Find the next state x_{t+1}
  3. if ( x_{t+1} = gv) {
       Q'(x_t, a_t) = max[r_t + Q'(x_t, a)]
       x_t state value = gv
     }
     else Q'(x_t, a_t) = 0.0
}
```

(그림 7) 지역학습 알고리즘

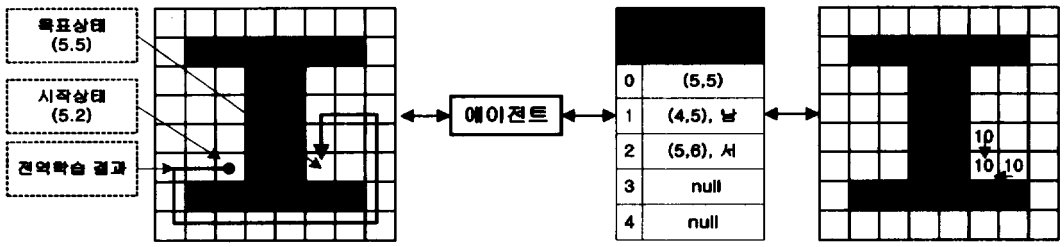
예로서 장애물 상태는 1, 장애물이 아닌 상태는 0, 그리고 목표상태는 10으로 표현된 미로 상태가 주어졌을 때, 전역학습을 수행하여 에이전트가 탐색한 경로가 (그림 8) (a)와 같은 경우 지역학습을 위한 SQ -테이블을 (그림 8) (b)와 같다. 지역학습을 수행하고 나서 목표상태 값이 인접 상태들의 상태 값으로 전파된 상태는 (그림 8) (c)와 같다. 지역학습은 (그림 8) (c)와 같이 목표상태와 인접 상태에서 목표상태를 찾기 위해 학습한다. 지역학습이 끝나고 나면, 전역학습은 목표상태 값과 동일한 상태 값을 갖는 상태를 찾기 위해 다시 학습을 수행한다.

4. 실험 및 분석

Q-학습, *TD*(0)-학습, *TD*(λ)-학습 등과 본 논문에서 제안된 GP-강화학습과의 효율성 분석을 위해 최적 해에 탐색하기 위해 얼마나 많은 상태 전이를 하였는가와 학습율에 따라 수렴 속도가 어떻게 변하는가를 각각 비교 분석하였다. 이를 위해 (그림 9)와 같은 미로 환경을 설정하였으며, (그림 9)에서 S는 시작상태(3, 3), G1(6, 3)과 G2(6, 6)는 서로 다른 두 개의 목표상태이다.



(그림 9) 미로 환경

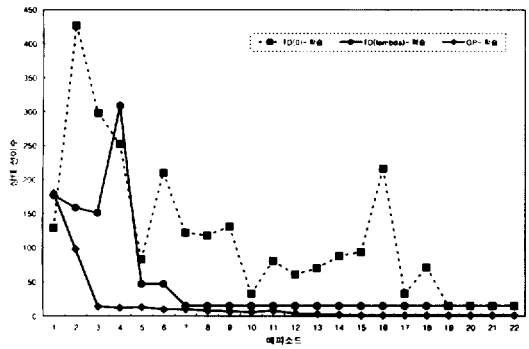


(8-a) 전역 학습 결과 (8-b) SQ-테이블 (8-c) 지역 학습 결과

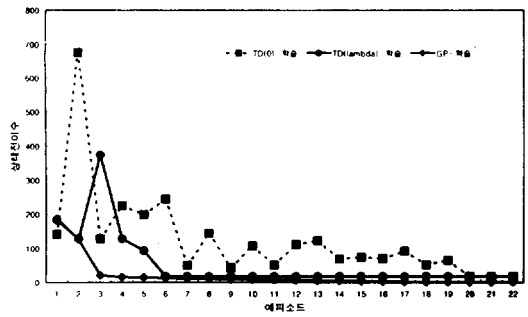
(그림 8) 전역학습과 지역학습 결과

4.1 상태전이 수에 의한 성능 분석

GP-강화학습의 성능을 평가하기 위해 시작 상태에서 목표상태 G1까지의 최단 경로를 탐색하기 위한 실험과 시작 상태에서 또 다른 목표상태 G2까지의 최단 경로를 탐색하기 위한 실험을 각각 수행하였다. 각 학습 알고리즘의 학습율(α)을 0.7, discount factor(γ)를 0.9로 고정하고 나서 시작 상태(3, 3)에서 G1(6, 3)까지의 최단 경로를 찾기 위한 학습 결과는 (그림 10)과 같고, 시작 상태에서 G2(6, 6)까지의 최단 경로를 찾기 위한 학습 결과는 (그림 11)과 같다.



(그림 10) G1 상태에 대한 학습 결과



(그림 11) G2 상태에 대한 학습 결과

(그림 10)과 (그림 11)에서 x축은 에피소드, y축은 각 에피소드에서 발생한 상태전이 수를 의미한다. GP-강화학습은 시작 상태에서 목표상태 G1까지의 최단 경로를 찾기 위해 15번의 에피소드가 발생하였고, 총 376번의 상태 전이를 하였다. 다른 목표상태 G2까지의 최단 경로를 찾기 위해 18번의 에피소드가 발생하였고, 총 456번의 상태 전이를 하였다. 반면에 Q-학습, $TD(0)$-학습, $TD(\lambda)$-학습에서 목표상태 G1과 G2를 찾기 위해 수행한 총 상태 전이 수는 <표 1>과 같다.

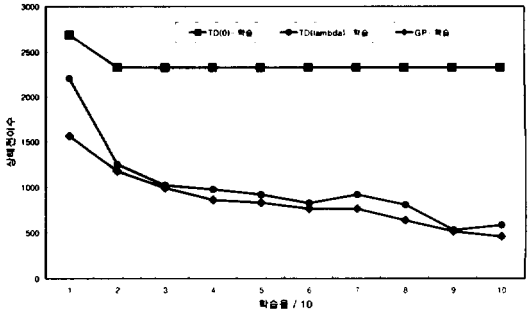
〈표 1〉 상태 전이 수에 의한 성능 분석

평가 기준 / 학습 방법	목표상태(G1) 총 상태전이 수	목표상태(G2) 총 상태전이 수
Q-학습	9977	12028
$TD(0)$-학습	2158	2687
$TD(\lambda)$-학습	850	917
GP-강화학습	376	456

<표 1>에 나타나는 바와 같이 Q-학습은 시작 상태에서 G1까지의 최단 경로를 찾기 위해 총 9977번의 상태 전이가 발생하였고, G2까지의 최단 경로를 찾기 위해 총 12028번의 상태 전이가 발생하였다. $TD(0)$-학습은 시작 상태에서 G1까지의 최단 경로를 찾기 위해 총 2158번의 상태 전이가 발생하였고, G2까지의 최단 경로를 찾기 위해 총 2687번의 상태 전이가 발생하였다. $TD(\lambda)$-학습은 시작 상태에서 G1까지의 최단 경로를 찾기 위해 총 850번의 상태 전이가 발생하였고, 목표상태 G2까지의 최단 경로를 찾기 위해 총 917번의 상태 전이가 발생하였다.

4.2 학습율에 따른 성능 분석

본 논문에서 제안한 GP-강화학습 방법과 비교 분석을 위해 사용된 $TD(0)$-학습과 $TD(\lambda)$-학습에서 목표상태 G2까지의 최단 경로를 찾기 위해 사용된 학습율(α)에 따른 상태 전이 수는 (그림 12)와 같다.



(그림 12) 학습율(α)에 따른 상태 전이 수

(그림 12)에서처럼 G2를 찾기 위해 $TD(\lambda)$-학습은 학습율이 0.9일 때 가장 빠르게 수렴하였고, $TD(0)$-학습은 학습율에 큰 영향을 받지 않음을 알 수 있었다.

그리고 본 논문에서 제안한 GP-강화학습은 학습율이 0.99일 때 가장 빠르게 수렴한다는 것을 알았다. 미로 환경에서 각 학습 알고리즘에서 사용한 최적의 매개변수 값과 상태 전이 수는 <표 2>와 같다.

〈표 2〉 최적 매개변수 값

매개변수 / 학습방법	α	γ	상태전이 수
$TD(0)$-학습	*0.7*	*0.99*	*2327*
$TD(\lambda)$-학습	*0.9*	*0.99*	*527*
GP-학습	***0.99***	***0.99***	***456***

5. 결론 및 향후 연구 방향

본 논문에서 제안한 GP-강화학습은 학습을 언제까지 수행하여야 최적의 정책을 발견하는가에 대한 문제점을 보완하였으며, 실험 결과 GP-강화학습은 Q-학습, $TD(0)$-학습, $TD(\lambda)$-학습과 비교한 결과 최단 경로에 매우 빠르게 수렴함을 알 수 있었다. 그러나 제안된 방법은 단지 미로 환경에 대한 학습 방법이므로 다른 환경에 그대로 적용하기에는 무리가 따른다는 단점이 있다. 그러므로 다른 환경에서도 무리 없이 적용할 수 있는 강화학습 방법에 관한 연구가 필요하다. 또한 학습을 수행하는 에이전트가 선택한 행동이 얼마나 적합한지에 대한 예측이 학습에 많은 영향을 준다는 것을 알 수 있었다. 따라서 주어진 환경에 대한 지식을 학습 과정에서 적절히 이용하는 방법을 모색해 보는 것이 바람직하다고 생각된다. 주어진 환경에 대한 모형(model)을 이용한 강화학습 방법들이 일부 제시되고는 있지만 실세계에 대한 정확한 모형을 얻기가 대단히 어려우므로 보다 효율적인 학습을 위해서는 계획(planning)과 학습을 유기적으로 통합한 형태의 강화학습 알고리즘 개발에 대한 연구가 필요하다.

참 고 문 헌

[1] L. P. Kaelbling, M. L. Littman and A. W. Moore, "Reinforcement Learning : A Survey," Journal of Artificial Intelligence Research, 4, pp.237-285, 1996.

[2] C. W. Anderson, "Learning to control an inverted pendulum using neural network," IEEE Control Systems Magazine, 9, pp.31-37, 1989.

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

[4] 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.

[5] S. P. Singh, "Transfer of Learning by Composing Solutions of Elemental Sequential Tasks," Machine Learning, 8, pp.323-339, 1992.

[6] C. J. Watkins and P. Dayan, "Technical note : Q-learning," Machine Learning, 8, pp.279-292, 1992.

[7] P. Cichosz, "Reinforcement Learning Algorithms Based on the Method of Temporal Difference," MS thesis, University of Warsaw, Computer Science, 1996.

[8] R. S. Sutton, A. G. Barto, "Reinforcement Learning : An Introduction," MIT Press, 1998.

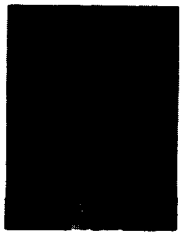
[9] P. Dayan, "The convergence of TD(λ) for general λ," Machine Learning, 8(3), pp.341-362, 1992.

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

[11] M. L. Puterman, "Markov Decision Processes-Discrete Stochastic Dynamic Programming," John Wiley & Sons, New York, 1994.

[12] R. S. Sutton, "TD models : Modeling the world at a mixture of time scales," Proceedings of the Twelfth International Conference on Machine Learning, pp.531-539, 1995.

[13] R. S. Sutton, "Generalization in Reinforcement Learning : Successful examples using sparse coarse coding," Advances in Neural Information Processing Systems, 8, pp.1038-1045, MIT Press, Cambridge MA, 1996.



김 병 천

e-mail : bckim@ce.ansung.ac.kr

1988년 한남대학교 전산학과(학사)

1990년 숭실대학교 대학원 전자계산학과(석사)

1997년 명지대학교 대학원 컴퓨터공학과 박사과정 수료

1991년~1993년 안성농업전문대학교 전자계산학과 전임강사

1993년~현재 한경대학교 컴퓨터공학과 조교수

관심분야 : Machine Learning, Artificial Neural Network, Knowledge-Based System

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