논문 2010-5-38

목표지향적 강화학습 시스템

Goal-Directed Reinforcement Learning System

이창훈*

Chang-Hoon Lee

요 약 강화학습(reinforcement learning)은 동적 환경과 시행-착오를 통해 상호 작용하면서 학습을 수행한다. 그러므로 동적 환경에서 TD-학습과 $TD(\lambda)$-학습과 같은 강화학습 방법들은 전통적인 통계적 학습 방법보다 더 빠르게 학습을 할 수 있다. 그러나 제안된 대부분의 강화학습 알고리즘들은 학습을 수행하는 에이전트(agent)가 목표 상태에 도달하였을 때만 강화 값(reinforcement value)이 주어지기 때무에 최적 해에 매우 늦게 수렴한다.
본 논문에서는 미로 환경(maze environment)에서 최단 경로를 빠르게 찾을 수 있는 강화학습 방법(GORLS : Goal-Directed Reinforcement Learning System)을 제안하였다. GDRLS 미로 환경에서 최단 경로가 될 수 있는 후보 상태들을 선택한다. 그리고 나서 최단 경로를 탐색하기 위해 후보 상태들을 학습한다. 실험을 통해, GDRLS는 미로 환경에서 TD-학습과 $TD(\lambda)$-학습보다 더 빠르게 최단 경로를 탐색할 수 있음을 알 수 있다.

Abstract Reinforcement learning performs learning through interacting with trial-and-error in dynamic environment. Therefore, in dynamic environment, reinforcement learning method like TD-learning and $TD(\lambda)$-learning are faster in learning than the conventional stochastic learning method. However, because many of the proposed reinforcement learning algorithms are given the reinforcement value only when the learning agent has reached its goal state, most of the reinforcement algorithms converge to the optimal solution too slowly.
In this paper, we present GDRLS algorithm for finding the shortest path faster in a maze environment. GDRLS is select the candidate states that can guide the shortest path in maze environment , and learn only the candidate states to find the shortest path. Through experiments, we can see that GDRLS can search the shortest path faster than TD-learning and $TD(\lambda)$-learning in maze environment

Key Words : Reinforcement learning, TD-learning, $TD(\lambda)$-learning, Shortest path,

Ⅰ. 서 론

M. L. Minsky에 의해 소개된 강화 학습(reinforcement learning)은 동적프로그래밍(dynamic programming)과 교사학습(supervised learning)을 혼합한 형태의 학습 방법으로서 학습을 수행하는 에이전트(agent)는 에이전트 외부에 존재하는 환경(environment)과 시행-착오(trial-and-error)를 통해 상호작용(interaction)하면서 학습한다[1]. 일반적으로 강화 학습을 위해 제시된 많은 학습 방법들은 동적프로그래밍[2], 몬테카를로(Monte Carlo)[3] 그리고 TD-학습[4] 등으로 구분할 수 있다.

동적프로그래밍을 이용한 학습 방법은 최적화(optimization)와 제어(control)에 관한 문제를 해결하기 위해 널리 이용된 학습방법이다. 동적프로그래밍을 이용한 학습 방법들은 가능한 모든 상태전이들에 대한 확률분포(probability distribution)가 필요하므로 주어진 환경의 복잡성과 크기에 많은 제약을 받는다.

몬테카를로 학습 방법은 특정 상태로 상태전이를 하

*정회원, 한경대학교 컴퓨터공학과(교신저자)
접수일자 2010.10.8 수정일자 2010.10.15
게재확정일자 2010.10.15

기 위한 확률만 필요로 하지만, 특정 에피소드(episode)가 끝난 후 현재 상태의 상태 값이 갱신되기 때문에 실시간 강화학습에 적합하지 않다.

TD-오류(Temporal Difference error)를 이용한 강화학습은 동적프로그래밍과 몬테카를로 학습 방법을 혼합한 형태의 학습 방법으로 현재 상태의 상태 값은 다음 상태로 상태전이 하기 전에 갱신되므로 점진적 학습 방법에 적합하다. 그러나 최적 함수에 매우 느리게 수렴하며, 언제까지 학습을 수행해야 최적 값-함수를 구할 수 있는가에 대해 명확하지 않다.

본 논문에서는 미로 환경(maze environment)과 같은 에피소딕 환경에서 최적 함수에 빠르게 수렴할 수 있는 GDRLS 방법을 제안하였다. 일반적으로 미로 환경에서 학습을 수행하기 위한 강화 학습 방법들은 최적 함수에 얼마나 빠르게 수렴하는가? 즉. 시작 상태(start state)에서 목표 상태(goal state)까지의 최단 경로를 얼마나 빠르고 정확하게 찾을 수 있는가를 기준으로 한다.

본 논문에서 제안된 GDRLS는 탐색 모듈(search module)과 학습 모듈(learning module)로 구성되어 있다. 탐색 모듈은 최적 경로를 찾기 위해 최적 경로가 될 수 있는 후보 상태들을 탐색하기 위해 학습을 수행하는 단계이고, 학습 모듈은 탐색 모듈에서 선택된 상태들에 대해 최단 경로를 탐색하기 위한 학습하는 단계이다. 제안된 학습 방법은 TD-오류를 이용한 Q-학습[5] 그리고 TD(λ)를 이용한 $Q(\lambda)$-학습[6]을 Abbott, R이 제안한 미로 환경[7]에 적용한 결과 제안된 방법이 최적 함수에 매우 빠르게 수렴함을 알 수 있었다.

본 논문은 5장으로 구성되어 있으며 각 장의 주요 내용은 다음과 같다. 2장에서는 Q-학습과 $Q(\lambda)$-학습의 학습 방법에 대하여 기술하고, 3장에서는 본 논문에서 제안한 학습방법과 특징에 대하여 설명한다. 4장에서는 본 논문에서 제안한 학습 방법과 Q-학습 그리고 $Q(\lambda)$-학습을 미로 환경에서 비교 분석한다. 그리고 5장에서는 결론과 함께 향후 연구 과제를 제시한다.

II. 관련연구

강화학습에서 Temporal-credit 할당 문제 즉, 학습을 수행하는 에이전트가 현재 상태에서 어떤 행동을 선택하여 상태전이를 하였을 때 에이전트가 선택한 행동에 대해 어떻게 보상(reward)할 것인가는 강화 학습에서 가장 중요한 과제라 할 수 있다. 이를 위해 TD-오류(temporal difference error)를 이용한 $TD(0)$-학습과 $TD(\lambda)$-학습 방법이 널리 이용되고 있다.

1. $TD(0)$-학습

TD-오류를 이용한 $TD(0)$-학습은 현재 상태의 출력에 대한 예측과 다음 상태의 출력에 대한 예측과의 차를 이용하여 현재 상태의 값-함수(value-function)에 대한 평가 값을 갱신함으로서 Temporal-credit 할당 문제를 해결한다. TD-오류를 이용한 대표적인 $TD(0)$-학습은 Q-학습과 $SARA$-학습 등이 있다.

Q-학습은 강화 학습을 위해 가장 널리 이용되는 학습 방법으로서 통계적 동적프로그래밍에 근거한 학습 방법이다. Q-학습은 학습을 수행하는 에이전트가 현재 상태(s_t)에서 어떤 행동(a_t)을 선택했을 때 받는 강화 값을 (상태-행동) 쌍에 대한 $Q(s_t,\ a_t)$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 Q-함수 값을 식(2.1)과 같이 갱신한다.

$$Q(s_{t,}a_t)=(1-\alpha)Q(s_t,a_t)+\alpha\{r_{t+1}+\gamma\max Q(s_{t+1,}a_{t+1})\} \quad (1)$$

식1을 이용한 Q-학습을 1-단계 Q-학습이라 하며, Q-함수가 수렴하면 최적의 정책은 각 상태에서 가장 큰 Q-값을 갖는 행동을 선택함으로서 구현된다. 1단계 Q-학습은 동적 프로그래밍을 이용한 Value-Iteration 학습 방법이나, Policy-Iteration 학습 방법보다 최적 값-함수에 빠르게 수렴한다. 그러나 최적 값-함수를 구하기 위해 많은 상태전이가 발생한다. 이를 위해 TD-오류를 이용한 Q-학습을 제안하였다. TD-오류를 이용한 Q-학습은 현재 상태의 (상태-행동) 쌍에 대한 Q-함수 값을 갱신하기 위해 식.2와 같이 TD-오류를 이용한다.

$$\delta_t=\ r_{t+1}+\ \gamma\{\max Q(s_{t+1,}a_{t+1})-Q(s_t,a_t)\} \quad (2)$$

TD-오류를 이용한 Q-학습은 최적 값-함수를 구하기 위해 현재 상태의 (상태-행동) 쌍에 대한 Q-함수 값을 식3과 같이 갱신한다.

$$Q(s_t,a_t) = (1-\alpha)Q(s_t,a_t) + \alpha \bullet \delta_t \quad (3)$$

현재 상태에서 Q-함수에 의해 계산된 Q-값들 중에서 가장 적당한 행동을 선택하는 방법은 일반적으로 식4와 같이 볼쯔만(Boltzmann) 확률 분포를 이용한다.

$$p(\bar{a} \mid s) = \frac{e^{\frac{Q(s_t,\bar{a})}{T}}}{\sum_{a \in A(s_t)} e^{\frac{Q(s_t,a)}{T}}} \quad (4)$$

식4에서 T는 행동 선택의 임의성(randomness) 정도를 제어하는 온도변수(temperature variable)이다. 볼쯔만 확률 분포에 의한 행동 선택은 현재 상태 s_t에서 선택 가능한 모든 행동 $\bar{a}$ 에 대한 확률 값 $p(\bar{a}|s_t)$를 계산하고 그 값 중에서 가장 큰 값과 (0,1) 사이의 난수(random number)를 비교하여 행동을 선택한다. 이와 같은 방법은 각 상태에서 선택 가능한 모든 행동들에 대해 확률 값을 부여한다는 장점이 있다.

2. $TD(\lambda)$-학습

$TD(\lambda)$-학습은 $TD(0)$-학습의 학습 성능을 개선하기 위해 현재 상태에서 선택된 (상태-행동) 쌍에 대한 eligibility trace를 이용한 강화 학습이다. 즉 현재 상태에서 선택된 (상태-행동) 쌍이 얼마나 바람직한가(eligible)를 의미한다. $TD(\lambda)$를 이용한 가장 대표적인 학습 방법은 $Q(\lambda)$-학습과 $SARA(\lambda)$-학습[8] 등이 있다.

$SARA(\lambda)$-학습은 TD-오류에 대한 예측(prediction)을 각 상태의 (상태-행동)쌍에 적용하여 식5와 같이 갱신한다.

$$Q(s_t,a_t) = Q(s_t,a_t) + \alpha\delta_t e(s_t,a_t) \quad (5)$$

식5에서 α는 학습률, δ_t는 TD-오류로서 식.6과 같다. 그리고 $e(s_t,a_t)$는 현재 상태에서 선택한 (상태-행동) 쌍에 대한 적합도(eligibility factor)로서 식7과 같이 계산된다.

$$\delta_t = r_{t+1} + \gamma\{Q(s_{t+1},a_{t+1}) - Q(s_t,a_t)\} \quad (6)$$

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

식7은 현재 상태에서 선택한 (상태-행동) 쌍이 이전에 선택한 (상태-행동) 쌍과 동일한 경우 (상태-행동) 쌍에 대한 적합도 1만큼씩 증가하며, 새로운 (상태-행동) 쌍을 선택할 경우 $\gamma\lambda$ 만큼씩 감소됨을 의미한다.

Ⅲ. 목표 지향적 강화 학습

본 논문에서는 미로 환경에서 최적 값-함수에 매우 빠르게 수렴할 수 있는 목표 지향적(goal-directed) 강화 학습(GDRLS) 방법을 제안한다. 제안된 학습 시스템은 초기 상태에서 목표 상태까지 최단 경로를 빠르게 탐색할 수 있으며 그림1과 같이 구성되어 있다.

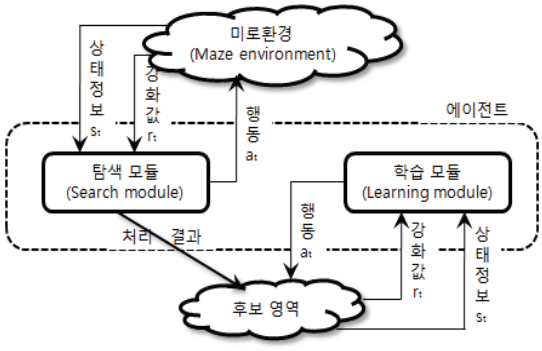


그림1. 목표지향적 강화 학습 시스템
Fig. 1 Goal-Directed Reinforcement Learning System

본 논문에 제안한 GDRLS은 그림1과 같이 최단 경로에 대한 후보 영역들을 탐색하는 탐색 모듈과 후보 영역에서 최단 경로를 탐색하는 학습 모듈로 구성되어있다.

1. 탐색 모듈

탐색 모듈은 미로 환경에서 최단 경로가 될 수 있는 후보 영역들을 선택하기 위해 학습하는 모듈이다. 탐색 모듈은 현재 상태에서 선택 가능한 모든 (상태-행동) 쌍들에 대한 *TD*-오류를 이용하여 모든 (상태-행동) 쌍들에 대한 평가 값을 식8과 같이 갱신한다. 그리고 현재 상태에서 최적의 (상태-행동) 쌍을 선택한다.

$$Q(s_t,a_t) = (1-\alpha)\,Q(s_t,a_t) + \alpha\delta_t e(s_t,a_t) \tag{8}$$

식(8)에서 δ_t는 *TD*-오류로서 식9와 같이 계산된다.

$$\delta_t = r_{t+1} + \gamma\{\max_{a\in A(s_t)} Q(s_{t+1},a_{t+1}) - Q(s_t,a_t)\} \tag{9}$$

식9에서 $e(s_t,a_t)$는 현재 상태에서 선택한 (상태-행동) 쌍이 얼마나 좋은가를 나타내는 적합도로서 식10과 같다.

$$e(s_t,a_t) = \begin{cases} 1\ , & \text{if } s=s_t \text{ and } a=a_t \\ \gamma\lambda e(s_{t-1},a_{t-1}, & otherwise \end{cases} \tag{10}$$

식10은 에이전트가 탐색 과정에서 선택한 (상태-행동) 쌍이 이미 선택되었던 (상태-행동) 쌍일 경우 적합도를 1로 하고, 그렇지 않은 겨우 적합도는 $\gamma\lambda$ 만큼씩 감소된다. 이는 임의의 상태가 다시 방문되었을 때 적합도가 다시 1의 값을 갖기 때문에 매우 긴 에피소딕 환경에서도 적합하다. 식(3.3)을 이용하여 현재 상태에서 선택 가능한 모든 (상태-행동) 쌍에 대한 값을 갱신하고 나서, 식11의 Gibbs 확률 분포를 이용하여 현재 상태에서 가장 적합한 (상태-행동) 쌍을 선택한다.

$$p(a \mid x) = \frac{e^{k\,Q(s_t,a_t)}}{\sum_{a'\in A(s_t)} e^{kQ(s_t,a')}} \tag{11}$$

식11에서, $k(0<k<1)$는 이미 학습된 Q-값을 어느 정도의 가중치를 가지고 참조할 것인가를 결정하는 상수이다. k값이 작을수록 현재의 Q 값을 무시하고 새로운 (상태-행동) 쌍을 선택할 확률이 높아진다. 그러므로 초기 학습 단계에서는 k 값을 낮게 설정하여 에이전트가 새로운 상태를 탐색할 수 있게 하고, 학습을 반복할수록 k값을 증가시켜 큰 값을 갖는 (상태-행동) 쌍을 선택하게 한다. 학습 모듈의 학습 알고리즘은 그림2와 같다.

```
Search_Module()
{
   Initialize Q(s,a) and e(s,a);
   Initialize s_t;
   Choose a_t from s_t using policy derived from
                                        Q(s_t,a_t);
   Repeat {
     Update_Qvalue(s_t);
     Observe s_t,   r_{t+1};
     Choose a_{t+1} from s_{t+1};
     Update_Evalue(s_t, a_t, r_{t+1}, s_{t+1});
     s_t = s_{t+1};   a_t = a_{t+1};
   } Until(s_t == goal state value)
}
```

그림 2. 탐색모듈의 학습 알고리즘
Fig. 2 Learning algorithm of Search module

그림.2의 탐색 모듈에서 ***Update_Qvalue***(s_t) 함수는 학습을 수행하는 에이전트가 현재 상태에서 선택 가능한 모든 (상태-행동) 쌍들에 대한 상태 값을 갱신하는 함수로서 그림 3과 같다.

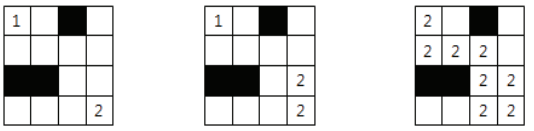
```
Update_Qvalue(s_t) {
   For all s_t, a (a∈A(s_t)) {
     if (observable s_{t+1}) {
        r_{t+1} = Reward;
        Q(s_t,a_t) = (1-α) Q(s_t,a_t) + αδ_t e(s_t,a_t);
        if(s_{t+1} == goal);
           s_t 상태값 = goal 상태값;
     }
     else
     r_{t+1} = Penalty;
}
```

그림 3. Update_Qvalue(s_t) 함수
Fig. 3. Update_Qvalue(s_t) Function

그림 2의 *Update_Evalue*$(s_t, a_t, r_{t+1}, s_{t+1})$ 함수는 식10을 이용하여 현재 상태에서 선택한 (상태-행동) 쌍에 대한 적합도를 갱신하는 함수이다.

2. 학습 모듈

학습 모듈은 탐색모듈이 탐색한 최단 경로 후보 영역들에 대해서 초기 상태에서 목표 상태까지 최단 경로를 탐색하기 위해 학습을 수행하는 모듈이다. 예를 들어 (그림4의 (a)와 같은 초기 상태의 상태 값을 1이라 하고, 목표 상태의 상태 값이 2인 4×4인 미로 환경에서, 선택 모듈의 1차 에피소딕 학습 결과는 (그림 3.4-b)와 같고, 최종 학습 결과는 (그림3.4-c)와 같다.



(a) 미로환경 (b) 1차 에피소딕 (c) 최종 탐색

그림 4. 탐색 모듈의 처리 절차
Fig. 4. Processing result of search module

학습 모듈은 탐색 모듈에 의해 탐색된 그림 4의 c와 같은 후보 영역에 대해 최단 경로를 탐색하기 위해 상태 값이 2인 상태들만 학습을 수행하며, 학습 알고리즘은 (그림 5와 같다.

```
Learning_Module()
{
   Initialize Q(s,a) and e(s,a);
                  // for all s,   a
   Repeat {
     Select s_t as a start state;
     Choose a_t from s_t using policy
           derived from 식11;
     Repeat {
       Update_Qvalue(s_t);
       Take an action a_t  using
                  policy derived from 식11;
       Observe s_t,   r_{t+1};
       Choose a_{t+1} from s_{t+1};
       Update_Evalue(s_t, a_t, r_{t+1}, s_{t+1});
       s_t = s_{t+1};   a_t = a_{t+1};
     } Until(( s_t == 2) && ( s_t != goal state ))
   } Until( a certain number of episodes)
}
```

그림 5. 학습모듈의 학습 알고리즘
Fig. 5. Learning algorithm of learning module

Ⅳ. 실험 및 결과

본 논문에서 제안한 GDRLS의 학습 성능을 비교 평가하기 위해 Rohit Kelkar가 제안한 그림 6과 같은 8×8 미로 환경을 이용하였다. 그림8과 같은 미로 환경에서 1의 초기 상태, 2는 목표 상태 그리고 음영부분은 상태값 3을 갖는 장애물이다.

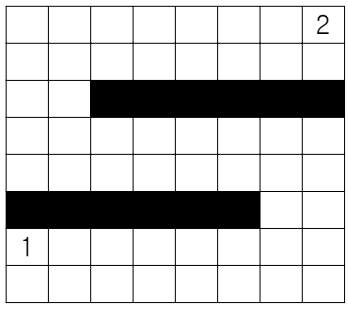


그림 6. 미로 환경
Fig. 6. Maze Environment

GDRLS의 학습 성능을 평가하기 위해 $TD(\lambda)$를 이용한 $Q(\lambda)$-학습과 비교하였다. 학습을 위해 $\alpha = 0.8$, $\gamma = 0.9$ 그리고 $\lambda = 0.9$로 하였고, 각 학습 방법들이 최단 경로에 수렴할 때 까지 각 시도(trial)에 대한 상태 전이 수를 비교하였다. $Q(\lambda)$-학습이 최단 경로를 탐색하기위해 그림 7과같이 270번의 시도(trial) 만에 성공하였고, 총 19460번의 상태 전이를 하였다.

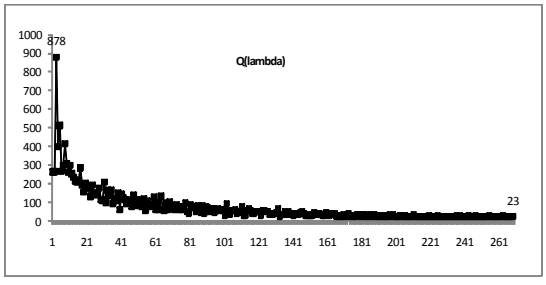


그림 7. $Q(\lambda)$-학습의 학습 결과
Fig. 7. $Q(\lambda)$-learning result

본 논문에서 제안한 GDRLS의 탐색 모듈은 그림4와 같은 최단 경로 후보 영역들의 탐색하기 위해 34번의 시도를 하였고, 총 5,436번의 상태전이를 하였다. 탐색모듈의 처리 결과는 그림 8과 같다.

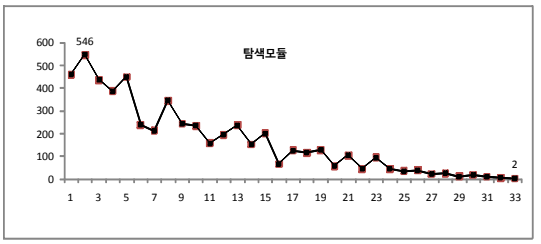


그림 8. 탐색 모듈의 결과
Fig. 8. Result of Search module

GDRLS의 탐색 모듈을 수행하고 나서 학습 모듈을 수행한 결과는 그림9와 같다.

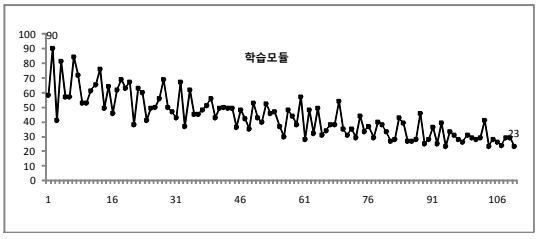


그림 9. 학습 모듈 결과
Fig. 9. Result of Learning module

학습 모듈은 111번의 시도 끝에 E E E E E E N N W W W W W N N N E E E E E N E 의 경로를 찾아냈으며, 총 상태 전이수는 4,816번 이루어졌다.

V. 결 론

본 논문에서 제안한 GDRLS의 학습은 4장에서 언급한 바와 같이 탐색 모듈이 34번의 시도를 하였고, 5,436번의 상태 전이를 하였다. 그리고 나서 학습 모듈이 111번의 시도를 하였고 4,816번의 상태 전이를 하였다. 그러므로 주어진 환경에서 최단 경로를 탐색하기 위해 총 10,252번의 상태 전이를 하여 $Q(\lambda)$-학습보다 약 50%의 성능이 개선됨을 알 수 있었다. 이는 주어진 환경에서의 선행 학습이 매우 중요함을 알 수 있었다.

$Q(\lambda)$-학습과 본 논문에서 주어진 GDRELS 학습 방법은 언제 까지 학습을 수행하는 것이 최적인가에 대한 명확한 기준이 없는 단점이 있다. 그러므로 학습을 언제까지 수행하여야 되는가에 대한 연구가 필요하다. 또한 본 논문에서는 하나의 목표를 찾는 문제를 다루었으나 다중의 목표를 추적할 수 있는 학습 방법이 필요하다.

참 고 문 헌

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

[2] D. P. Bertsekas, "Dynamic Programming and Optimal Control", Athena Scientific, Belmont, MA., 1995.

[3] M.H.Kalos and P. A. Whitlock, " Monte Carlo Methods", Wiley, NY., 1986.

[4] P. Dayan and G. E. Hinton, "Improving generalization for temporal difference learning : the successor representation", Neural Computation, 5, pp.613-624, 1993.

[5] C. J. C. H. Watkins, "Learning from Delayed Rewards, Ph.D. Thesis, King's College, Cambride, U.K., 1989.

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

[7] Abbott, R: Mad Mazes: Intriguing Mind Twisters for Puzzle Buffs, Game Nuts and Other Smart People. Adams Media, 1990

[8] S.P. Singh and R. S. Sutton, "Reinforcement learning 조소 Replacing Eigibility Traces", Machine Learning, 22, pp. 123-158, 1996.

저자 소개

이 창 훈(정회원)

- 1998년 중앙대학교 대학원 컴퓨터공학과(공학,박사)
- 2002년 ~ 현재 한경대학교 컴퓨터공학과 교수

<주관심분야 : 객체지향, 정형화방법, 컴포넌트, 영상처리 등>