# 복수의 부분 작업을 처리할 수 있는 확장된 Q-Learning†

## Extended Q-Learning under Multiple Subtasks

오 도 훈* 이 현 숙** 오 경 환***
(Do-Hun Oh) (Hyun-Suk Lee) (Kyung-Hwan Oh)

**요 약** 지식을 관리하는 것에 주력했던 기존의 인공지능 연구 방향은 동적으로 움직이는 외부 환경에서 적응할 수 있는 시스템 구축으로 변화하고 있다. 이러한 시스템의 기본 능력을 이루는 많은 학습 방법 중에서 비교적 최근에 제시된 강화학습은 일반적인 사례에 적용하기 쉽고 동적인 환경에서 뛰어난 적응 능력을 보여주었다. 이런 장점을 바탕으로 강화학습은 에이전트 연구에 많이 사용되고 있다. 하지만, 현재까지 연구 결과는 강화학습으로 구축된 에이전트로 해결할 수 있는 작업의 난이도에 한계가 있음을 보이고 있다. 특히, 복수의 부분 작업으로 구성되어 있는 작업을 처리할 경우에 기존의 강화학습 방법은 문제 해결에 한계를 보여주고 있다.

본 논문에서는 복수의 부분 작업으로 구성된 작업이 왜 처리하기 힘든가를 분석하고, 이런 문제를 처리할 수 있는 방안을 제안한다. 본 논문에서 제안하고 있는 EQ-Learning은 강화학습 방법의 대표적인 Q-Learning을 확장시켜 기존의 문제를 해결한다. 이 방법은 각각의 부분 작업 해결 방안을 학습시키고 그 학습 결과들의 적절한 순서를 찾아내 전체 작업을 해결한다. EQ-Learning의 타당성을 검증하기 위해 격자 공간에서 복수의 부분작업으로 구성된 미로 문제를 통하여 실험하였다.

**주제어** 강화학습, Q-Learning, 지능형 에이전트 시스템

***Abstract*** Intelligent Agent System, in these days, has became a new trend of researches in Artificial Intelligence field. The goal of existing research has mainly based on the Knowledge Management having learning capability.

Reinforcement learning is easily applied to general problems and successively worked well in dynamic environments. For these advantages it has been frequently used for agent researches. Many researches show that reinforcement learning is not sufficient to solve general problems. Especially typical reinforcement learning method can scarcely solve the problem composed with multiple subtasks.

In this thesis, we investigates the reason why it is difficult to process problems composed with multiple subtasks, and proposes a new solution, EQ-Learning. EQ-Learning is a extended method of Q-Learning which is a typical type of reinforcement learning. It is composed with two steps. The first step is to learn how to solve each subtask. The second step is to find the adequate order to apply the learning result of step one, and then the problems can be solved through step 1, and 2. To validate usefulness of EQ-Learning, a maze problem composed of multiple subtasks in grid-world is used.

† 본 연구는 교육부 BK21 연구사업지원과 서강대학교 산업기술연구소의 부분지원에 의하여 이루어짐.
* 동양시스템즈
** 동양공업전문대학 전산경영기술공학부
*** 서강대학교 컴퓨터학과
주소 : 121-742
서울시 마포구 신수동 1번지 서강대학교 컴퓨터학과
전화 : 02)705-8489 FAX : 02)704-8273
Email : kwoh@ccs.sogang.ac.kr
연구 세부분야 : 인공지능, 에이전트시스템

## 1. 서론

'패턴 인식', '음성 인식', '전문가 시스템', '컴퓨터 비전' 등과 같이 특정 영역에 초점을 두었던 인공지능은 최근 들어 그 연구 범위가 변화하기 시작하였다. 인공생명체의 개념을 지닌 '에이전트(Agent)'라는 개념이 등장하면서 기존의 연구 범위에서 벗어나 넓은 범위에

걸쳐 새롭게 적용될 수 있는 가능성을 사람들에게 인식시키기 시작했다.

에이전트를 가장 포괄적으로 정의하자면 '사용자를 대신하여 사용자가 원하는 작업을 자동적으로 해결해 주는 소프트웨어'[1][2]라고 할 수 있다. 소프트웨어가 단순히 하나의 개념으로 정의되지 않고, 그 특성과 용도 등으로 여러 가지로 분류되듯이 에이전트 역시 큰 범주를 가리키는 용어일 뿐 그 특성도 용도에 따라 다르다. 하지만 인공생명체의 개념으로 바라본 에이전트에게 사용자들이 기대하는 특성은 일반적으로 외부 환경에 대한 적절한 반응(Reactivity) 및 적응(Adaptation), 자율적인 행동 결정(Autonomy), 능동적인 행동을 통한 목표 달성(Goal-oriented)등이 포함되어 있다. 이러한 일반적인 에이전트의 특성 중에서 본 논문에서 중심을 두려고 하는 부분은 '외부 환경에 대한 적응'이다. 외부 환경에 적응한다는 의미는 외부 환경에 대한 모든 지식이나 규칙 등을 제공하지 않고 스스로 환경에 대한 정보를 받아들이고 분석해서 외부 환경에 대해 최적의 행동을 이끌어 내는 것을 의미한다.

학습(Learning)은 환경에 대해 적응하는데 필요한 지식을 습득하는 행동이라고 할 수 있다. 학습 방법은 이미 많은 수가 제안되어 있고 강화학습(Reinforcement Learning)은 이런 방법들 중에서 비교적 최근에 제시된 방법으로, 외부에 교사(supervisor)를 가정한 상태에서 학습 과정을 진행하는 교사학습의 한 방법이다. 그러나 일반적인 교사학습이 정확한 입력과 출력의 쌍으로 이루어진 학습용 데이터를 통해 학습을 하는 것에 비해[1], 강화학습은 행동 결과에 따른 적절한 평가를 통해 학습을 한다[2]. 때문에 이전에 교사학습 방법이 가지고 있는 유동성 부족의 문제를 해결하고, 네트웍 환경과 같이 시시각각으로 급변하는 환경에서도 학습할 수 있는 높은 적응력을 보여주었다.

그러나 이런 장점에도 불구하고 실생활에 많은 문제를 강화학습을 이용하여 해결할 수는 없다. 실제 생활에서 존재하는 대부분의 작업들은 보통 각기 다른 부분 작업(Subtask)들이 여러 단계에 걸쳐 구성되어 있거나 병렬 작업 형태로 구성되어 있어 강화학습 과정 중에 주어지는 보상(Reward)의 전달 방향이 모호해진다는 문제를 가지고 있다. 또한 부분 작업들을 어떻게 나눌 것이며, 그 해결 방법들을 어떻게 통합시켜야 하는 지에 대한 고려도 필요하게 된다.

그러므로 기존에 강화학습이 가지고 있는 장점들을 살리면서 위에서 제기되는 문제점들을 해결하기 위해서 에이전트에게는 새로운 학습 방법이 요구된다. 즉, 학습된 에이전트는 하나의 복잡한 작업을 완수하기 위해 해당 작업을 여러 개의 단순한 부분 작업으로 나누고, 각각의 부분 작업을 해결하는 방법을 탐색하고, 그 방법들을 적절하게 조합할 수 있는 능력을 가져야만 된다.[3]. 본 논문에서 제시하는 EQ-Learning은 기존에 강화학습이 가지고 있는 한계를 극복하면서 앞에서 제시된 문제를 해결하는 방법으로 제안된 새로운 강화학습의 방법이다.

본 논문의 구성은 다음과 같다. 제 2장에서는 부분 작업들로 구성된 복잡한 환경에 대해 알아보고, 이를 해결하기 위해 본 논문에서 제시하고 있는 모델을 설명한다. 제 3장에서는 본 논문에서 제시하고 있는 모델의 타당성을 실험을 통해 알아보고, 마지막으로 제 4장에서는 결론과 앞으로의 연구방향을 제시한다.

## 2. 확장된 Q-Learning의 설계

실생활의 많은 작업들은 몇 개의 부분 목표(Sub Goal)를 가지는 부분 작업(Subtask)으로 구성되어 있는 복합 작업의 형태로 되어 있다. 이런 복합 작업을 해결하려면 각각의 부분 작업을 처리하는 방법은 물론 부분 작업들의 처리 순서까지 올바르게 학습해야한다. 그러나, 단순히 작업 결과에 따른 보상을 통해 학습을 이루는 강화학습의 방법에서 이러한 복합 작업을 처리하는 학습을 이루어 내기는 힘들다. 특히 모델을 기반으로 하는 강화학습의 경우에는 잘못된 학습을 야기할 가능성이 높다. 본 논문에서는 모델을 기반으로 하는 강화학습의 방법에서 이러한 복합 작업의 처리 문제를 해결하는 방안을 제시하고 있다.

여러 개의 부분 작업으로 이루어진 복합 작업은, 부분 작업들의 관계에 따라 순차형 작업(Sequential Task)과 무작위형 작업(Random Task)이 있다. 순차형 작업은 부분 작업들이 일련의 순서를 가지고 진행되는 복합 작업을 의미하고, 무작위형 작업은 이런 일련의 순서 없이 이루어지는 복합 작업을 의미한다. 이

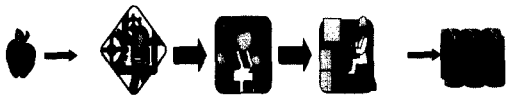
---

1) (x, y)와 같이 입력 x에 대해 출력이 y라는 사실을 반복적으로 학습시킨다. 그 결과 입력이 x가 들어왔을 때 학습된 기계가 y라고 올바른 출력을 만들어 낼 수 있으나, 입력으로 x와 유사한 x'가 들어왔을 때 출력으로 기대되는 y대신에 전혀 다른 y'가 출력으로 나올 수 있는 문제를 가지게 된다.

2) 강화학습의 기본 개념은 입력 x에 대해 y'를 출력으로 내놓았을 때에 주어지는 평가치 r이 커지는 방향으로 학습을 이루어 내는 것이다. 이런 특징은 학습 상황이 바뀌었을 때, 평가함수만 바꾸어주면 학습을 다시 이루어 낼 수 있다는 측면과 입력 x와 유사한 x'가 입력으로 들어왔을 때에도 비슷한 내용의 학습 결과를 보인다는 측면에서 기존 개념의 교사학습보다 유동성이 우수하다고 할 수 있다.

중 무작위형 작업은 포함하고 있는 부분 작업이 경우에 따라 필요하기도 하고 그렇지 않기도 하기 때문에, 모든 부분 작업이 특정한 순서를 갖는 순차형 작업에 비해서 해결하기 까다롭다는 특징을 가지고 있다.

(a) 단일 작업으로 관측되는 작업의 예



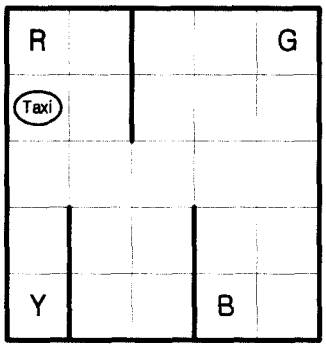
(b) 실제로 작업을 구성하는 부분 작업의 예

(그림 1) 순차형 작업의 예

순차형 작업은 여러 개의 부분 작업들이 하나의 작업을 이루지만 그 순서가 단계적으로 이루어진다는 특징이 있다. 순차형 작업의 대표적 예로 (그림 1)에 공장에서 제품을 생산하는 과정을 보이고 있다. (그림 1)의 (a)는 공장에서 처리하는 전체 작업을 간략하게 나타내고 있다. 하지만, 이처럼 공장에서 제품이 나오는 과정은 외부에서 보이는 것처럼 하나의 작업으로 이루어지는 것이 아니라 (그림 1)의 (b)와 같이 몇 개의 부분 작업으로 이루어지는 것이 일반적이다. 각 단계에서의 결과물은 다음 단계의 입력이 된다. 이런 과정이 반복되면서 전체 과정이 수행되고, 최종 단계 때에 제품이 나오게 되는 것이다.

무작위형 작업은 순차형 작업과 마찬가지로 여러 개의 부분 작업들이 하나의 작업을 이루고 있다. 그러나, 부분 작업들이 일정한 순서를 이루는 순차형 작업에 비해 무작위형 작업은 부분 작업간에 연관성이 상대적으로 약해서 사용되는 부분 작업의 내용과 순서가 경우에 따라 틀려진다.

(그림 2)는 복합형 작업에서 가장 대표적인 예인 'Taxi Domain'을 나타내고 있다[10]. 택시에 탑승하는 손님은 R, G, B, Y 중 한 장소에서 탑승할 수 있으며, 각각의 손님은 자신이 탑승한 장소를 제외한 나머지 세 장소 중 한 곳으로 가세된다. Taxi로 표현되는 에이전트의 목적은 손님이 위치하고 있는 장소로 가서 손님을 탑승시킨 후에 손님을 원하는 목적지에 내려주는 것이다. 복합작업은 R, G, B, Y에 도달하는 것을 목표로 하는 부분 작업들로 구성되지만, 손님이 발생하는 위치나 행선지에 따라 필요로 하는 부분 작업의 내용이 다르게 된다. 이렇듯 순차형 작업과는 달리 무작위형 작업은 부분 작업의 순서뿐만 아니라 어떤 부분 작업을 선택하느냐의 문제도 학습의 중심 과제가 된다.




(그림 2) Taxi Domain

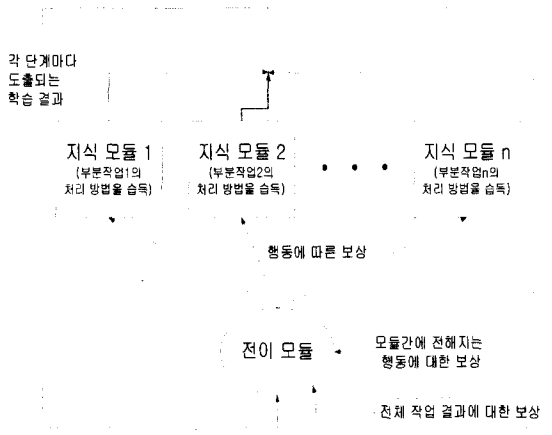
위와 같은 복합 작업의 처리 과정을 일반적인 강화학습으로 학습시키는 쉽지 않다. 물론 규칙(Rule)이나 지식(Knowledge)을 기반으로 하는 시스템을 사용해서 이 같은 문제를 해결할 수 있지만, 이렇게 되면 강화학습이 가지고 있는 높은 적응성을 잃게 된다. 이와 같은 예는 패킷 라우팅 문제를 강화학습으로 해결하는 과정에서 찾아볼 수 있다.[4]

실세계의 많은 경우에서 작업 결과에 대한 보상은 작업이 완전히 끝날 때까지 주어지지 않는다.[5] 이로 인해 각각의 부분 작업이 끝나도 에이전트에게는 어떠한 보상도 주어지지 않게 되고, 결국 학습은 올바로 수행되지 않는다. 이를 해결하기 위해 부분 작업을 나누는 능력, 부분 작업들의 순서를 정하는 능력 등을 학습시키는 별도의 과정이 필요하다. 본 논문에서는 복학 작업을 처리할 수 있도록 확장된 강화학습을 제안한다..

## 2.1 EQ-Learning (Extended Q-Learning) 개요

앞에서 제시되었던 부분 작업에 대한 지식 모듈을 강화학습에 응용해서 새롭게 이 논문에서 제시하고 있는 모델이 EQ-Learning이다. 이 방법은 복합 작업을 손쉽게 처리할 수 있는 능력을 가지면서, 이제까지 제시된 모델에 비해 어느 환경에나 쉽게 적용시킬 수 있는 장점을 가지고 있다.

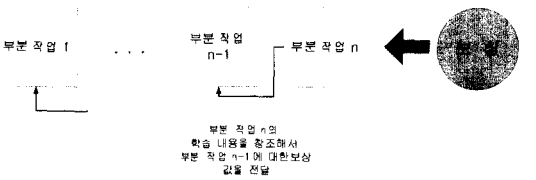
n개의 부분 작업으로 구성된 복합작업을 수행하기 위한 EQ-Learner의 전체적인 구조는 (그림 3)과 같다.




(그림 3) EQ-Learner의 구조

(그림 3)에서 지식 모듈들은 각각 서로 다른 하나의 부분 작업을 수행할 수 있는 방법을 학습하게 된다. 이 때 각 지식 모듈들은 전이 모듈을 통해서 보내지는 보상을 제외하고는 독립적으로 학습을 한다. 그리고 이러한 지식 모듈들이 어떠한 부분 작업을 맡아서 학습하게 되는가를 전이 모듈(Transition Module)에서 제어하게 된다. 전이 모듈에서 하는 일은 크게 두 가지이다. 첫째, 현재 수행해야 하는 부분 작업을 인지하고 해결하는 방법을 학습시킬 지식 모듈을 선택하고, 둘째 전체 작업이나 부분 작업을 완료했을 경우, 해당 지식 모듈에 적절한 보상을 하는 임무이다. 전이 모듈은 초기에 임의의 지식 모듈을 하나 선택해 현재 작업을 수행하게 한다. 해당 지식 모듈은 Q-Learning의 방법을 이용하여 부분 작업을 처리하는 방법을 학습하게 되고, 학습의 매 단계마다 도출되는 결과가 다시 전이 모듈의 입력으로 들어가게 된다. 전이 모듈은 현재 활동하는 지식 모듈의 수행 결과가 현재 작업의 목표를 달성하면 다른 지식 모듈을 활동하게 하거나 작업을 종료시키고, 작업의 목표를 달성하지 못하면 계속해서 하나의 지식 모듈에게 작업을 수행하게 한다.

부분 작업이 종료되면 전이 모듈은 다른 지식 모듈에서 다른 부분 작업을 수행하기 시작한다. 이 때, 이전까지 수행하던 지식 모듈은 부분 작업이 완료되어도 특정한 보상이 주어지지 않기 때문에, 학습을 제대로 이루지 못하게 된다. EQ-Learning에서는 이 문제를 해결하기 위해 이어지는 지식 모듈에 저장되어 있는 축적된 보상(Cumulative Reward)을 참조해 이전까지 처리한 부분 작업에 대해 보상을 내린다. 결과적으로 지식 모듈은 최종 결과에 대한 보상을 이전까지 수행했던 부분 작업들에 대해서 역전파(Back Propagation)시키는 역할을 한다. 이런 일련의 과정을 그림으로 나타낸 것이 (그림 4)이다.




(그림 4) 전체 작업에 대한 보상의 역전파

## 2.2 EQ-Learning에서의 학습 방법

기존에 강화학습 방법으로 제시되었던 Q-Learning의 원리는 다음과 같다. 일단 특정 상태에서 취하는 행동에 따라 Q-Value라는 기대치를 설정한다. Qf(x,a)로 표현되는 Q-Value는 정책 f에 따라 특정 상태 x에서 행동 a가 이루어내는 결과에 대한 기대치를 나타낸다. 그리고, Q-Learning은 이러한 Q-Value를 가장 극대화시키는 방향으로 정책을 수립하게 된다. Q-Learning을 통한 학습은 특정 상태에 있는 에이전트가 취한 행동에 대해 얻는 보상으로 시작된다. 시간 t일 때, 에이전트가 취한 행동에 대한 보상 r은 (1)과 같이 나타낼 수 있다.

$$r(t) = \sum_{i=0}^{\infty} \gamma^i R_{t+1} \qquad (1)$$

Rt+1은 시간 t+1에 받게 되는 보상을 의미하고, $\gamma$은 감소인자(Discount vector)로써 0과 1사이의 값을 갖는다. 이 감소인자는 목표를 달성했을 때 얻어지는 보상을 거리에 비례해서 감소시키는 역할을 한다. Q-Learning은 이런 보상 $r$을 가장 극대화시키는 방향으로 정책을 수립하게 되고, 이러한 보상은 감소인자 $\gamma$에 의해 순위가 매겨지기 때문에 보상이 이루어지는 거리가 짧은 방향으로 정책이 수립되게 된다. 정책(policy)은 현재 상태(state)에서 취할 수 있는 행동 중에서 가장 큰 Q-Value를 보장하는 행동을 취하는 방향으로 이루어지며 (2)와 같이 표현된다.

$$f(x) = \arg\max_{a \in A} Q(x, a) \qquad (2)$$

위와 같은 식으로 표현되는 정책을 통해 특정 상태

x에서 행동을 취해 상태 y로 전이될 때, Q-Value는 (3)식으로 표시될 수 있다.

$$Q_t(x,a) = (1-\alpha)Q_{t-1}(x,a) + \alpha[R(y) + \gamma U(y)] \quad (3)$$

$$U(y) = \max_{a \in A} Q(y,a)$$

($\alpha$는 학습률(Learning Rate), $R(y)$는 상태 y와 관련된 보상을 의미하고 $U(y)$는 다음 상태 y에서 얻을 수 있는 최대의 Q-Value를 의미한다.)

EQ-Learning은 앞에서 설명한 Q-Learning의 식에 부분 작업에 대한 고려를 추가시키는 것을 근간으로 하고 있다. 아래와 같은 순서에 의해 구성된다. 본 논문에서 제시하고 있는 EQ-Learning은 (4)와 같이 구축된다.

$$EQ_t(ta,x,a) = (1-\alpha)Q_{t-1}(ta,x,a) + \alpha[r(T) + \gamma U(ta,y)] \quad (4)$$

단, $U(ta,y) = \max_{a \in A} EQ(ta,y,a)$

r(T)는 보상 함수, ta는 현재 수행하는 부분 작업

일반적인 Q-Learning이 단순히 상태와 해당 상태에서의 행동만을 고려해서 작업의 해답을 구축하는 데 반해, EQ-Learning은 기존의 상태 x와 행동 a를 취할 때 수행해야 하는 부분작업 ta를 고려해서 학습을 이루고 있다. 하지만, 전체 작업이 끝날 때에만 학습을 위한 보상이 내려지게 되므로, 위에서 보상 함수를 나타낸 R(T)는 이 점을 고려해서 설계해야 한다. 전체 작업을 T, T를 이루는 부분 작업들을 ti(단, $0 \le i \le n$, n은 부분 작업의 갯수)라 하자. 부분 작업 tk-1이 완료된 후에 부분 작업 tk를 시작한다(단, $1 \le k \le n$). 그러면 전체 작업 T는 $T=\{t_1, t_2, t_3 \ldots t_n\}$라고 표기할 수 있다. 작업에 대한 보상은 부분 작업 tn-1까지의 모든 부분 작업을 합당한 순서에 따라 모두 완료한 후에 부분 작업 tn의 수행을 끝내면 주어지게 된다. 보상이 전체 작업이 끝났을 경우에만 주어지기 때문에 각각의 부분 작업을 수행하는 과정을 학습하기 위해서는 위에서 언급했던 바와 같이 전체 작업에 대한 보상을 각각의 부분 작업에 대해 역전파 시켜줘야 한다. 이를 위해 보상함수를 아래의 식들과 같이 정의한다. 우선 전체 작업이 끝났을 때의 보상 함수는 (5)와 같이 일정한 상수를 부여하는 방식으로 이루어진다.

$$r(T) = c \text{ (단, c는 상수)} \quad (5)$$

그리고, k 번째 부분 작업을 수행하는 중에 특정 상태에서 취한 행동에 의해 상태 y로 전이되면서 수행하던 부분작업이 완료하게 되면 (6)과 같이 정의된 보상이 주어지게 된다.

$$r_k(T) = \max_{a \in A} EQ(ta_{k+1}, y, a) \quad (6)$$

단, rk : k번 째 부분작업이 완료되었을 때의 보상,
sk : k번 째 부분작업을 수행하는 상황(단, $0 \le k \le n-1$)

위의 식은 하나의 부분 작업이 끝났을 때의 상태는 다음에 이어지는 부분 작업이 시작하는 상태와 일치한다는 점에서 착안한 것이며, 위에서 언급했던 전체 작업에 대한 보상을 부분 작업으로 역전파시킨다. 이 과정을 다음과 같이 자세히 표현할 수 있다.

tn에 대한 보상 : $R_n = c$(전체 작업이 끝났을 경우)
tn-1에 대한 보상 : $R_{n-1} = \gamma^{m(n-1)} R_n = \gamma^{m(n-1)} c$

.
.
.

t1에 대한 보상 : $R_1 = \gamma^{m(1)} R_2 = \gamma^{m(1)} \gamma^{m(2)} \ldots \gamma^{m(n)} c$

Rk는 부분 작업 k가 끝났을 때의 보상이고 c는 전체 작업 완료에 따른 보상 정도이며 $\gamma$는 감소 인자, m(k)는 부분 작업 k를 수행할 때 거치게 되는 상태의 수를 나타낸다.

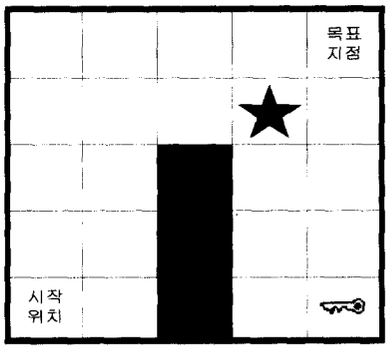
이와 같이 EQ-Learning은 전체 작업에 대한 보상을 각각의 부분 작업으로 역전파시킴으로써 학습의 과정에서 필요한 보상을 전달해주고 있고, 이를 통해서 부분 작업들의 해결 방법이 각각의 지식 모듈에 작성될 수 있다.

## 3. 실험 및 결과

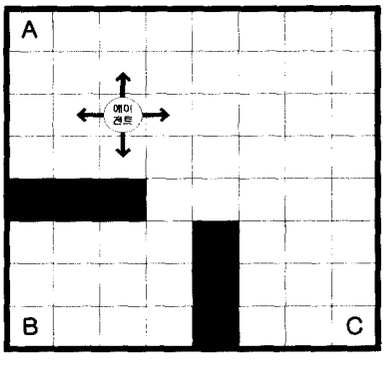
본 절에서는 3절에서 제안한 EQ-Learning의 타당성과 성능을 검증하기 위해, 강화학습으로 학습된 에이전트의 성능을 시험하기 위해 많이 사용하는 격자 공간에서의 미로 문제를 사용하였다. 이을 통해 기존 강화학습 방법의 한계점을 알아보고 EQ-Learning을 통해 기존 방법의 문제를 해결해 보임으로써 본 논문에서 제시된 방법의 타당성을 증명한다.

### 3.1 실험방법

실험은 크기가 다른 두 개의 격자 공간에서 이루어졌다. 각각의 세계의 크기는 5x5와 8x8이며, (그림 5)와 같다.




(a) 실험 1에서 사용하는 5x5의 격자 공간




(b) 실험 2에서 사용하는 8x8의 격자 공간

(그림 5) 실험에 사용되는 격자 공간

(그림 5)의 (a)는 EQ-Learning의 타당성을 검증하기 위한 실험 공간이다. 이 영역에서 에이전트는 왼쪽 아래 '시작 위치'에서 출발해서, 오른쪽 아래에 있는 열쇠를 획득한 후에, 오른쪽 위에 있는 출구를 열고 나가는 방법을 학습해야 한다. 가운데 회색으로 칠해진 영역과 5x5의 Grid 영역의 가장자리는 벽으로 막혀져 있기 때문에 에이전트가 통과할 수 없다. 열쇠를 획득해서 출구에 도착하면 보상 100이 주어진다.

(그림 5)의 (b)는 1992년 Singh이 자신의 논문에서 제시한 학습 모델을 시험하기 위해 사용했던 격자 공간이다[3][6]. 단순 작업(Simple task)과 복합 작업에서의 학습, 작업 내용의 추가 및 변경 시의 적응에 대한 실험을 수행할 수 있다. 단순 작업은 임의의 위치에서 출발해서 A, B, C 세 목표점 중 하나에 도달하는 것이고, 복합 작업은 임의의 위치에서 출발해서 한 점을 거쳐 다른 점에 도달하는 것이다. 작업 내용의 변경 및 추가 실험의 목적은 해결해야 하는 부분 작업을 추가하거나 변경함으로써 임의의 외부 환경 변화에 대한 적응력을 알아보는 것이다. 본 논문에서는 단순 작업과 복합 작업을 처리하는 실험만 행해지며, 작업 목표를 달성했을 때에 보상으로 100이 주어진다.

이 실험들에서 강화학습의 탐색 전략으로 볼쯔만(Boltzmann)[11] 탐색 전략을 사용하고 있다. 볼쯔만 탐색 전략은 식 (4)의 볼쯔만 분포를 이용하여 탐색을 해나간다.

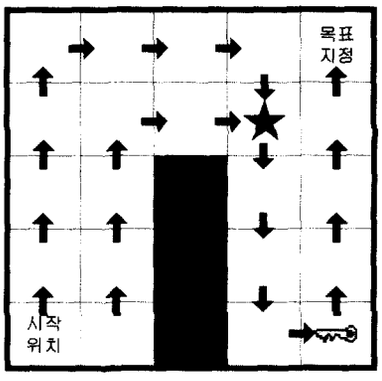
$$p(a_i \mid s) = \frac{e^{\frac{Q(s,a_i)}{T}}}{\Sigma_j e^{\frac{Q(s,a_j)}{T}}} \qquad (4)$$

상태 s에서 취할 수 있는 모든 행동 aj에 대한 확률 $p(a_i \mid s)$를 계산한 후에, (0, 1)사이의 임의의 난수와 비교해 구해진 확률이 높은 값을 가지고 있으면 그 행동을 취하게 되고, 그렇지 않으면 임의의 행동을 취하게 된다. 이 탐색 방법은 학습이 충분히 이루어진 후에도 학습의 내용을 완벽하게 활용할 수 없는 단점이 있지만, 각 상태에서 취할 수 있는 행동들에 골고루 확률을 부여해서 새로운 행동을 학습하는 기회를 가진다는 장점이 있다[7]. 위에서 제시된 실험은 최적의 행동을 찾아야하는 것보다 계속적인 학습이 보장되는 가운데 전반적인 성능을 평가하는 데 중점을 두었기 때문에 볼쯔만 탐색 전략을 사용했다.

볼쯔만 탐색 전략을 사용한 Q-Learning[8]은 최종적인 학습 결과는 거의 비슷한 양상을 보이나 행동의 임의성을 조절하는 온도(Temperature)의 크기와 변화 정책, 학습률(Learning Rate), 결과에 따른 보상의 크기 등의 인자에 따라 학습 시간과 결과에 약간의 영향을 미친다. 본 실험은 제시하는 EQ-Learning의 타당성만을 증명하는 실험이기 때문에 인자 결정 실험을 하지 않았고, 보상의 크기는 격자 공간의 크기가 비슷한 환경에서의 실험[9]을 참조해서 정했다.

### 3.2 실험 결과 : EQ-Learning의 타당성 증명

본 실험의 실험 공간은 (그림 5)의 (a)와 같다. 이 실험공간에서 모델을 기반으로 하는 기존의 강화학습 방법으로 제시된 문제를 해결하는 과정을 학습하려면, ★로 표시된 상태에서 어떤 행동을 학습하느냐가 최종 학습 결과에 지대한 영향을 미치게 된다. 만약 이 상태에서 아래로 가는 행동을 학습하지 못하면, 학습의 '시작위치'를 고려할 때, 열쇠를 획득하는 방법을 제대로 학습하지 못하는 결과를 야기하게 된다.

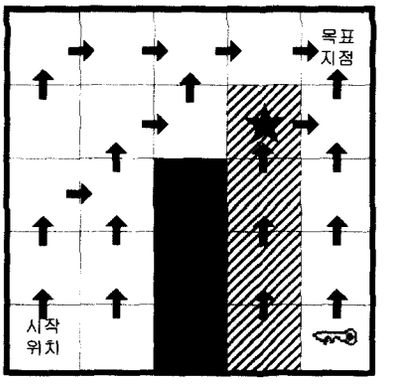



(그림 6) 이상적인 학습결과

(그림 6)은 가장 이상적인 학습 결과를 나타내고 있다. ★이 놓여진 상태에서 아래로 가는 행동을 취하기 때문에 열쇠가 위치한 상태에 이를 수 있었고, 결과적으로 시작위치에서 출발해 열쇠를 습득한 후에 출구에 이르는 과정이 올바로 학습될 수 있었다. 그러나 Q-Learning을 이용하면 이러한 이상적인 학습 결과는 쉽게 얻어지지 않는다. 일반적으로 특정 상태에서 취하는 행동은 보상이 가장 큰 쪽으로 움직이게 되어 있다. 감소 인자로 인해 최종 목표 상태에 가까울수록 많은 보상이 주어지기 때문에 대부분의 행동은 최종 목표 상태에 가까운 방향으로 움직이려는 성향을 보인다.

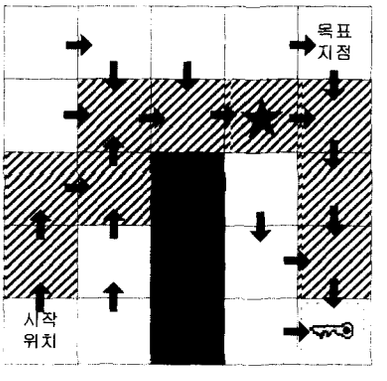
약 30회에 걸친 학습 결과의 평균으로 구해진 값을 통해 특정 상태에서 선호하는 행동 양식을 알아보았다. 한 회의 학습은 500번의 학습 단계에 걸쳐 반복적으로 이루어지며, 1번의 학습 단계는 '시작위치'에서 출발한 에이전트가 열쇠를 획득한 후에 오른쪽 위의 출구에 도달하거나 1000회의 행동으로도 이 문제를 해결하지 못하면 문제를 해결하지 못하고 종료하게 된다. (그림 7)은 Q-Learning을 통해 위에서 언급한 실험 과정의 결과로써 얻어진 행동을 나타내고 있으며 각각의 행동은 해당 상태에서 가장 높은 확률로 취해진 행동들이다.

(그림 7)에서 나타내고 있는 학습 결과는 시작 위치에서 열쇠를 획득하는 과정이 바르게 학습되지 않은 것을 보여주고 있다. (그림 7)에서 빗금이 그려진 상태에서의 학습 결과는 (그림 6)에서 나타낸 이상적인 학습 결과와는 반대되는 것을 보이고 있다. 결국 (그림 7)에서 표시된 학습 결과로는 '시작 위치'에서 열쇠가 놓인 상태를 거쳐 '목표 지점'까지 이르는 행동을 취할 수 없다. 즉, 기존의 강화학습이 하나의 목표만을 가정한 환경에서 주어지는 평가를 통해 학습할 때에 발생하는 문제점을 해결하는 방안이 요구된다.

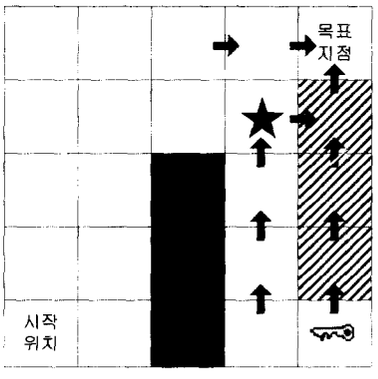



(그림 7) Q-Learning의 실험결과
(빗금이 그려진 상태에서 잘못된 학습 결과를 보여준다.)

EQ-Learning을 사용한 학습 결과는 (그림 8)에서 나타난다. EQ-Learning은 복합작업을 구성하고 있는 부분 작업에 따라 나누어 각각 학습을 하며, 만족할만한 학습 결과를 보여준다. 이 때, 전이모듈은 열쇠를 획득했을 때 상황이 변한다는 것을 인지하고 있으며, 두 개의 지식 모듈간에는 감소인자로 인한 Q-Value의 차이가 있기 때문에 이 크기를 기준으로 지식모듈 간의 순서를 정한다. 본 논문에서 가정하고 있는 전이모듈의 사전 지식은 '열쇠 인식'이 전부이며, '열쇠 인식'이 행해 졌을 때 전이모듈은 새로운 학습 시기임을 인지하고 현 상태의 Q-Value를 다음 지식 모듈의 보상으로 이용한다.




(a) 열쇠를 획득하는 부분 작업의 학습 결과




(b) 출구를 탐색하는 부분 작업의 학습 결과

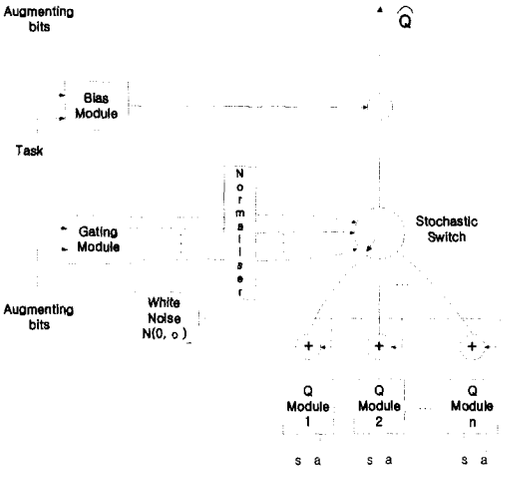
(그림 8) EQ-Learning의 실제 실험 결과
(전체 작업을 두 개의 부분 작업으로 나누고 각 부분 작업의 목표를 이루는 학습 결과가 빗금으로 표시되어 있다.)

(그림 8)에서 보는 것과 같이 EQ-Learning은 부분 작업에 따라 지식 모듈을 나누어서 각각의 부분작업에 적절한 행동을 학습한다. (a), (b)의 빗금이 그려진 영역은 부분 작업에 따른 올바른 학습 결과가 존재한다는 것을 보여주고 있다. 단, (b)에서 왼쪽 부분이 학습이 되지 않은 이유는 ★에서 오른쪽으로 움직이는 행동이 학습된 후에, ★ 왼쪽으로 이동하는 확률이 낮기 때문에 이런 현상이 나타난 것이다.

이는 각각의 부분 작업을 Q-Learning으로 학습한 후에 이를 통해 복합 작업을 해결하는 방법과는 차이가 있다. 첫째 일반적인 작업을 수행할 때, 중간 과정마다 어떠한 보상을 부여하기 힘들다는 점이다. 위의 예를 적용하면 미로를 탈출했을 때에 보상을 부여하는 것은 당연하지만, 미로를 탈출하기 위해 열쇠를 획득하는 부분 작업을 수행했을 때에 보상을 주어지는 것은 무리가 있다. 둘째, 설사 이런 식으로 중간에 보상이 주어진다고 해도, Q-Learning으로 학습된 결과들을 어느 순서로 적용할 것 인가도 문제가 된다. 두 개의 학습 결과를 일정한 순서로 적용시키려면 어떤 기준이 필요하지만, 위에서 제기한 이의는 이런 점을 간과하고 있다.

본 논문에서 제시한 EQ-Learning은 해결해야할 부분 작업을 처리하는데 지식모듈을 사용함으로써 첫 번째 언급되었던 문제를 극복하였고, 결과에 대한 보상이 일정한 감소 인자에 의해 다른 지식 모듈들로 감소되어 가면서 전이되어 가기 때문에 이 차이에 의해 지식모듈간의 순서를 정함으로써 두 번째에 제시된 문제를 해결할 수 있었다.

### 3.3 이전 모델과의 비교 실험 결과

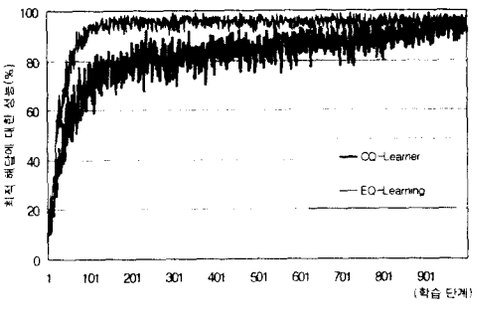



〈그림 9〉 CQ-Learner Structure

〈그림 9〉는 Singh이 1992년에 기본 작업(Elemental Task)을 기반으로 해서 임의의 순차형 작업을 처리하기 위해 제시된 CQ-Learner[3]을 나타낸다. 기본 작업은 상태 s와 행동 a를 입력으로 하는 Q-Learning을 통해 학습되며, Q-Module이 그 내용을 저장한다. 전체 작업은 이런 구성 작업들이 일정한 순서로 배열된 형태로 이루어지며, "augmenting bits"라고 불리는 2진 비트들이 구성작업과 그 순서를 나타낸다. Gating 모듈은 Jacobs에 의해 만들어진 "Gating Architecture"를 개량시켜 만들어진 것으로, 각각의 Q-Module이 하나의 기본 작업만을 담당하도록 제어하는 역할을 학습한다. 가장 위에 위치한 Bias 모듈은 현재 진행하는 작업이 목표로 하는 구성 작업에 얼마나 합당한지를 학습한다.

CQ-Learner는 이미 기본 작업들을 학습했을 경우에 그 학습 내용을 이용해 복합 작업을 쉽게 처리한다는 장점이 있다. 그러나 기본 작업, Bias 모듈과 Gating 모듈이 동시에 학습되어야 하기 때문에 효율이 떨어지는 단점이 있다. 그리고 Gating 모듈과 Bias 모듈이 신경망으로 구성되어 있어 학습시간이 오래 걸리고, 복합 작업을 구성하는 기본 작업들과 그 순서를 "augmenting bit"로 인위적으로 표시해야 한다는 문제점을 가지고 있다.

비교실험으로 CQ-Learner와 단순작업을 처리하는 성능을 비교했다. 실험 환경은 앞에서 언급한 (그림 5)의 (b)가 되겠다. CQ-Learner는 먼저 부분 작업에 대한 해답을 구한 후에 이를 통해 복합 작업을 처리하기 때문에, 부분 작업 처리에서 보여주는 성능만을 비교했다.




(그림 10) 비교 실험의 결과

(그림 10)에서 보는바와 같이 EQ-Learning이 100회의 학습 단계를 넘어설 때, 최적의 해답에 대해 95%이상의 성능을 보여주는데 비해, CQ-Learner는 900회의 학습 단계가 지나면서 비슷한 성능을 보여주기 시작한다. 복합 작업을 학습하는데 EQ-Learning이 소비하는 학습 시간은 구성하는 부분 작업의 수에 따라 약간씩 증가하는 데 비해, CQ-Learner는 부분 작업의 수에 비례해서 산술 급수적으로 증가하기 때문에 본 논문에서 제시하고 있는 EQ-Learning은 이전에 제시되었던 CQ-Learner에 비해 우수한 성능을 보인다고 말할 수 있다.

## 4. 결론

본 논문에서는 기존의 강화학습 방법으로 쉽게 해결할 수 없었던 복합 작업을 처리할 수 있는 EQ-Learning의 방법을 제안하였다. 제안된 방법은 기존에 제시되었던 방법들이 가지고 있었던 본질적인 문제를 해결함과 동시에 그 전까지 존재했던 제약사항들을 극복하고 나아진 성능을 보여주었다. 무엇보다 이전에 제시되었던 방법들에 비해 구조적으로 간단해졌기 때문에 어느 문제에서나 적용하기 쉽다는 장점을 가지고 있다.

그러나 아직 몇 가지 해결할 문제가 남아 있다. 본 논문에서 해결한 문제는 모두 구성하는 부분 작업이 확실하게 분리되었다. 하지만, 경우에 따라서 구성하는 부분 작업이 분명한 경계를 가지고 지 않고, 하나의 부분 작업이 서서히 끝나감에 따라 다른 부분 작업이 서서히 시작될 수 있다. 이런 문제를 해결하기 위해 전이 모듈에 퍼지 이론을 결합하는 연구가 진행 되야 할 것이다.

그리고, 환경을 구성하는 상태의 수와 복합 작업을 구성하고 있는 부분 작업들의 수에 비례해서 학습에 사용되는 비용이 증가한다는 문제가 있다. 이 문제를 해결하기 위해 각 지식 모듈을 계층적인 구조로 구성하는 방법이나, 부분 작업들의 유사 정도에 따라 학습 내용을 서로 이용할 수 있는 방법을 마련하는 연구가 진행되어야 할 것이다.

마지막으로 지식모듈의 생성과 삭제에 대한 고려도 필요할 것이다. 본 논문에서는 일정 수의 지식 모듈을 마련하고 새로운 상황이 들어올 때마다 남아 있는 지식 모듈을 학습시키는 방법을 취했다. 그러나 실제 생활에서는 필요에 따라 해결 방안을 학습하고, 이미 학습한 내용이라도 별로 쓸모가 없을 경우 서서히 기억에서 지워나간다. 이러한 생물의 학습 방식을 차용하는 방법을 키워내 학습된 내용을 담는 공간에 유동성을 확보해야 한다.

이런 해결해야할 문제점에도 불구하고 본 논문에서 제시된 방법은 기본적인 학습방법을 개량하였기 때문에 본 논문에서 실행한 격자 공간의 예를 제외하더라도 네트웍 라우팅의 학습, 이동형 에이전트의 제어 등에 사용될 수 있을 것이다.

## 참고 문헌

[1] Padraig Cunningham and Richard Evans (1997), "Software Agents : A review", *http://www.cs.tcd.ie/Brenda.Nangle/iag.html.*

[2] 최중민 (1997), "에이전트 개요와 연구방향", **정보과학회지**, 제 15권, 제 3호, 7-16쪽.

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

[4] Justin Boyan, Mitchael Littman (1993), "A Distributed Reinforcement Learning Schme for Network Routing", *Technical Report CMU-CS-93-165.*

[5] P. Tadepalli and Thomas G. Dietterich (1997), "Hierachical Explanation-Based Reinforcement Learning.", *Machine learning research. AI Magazine (pp. 97-136)*, 1997

[6] S. P. Singh(1992) "On The Efficient Learning of Multiple Task Sequences", *Advances in Neural Information Processing Systems 4*, 1992

[7] 고영범 (1998), "변형된 볼쯔만 탐색 전략을 사용하는 강화 학습 기반 지능형 에이전트", 서강대학교 석사 학위 논문.

[8] Christoper J. C. H. Watkins and Peter Dayan

(1992). "Q-Learning". *Machine Learning 8*. pp. 279-292.

[9] Marco Wiering and Jurgen Schmidhuber (1997). "HQ-learning: Discovering Markovian subgoals for non-Markovian reinforcement learning". *Tech. rep., Istituo Dalle Molle di Studi sull'Intelligenza Artificiale. Lugano. Switzerland.*

[10] Thomas G. Dietterich (1999). "Hierarchical Reinforcement Learning with the MAXQ Value Function Decomposition". *Tech.rep., Department of Computer Science. Oregon State University.*

[11] Kaelbling. Lesile P and Michael L. Littman (1996). "Reinforcement Learning : A Survey". *Journal of Artificial Intelligence Research. vol. 4.* pp. 237-285. 1996